

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

IRIS: PLATAFORMA PARA TRIAGEM DE DEFICIÊNCIAS
VISUAIS DE CRIANÇAS EM IDADE ESCOLAR

FILIPE RODRIGO MIGUEL

BLUMENAU
2016

FILIPE RODRIGO MIGUEL

**IRIS: PLATAFORMA PARA TRIAGEM DE DEFICIÊNCIAS
DE CRIANÇAS EM IDADE ESCOLAR**

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

Prof. Aurélio Faustino Hoppe, Mestre - Orientador

**BLUMENAU
2016**

**IRIS: PLATAFORMA PARA TRIAGEM DE DEFICIÊNCIAS
VISUAIS DE CRIANÇAS EM IDADE ESCOLAR**

Por

FILIPE RODRIGO MIGUEL

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

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

Membro: _____
Prof. Matheus Luan Krueger, Mestre – FURB

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

Blumenau, 6 de dezembro de 2016.

Dedico este trabalho aos meus pais que sempre me incentivaram nos estudos e ao meu irmão que, apesar de não estar mais entre nós, sempre me acompanhou nessa jornada e acreditava na conclusão deste trabalho.

AGRADECIMENTOS

A Deus, que por todos zela.

À minha família, em especial minha mãe, que tanto se sacrificou e me incentivou em buscar conhecimento.

Aos meus amigos, pelo suporte e companheirismo oferecidos nessa jornada.

Ao meu orientador, Aurélio Faustino Hoppe, por ter acreditado na conclusão deste trabalho.

O sucesso é ir de fracasso em fracasso sem
perder o entusiasmo.

Winston Churchill

RESUMO

Este trabalho apresenta o desenvolvimento de uma plataforma para dispositivos móveis Android, voltada para triagem de anomalias visuais para crianças em idade escolar. Tal plataforma, permite ao corpo docente de uma instituição de ensino ou da secretaria de educação municipal o cadastro de testes a serem aplicados nas escolas. Após a aplicação dos testes, a plataforma disponibiliza relatórios detalhados sobre o desempenho dos avaliados para cada teste, permitindo o acompanhamento da evolução do aluno em determinado exame, sendo ilustrado na forma de gráficos. A partir desses dados, a idéia é que os responsáveis possam ter evidências mais concretas para avaliar a performance dos alunos e tomar as ações mais concisas em benefício dos estudantes. Apesar do desenvolvimento da plataforma ter sido focada para triagem de anomalias visuais nos alunos, é permitido o cadastro de novos testes de qualquer temática, não necessariamente para deficiências visuais, podendo ser utilizada para cadastrar provas ou tarefas das mais diversas matérias escolares, para fixação de conteúdo e análise de desempenho. A plataforma foi desenvolvida com o *framework* Phonegap e seus *plugins*, utilizando as tecnologias *web* tais como HTML, CSS, Javascript e AngularJS, além do *framework* Ionic. Os resultados dos testes da plataforma pelos usuários mostraram que a plataforma é capaz de auxiliar em diagnósticos de distúrbios visuais de baixa ordem e daltonismo porém, não serve como diagnóstico conclusivo. Os usuários julgaram interessante poder salvar os resultados e acompanhar a evolução de possíveis distúrbios para que com essa informação inicial possam agir mais assertivamente sobre os casos mais evidentes, visto que não há opções acessíveis conhecidas para realizar o diagnóstico de anomalias visuais em escolas.

Palavras-chave: Anomalias visuais. Daltonismo. Miopia. Hipermetropia. Teste de Ishihara. Tabela de Snellen. Triagem. Ionic. Phonegap.

ABSTRACT

This work presents the development of a platform for Android mobile devices, aimed at screening of visual anomalies for school children. This platform allows the faculty of a teaching institution or the municipal education secretariat to register the tests to be applied in schools. After the application of the tests, the platform provides detailed reports on the performance of the evaluated ones for each test, allowing the monitoring of the evolution of the student in a given exam, being illustrated in the form of graphs. From this data, the idea is that those responsible can have more concrete evidence to evaluate student performance and take the most concise actions to the benefit of students. Although the development of the platform has been focused on screening for visual anomalies in students, it is possible to register new tests of any subject, not necessarily for visual deficiencies, and can be used to register tests or tasks of the most diverse school subjects, to remind the classes contents and performance analysis. The platform was developed with the Phonegap framework and its plugins, using web technologies such as HTML, CSS, Javascript and AngularJS, in addition to the Ionic framework. The results of the tests of the platform by the users showed that the platform is able to aid in diagnoses of visual disturbances of low order and color blindness, however, it does not serve as conclusive diagnosis. The users thought it interesting to save the results and monitor the evolution of possible disorders so that with this initial information they can act more assertively on the most obvious cases, since there are no known accessible options for the diagnosis of visual anomalies in schools.

Key-words: Visual condition. Color blindness. Short-sightedness. Long-sightedness. Ishihara test. Snellen chart. Screening. Ionic. Phonegap.

LISTA DE FIGURAS

Figura 1 - Como a criança enxerga.....	17
Figura 2 - Triagem de deficiências visuais em uma escola do Piauí.....	18
Figura 3 - Lâminas utilizadas no Teste de Ishihara	20
Figura 4 - Teste de Snellen	22
Figura 5 - Tabela de Snellen para indicar direção da letra “E”	23
Figura 6 - Arquitetura do <i>framework</i> Phonegap.....	24
Figura 7 - Arquitetura do <i>framework</i> Ionic	27
Figura 8 - Lâminas do Teste de Ishihara citadas por Paula Júnior (2014)	28
Figura 9 - Teste de acuidade visual	30
Figura 10 - Teste de foria	31
Figura 11 - Telas e funcionalidades do Eye Test – Eye Exam.....	32
Figura 12 - Oftalmoscópio utilizado no aplicativo Peek Vision	33
Figura 13 - Mapa de localizações da aplicação dos testes do Peek Vision	34
Figura 14 - Triagem de anomalias visuais em escola do Condado de Trans Nzoia no Quênia.....	34
Figura 15 - Casos de uso da plataforma	39
Figura 16 - Diagrama de classes do front-end.....	41
Figura 17 - Diagrama de pacotes da plataforma.....	42
Figura 18 - Conteúdo do pacote db.....	43
Figura 19 - Diagrama de classes do sub-pacote dao	44
Figura 20 - Diagrama de classes do sub-pacote model	45
Figura 21 - Diagrama de classes do pacote filter.....	45
Figura 22 - Diagrama de classes do pacote mapper	46
Figura 23 - Diagrama de classes do pacote service.....	47
Figura 24 - Diagrama de classes do pacote utils	48
Figura 25 - Arquitetura da plataforma.....	48
Figura 26 - Diagrama de atividades.....	56
Figura 27 - Tela de login	57
Figura 28 - Menu principal.....	58
Figura 29 - Tela de cadastro de teste	58
Figura 30 - Exemplo de cadastro de teste.....	59
Figura 31 - Visualização dos testes cadastrados.....	60

Figura 32 - Menu de usuário.....	60
Figura 33 - Opções de usuário.....	61
Figura 34 - Fluxo da aplicação do teste ao aluno	61
Figura 35 - Cadastro do aluno	62
Figura 36 - Menu de relatórios	62
Figura 37 - Relatório do aluno em determinado teste	63
Figura 38 - Relatório por teste.....	64
Figura 39 - Acompanhamento do histórico do aluno	64

LISTA DE QUADROS

Quadro 1 - Roteiro para a promoção da saúde ocular na infância.....	19
Quadro 2 - Teste para detectar a severidade do daltonismo	21
Quadro 3 - Suporte de plataformas do Phonegap	25
Quadro 4 - Siglas correspondentes às funcionalidades dos sistemas	29
Quadro 5 - Comparativo dos aplicativos <i>Android</i> relacionados ao daltonismo	29
Quadro 6 - Comparativo dos aplicativos <i>iOS</i> relacionados ao daltonismo	30
Quadro 7 - Comparação com os trabalhos correlatos.....	35
Quadro 8 - Configuração inicial do cache	51
Quadro 9 - Configuração das instâncias dos caches.....	52
Quadro 10 - Utilização dos caches	52
Quadro 11 - Utilizando a biblioteca de imagens do dispositivo móvel.....	53
Quadro 12 - Configurando os mapeadores	54
Quadro 13 - Mapeamento de comandos SQL para métodos Java.....	55
Quadro 14 - Respostas às tarefas aplicadas aos administradores e executadores de testes.....	66
Quadro 15 - Avaliação de usabilidade da plataforma.....	68
Quadro 16 - Respostas relacionadas às funcionalidades e à eficácia do aplicativo	69
Quadro 17 - Dispositivos utilizados no experimento	71
Quadro 18 - Comparação com os trabalhos correlatos.....	72
Quadro 19 - Questionário de perfil do usuário	77
Quadro 20 - Lista de tarefas a serem executadas pelo administrador	77
Quadro 21 - Lista de tarefas a serem executadas pelos executadores	79
Quadro 22 - Questionário de usabilidade	81

LISTA DE ABREVIATURAS E SIGLAS

API – Application Programming Interface

CBO – Conselho Brasileiro de Oftalmologia

IBGE – Instituto Brasileiro de Geografia e Estatística

MEC – Ministério da Educação

OMS – Organização Mundial de Saúde

SEDPD – Secretaria Especial dos Direitos da Pessoa com Deficiência

URL – Uniform Resource Locator

SUMÁRIO

1 INTRODUÇÃO.....	14
1.1 OBJETIVOS.....	15
1.2 ESTRUTURA.....	15
2 FUNDAMENTAÇÃO TEÓRICA	17
2.1 PROGRAMAS DE TRIAGEM DE ANOMALIAS VISUAIS.....	17
2.2 TESTE DE ISHIHARA.....	19
2.3 TESTE DE ACUIDADE VISUAL	22
2.4 PHONEGAP.....	23
2.5 IONIC	26
2.6 TRABALHOS CORRELATOS	27
2.6.1 Proposta de um aplicativo móvel <i>Open-Source</i> em auxílio a indivíduos com discromopsia baseado em um estudo qualitativo	28
2.6.2 Sistema de triagem visual e auditiva de crianças em idade escolar, conectado a um banco de dados	30
2.6.3 Eye Test – Eye Exam	32
2.6.4 Peek Vision	33
2.6.5 Comparativo e discussão das características dos trabalhos correlatos.....	35
3 DESENVOLVIMENTO.....	37
3.1 REQUISITOS.....	37
3.2 ESPECIFICAÇÃO	38
3.2.1 Diagrama de casos de uso	38
3.2.2 Diagrama de classes	40
3.2.3 Arquitetura e integrações do aplicativo.....	48
3.3 IMPLEMENTAÇÃO	49
3.3.1 Técnicas e ferramentas utilizadas.....	49
3.3.2 Principais implementações da plataforma.....	51
3.3.3 Diagrama de atividades	55
3.3.4 Operacionalidade da implementação	57
3.4 ANÁLISE DOS RESULTADOS	65
3.4.1 Análise dos resultados da lista de tarefas	65
3.4.2 Análise quanto à usabilidade e funcionalidades.....	67

3.4.3 Testes de carregamento de dados, layout e compatibilidade	70
3.4.4 Comparação com os trabalhos correlatos.....	71
4 CONCLUSÕES.....	73
4.1 EXTENSÕES	74
REFERÊNCIAS	75
APÊNDICE A – LISTA DE TAREFAS E QUESTIONÁRIO DE AVALIAÇÃO DE USABILIDADE.....	77

1 INTRODUÇÃO

De acordo com o último censo realizado no Brasil pelo Instituto Brasileiro de Geografia e Estatística (IBGE) em 2010, foi relatado a existência de algum tipo de anomalia visual em pelo menos 36 milhões de habitantes (SEDPD, 2010). Dentre estes, 0,02% apresentam cegueira completa, 3% possuem grande dificuldade de enxergar e 15,3% apresentam alguma manifestação de deficiência visual. O número de crianças entre 0 a 14 anos que possuem alguma dificuldade visual representam 5,3% da população total que mesmo não sendo um número tão alto, pode ser agravado se não for diagnosticado precocemente. A pesquisa ainda constata que 20,1% da população entre 15 a 64 anos possuem limitação visual, aumentando para 49,8% a incidência de anomalias visuais para pessoas acima de 65 anos.

Toledo et al. (2010) expõem que muitos pais acabam procurando acompanhamento pedagógico ou até mesmo psicológico para as crianças, afim de lidar com os problemas de aprendizagem. Porém, o problema pode estar relacionado à anomalias visuais, que na maioria dos casos passa despercebido e a criança pode não estar sendo tratada corretamente (TOLEDO et al., 2010). Conforme a Clínica de Olhos Oftalmovillas (2016) retrata, uma pesquisa realizada pelo Ministério da Educação (MEC) apontou que 23% dos casos de evasão escolar estão ligados a deficiências visuais manifestadas de alguma forma nas crianças, mostrando-se uma parcela significativa das causas dos problemas de aprendizagem em escolas. A consequência disso é confirmada pelo estudo do Instituto Penido Burnier em Campinas, que indica que 57% dos estudantes com problemas visuais também tem problemas de desatenção, são agitados e possuem dificuldades de concentração (CLÍNICA DE OLHOS OFTALMOVILLAS, 2016). Já o Conselho Brasileiro de Oftalmologia (CBO) apresenta dados preocupantes, segundo o órgão, 3% a 10% das crianças entre 7 a 10 anos precisam usar óculos e 80% nunca realizaram qualquer exame, conforme exposto por Taleb et al. (2012). Como Rocha et al. (2014) citam em seu artigo, em um estudo sobre a prevalência de doenças oculares realizado em Petrópolis, Rio de Janeiro, com crianças de 0 a 12 anos de idade, o diagnóstico mais frequente (60,9%) foi o erro refrativo, sendo a hipermetropia (56,88%) mais registrada, seguida pelo astigmatismo (35,31%) e miopia (7,81%). Outra condição, que na maioria das vezes é negligenciada ou o portador nem se dá conta de que possui, é a discromopsia, também conhecida com daltonismo que se caracteriza pelo portador não ser capaz de visualizar certas cores, podendo chegar a níveis onde o portador só é capaz de enxergar em preto e branco.

Segundo Soares (2009), os exames de detecção destas anomalias em âmbito escolar brasileiro atualmente são extremamente precários e não há controle algum de tratamento estatístico adequado. Após análise de diversos equipamentos e programas de triagem existentes, Soares (2009) apontou algumas deficiências no processo praticado. Dentre elas estão a ergonomia inadequada tanto das crianças quanto dos profissionais, ausência de padronização da realização dos exames, ausência de equipamento de baixo custo e fácil utilização e, o exame deve ser realizado periodicamente, porém normalmente não é feito. Outros pontos apontados por Soares (2009) são o alto custo para mobilizar os profissionais de saúde para a realizarem os testes de triagem e a ausência de integração dos dados obtidos entre as escolas ou, até mesmo, das secretarias municipais de educação.

Diante deste contexto, este trabalho apresenta o desenvolvimento de uma plataforma que auxilie a triagem do daltonismo e de erros refrativos de baixa ordem, abrangendo miopia e hipermetropia em crianças no ambiente escolar.

1.1 OBJETIVOS

O objetivo deste trabalho é desenvolver uma plataforma móvel de exames para detecção de erros refrativos de baixa ordem e da condição de discromopsia voltado para o uso em escolas.

Os objetivos específicos do trabalho são:

- a) desenvolver um mecanismo capaz de realizar a triagem de problemas de acuidade visual, em especial miopia e hipermetropia, e daltonismo em crianças de idade escolar;
- b) armazenar os dados dos testes realizados, na nuvem, para possibilitar o acompanhamento da evolução das anomalias visuais dos alunos ao longo do tempo.

1.2 ESTRUTURA

Para melhor organização e entendimento do leitor, este trabalho está separado em quatro capítulos. Iniciando o primeiro capítulo com a introdução ao tema do trabalho, seguido do segundo capítulo no qual estão descritos os fundamentos básicos para o entendimento das técnicas e ferramentas utilizadas. O terceiro capítulo apresenta informações sobre a construção do aplicativo como diagramas, arquitetura, comunicação entre cliente e servidores e o código fonte. As últimas seções trazem as telas do aplicativo demonstrando sua operacionalidade, os resultados obtidos após o desenvolvimento e comparação entre os

trabalhos correlatos. No quarto e último capítulo estão às conclusões finais e sugestões para o desenvolvimento de trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo está organizado em três seções. A seção 2.1 aborda os programas de triagem de anomalias visuais. A seção 2.2 descreve o teste de diagnóstico de deficiência na percepção de cores desenvolvido por Shinobu Ishihara. A seção 2.3 expõe o teste de acuidade visual proposto por Herman Snellen. A seção 2.4 mostra o *framework* Phoneygap utilizado para a construção deste trabalho. A seção 2.5 apresenta o *framework* Ionic que possibilitou o desenvolvimento mais ágil deste trabalho. Por fim, a seção 2.6 apresenta os trabalhos correlatos.

2.1 PROGRAMAS DE TRIAGEM DE ANOMALIAS VISUAIS

Segundo Toledo et al. (2010), a detecção precoce de anomalias visuais é uma medida de assistência primária pois cerca de 85% do nosso relacionamento com o mundo exterior é realizado principalmente por meio da visão. Uma pesquisa realizada pelo CBO, em 2004, apontou que 5% das crianças eram cegas de pelo menos um olho e 60% dos casos de cegueira poderiam ser evitados com o tratamento precoce. Ainda, de acordo com a Organização Mundial de Saúde (OMS), todo ano cerca de 500 mil crianças ficam cegas no mundo.

A visão nos bebês não está pronta e não é tão funcional como a visão de um adulto. Na cartilha do Ministério da Saúde (2013), doutora Teller ilustra a progressão do desenvolvimento da visão de uma criança de acordo com a Figura 1.

Figura 1 - Como a criança enxerga



Fonte: Teller (1997).

Segundo Toledo et al. (2010), é no ambiente escolar que as deficiências visuais se tornam proeminentes, resultando em diminuição de rendimento escolar, dores de cabeça para a criança, entre outros problemas. Nesse contexto, surgiram os testes de triagem de

deficiências visuais, cuja implementação em países desenvolvidos tem demonstrado que os custos dessas ações são incomparavelmente menores do que aqueles despendidos a portadores de distúrbios oculares. Já em países em desenvolvimento, por fatores socioeconômicos e culturais, estes testes são negligenciados ou não tem a devida importância pelo Estado (TOLEDO et al., 2010). Dessa forma, as crianças não possuem acesso ao diagnóstico e tratamento adequado. Fato, que pode agravar sua condição e, em casos extremos causando cegueira parcial ou até completa.

Soares (2009) comenta que um dos primeiros programas de saúde na escola do Brasil, foi realizado em 1967, onde vários exames foram realizados em crianças do município de São Paulo. Foi realizada uma triagem visual dos alunos onde se constatou que 12% das crianças tinham acuidade visual diminuída, dentre as quais apenas 40% usavam óculos. Eram testes de triagem não padronizados adequadamente e muitos diagnósticos entre corpo docente e oftalmologistas divergiam muito, pois exames eram feitos de forma inadequada em ambientes com baixa luminosidade e utilizando de métodos que não eram comprovadamente eficazes na detecção da deficiência visual. Atualmente, estes exames não são muito frequentes, mas auxiliam de alguma forma no diagnóstico dos distúrbios visuais conforme pode-se ver na Figura 2, onde os testes são aplicados em uma escola do Piauí.

Figura 2 - Triagem de deficiências visuais em uma escola do Piauí



Fonte: Soares (2009).

Segundo Soares (2009), uma vez realizado o diagnóstico, os alunos são orientados a procurar o auxílio de um médico especializado. Porém, não há controle algum de padronização do processo, não há um controle dos dados obtidos e dessa forma não é possível analisar e compartilhar esses dados de forma precisa para se ter uma idéia mais exata de como essas anomalias visuais atingem a população de estudantes impossibilitando a realização de políticas públicas de prevenção e tratamento dessas condições refrativas quando o diagnóstico for precoce, do contrário, ela pode se agravar onerando ainda mais o Estado, que tem a responsabilidade de prestar assistência médica aos cidadãos.

Na cartilha “Problemas de visão afetam o rendimento escolar” (MINISTÉRIO DA SAÚDE, 2013), foi elaborado um roteiro padronizando quais exames devem ser aplicados em cada período da infância, sendo recomendado para os pais e escolas como forma para realizar o acompanhamento da saúde da criança, conforme pode ser visto no Quadro 1.

Quadro 1 - Roteiro para a promoção da saúde ocular na infância

Roteiro para a promoção da saúde ocular na infância	Pré-natal	0 a 3 anos	3 anos e 1 mês a 5 anos	5 anos e 1 mês a 10 anos	10 anos e 1 mês a menores de 16 anos
Identificação de situações de risco					
Inspeção ocular e anexos					
Profilaxia da oftalmia neonatal					
Rastreamento de retinopatia da prematuridade (ROP)					
Teste do reflexo vermelho (TRV)					
Avaliação funcional					
Acuidade visual					

Fonte: Ministério da Saúde (2013).

Apesar dos esforços do Ministério da Saúde para padronizar os exames, Soares (2009) afirma que há falta de estrutura para realização dos testes nas escolas do país. Não há armazenamento e tratamento estatístico dos resultados obtidos nos testes e os profissionais que realizam os exames, em alguns casos, são submetidos a condições inadequadas para o bom exercício da profissão podendo acarretar problemas ortopédicos (SOARES, 2009). Neste contexto, é possível concluir que o Brasil ainda tem muito a melhorar nesta questão para melhorar os programas de triagens de deficiências visuais nas escolas e por consequência, minimizar a insuficiência escolar relacionada a estas anomalias.

Soares (2009), também menciona que existem diversos métodos para realizar a triagem de anomalias visuais, sendo que um dos exames mais utilizados para a triagem de distúrbios visuais são os de acuidade visual, que medem a capacidade do indivíduo de discernir objetos e visualizar imagens com definição, e o Teste de Ishihara, que verifica se a pessoa possui sensibilidade de percepção de determinadas cores para concluir se há manifestação do daltonismo.

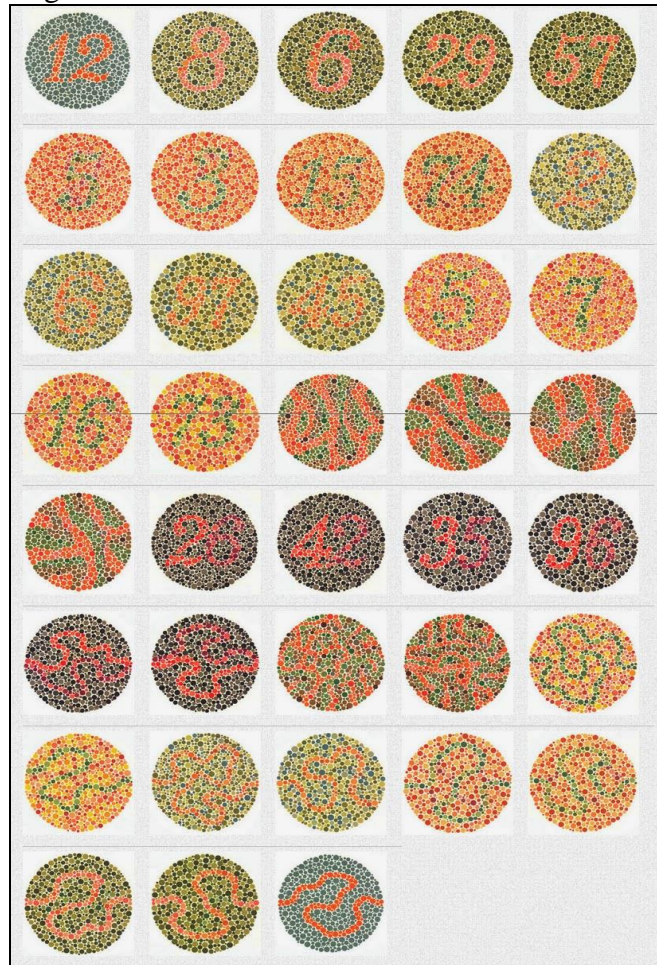
2.2 TESTE DE ISHIHARA

O teste de Ishihara é comumente proposto para identificar se um indivíduo possui daltonismo (SOARES, 2009). Segundo Ishihara (1972), esta doença é a que se manifesta mais frequentemente dentre todas as anomalias visuais relacionadas à detecção de cores.

Segundo Melo et al. (2014), a maioria dos problemas congênitos de visão de cores são caracterizados pela deficiência de vermelho-verde. Ainda de acordo com Melo et al. (2014) a taxa de prevalência desta anomalia é de 6% a 10% entre os homens, sendo que as mulheres podem ser portadoras, mas não manifestam o problema, visto que é um defeito genético relacionado ao cromossomo X, que é relacionado ao sexo masculino.

Ishihara (1972) recomenda que os testes de detecção de daltonismo devam ser realizados em ambientes bem iluminados com a luz do sol. Porém, não deve haver o contato direto da luz do sol com o examinado ou com as lâminas utilizadas nas avaliações. Também ressalta que a luz elétrica não deve ser utilizada pois é possível que resulte em discrepâncias nos resultados. As lâminas utilizadas nos testes, conforme mostra a Figura 3, devem ser mantidas a 75 cm do examinado e este deveria ter de 2 a 3 segundos para indicar o que se pede de cada lâmina, que normalmente se inicia pelas lâminas contendo números. Se o indivíduo não conseguir visualizar os números, outros tipos de lâminas são utilizadas com linhas e outras formas.

Figura 3 - Lâminas utilizadas no Teste de Ishihara



Fonte: Colblindor (2016).

Realizados os testes, os dados são analisados de acordo com a taxa de acerto do examinado conforme *checklist* proposto por Ishihara (1972). O Quadro 2 apresenta o teste para detectar a severidade do daltonismo.

Quadro 2 - Teste para detectar a severidade do daltonismo

Plate	Normal	Person with Red-Green Deficiencies				Person with Total Colour Blindness
1	12	12				12
2	8	3				X
3	29	70				X
4	5	2				X
5	3	5				X
6	15	17				X
7	74	21				X
8	6	X				X
9	45	X				X
10	5	X				X
11	7	X				X
12	16	X				X
13	73	X				X
14	X	5				X
15	X	45				X
		Protan		Deutan		
		Strong	Mild	Strong	Mild	
16	26	6	(2)6	2	2(6)	
17	42	2	(4)2	4	4(2)	

Fonte: adaptado de Ishihara (1972).

De acordo com o Quadro 2 acima proposto por Ishihara (1972), a coluna *plate* denota o número da lâmina utilizada no teste. Para quase todas as lâminas, com exceção da 14 e 15, há um número visível por pessoas consideradas normais. A terceira coluna ilustra o número que uma pessoa com daltonismo do tipo verde-vermelho conseguiria ver nas lâminas e a última coluna mostra como um indivíduo com daltonismo total interpretaria a imagem da lâmina. As colunas *protan* e *deutan* estão relacionadas à Deuteranopia e Protanopia, que correspondem respectivamente à incapacidade de visualizar a cor verde e vermelho. Desta forma, a partir do Quadro 2 é possível concluir que uma pessoa com deficiência na detecção do verde-vermelho, tipo de daltonismo mais comum, consegue interpretar um número, ainda que erroneamente. Já uma pessoa com daltonismo severo não consegue visualizar número qualquer na grande maioria dos casos o que também pode trazer grandes empecilhos no cotidiano da pessoa com esta condição.

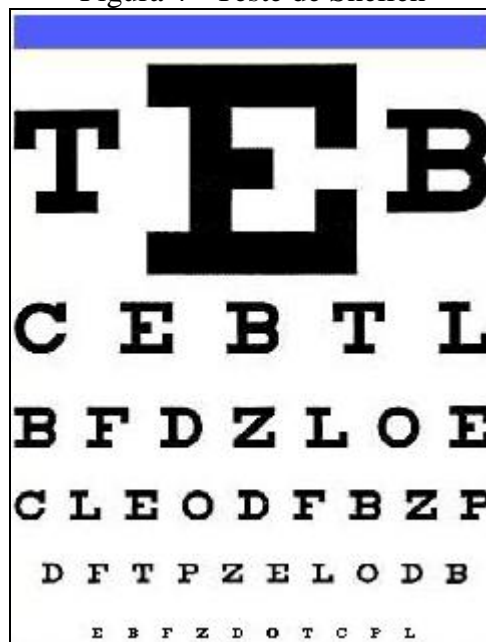
2.3 TESTE DE ACUIDADE VISUAL

Como Bicas (2002) define acuidade visual em seu trabalho:

Refere-se acuidade visual como a função (visual) que exprime a capacidade discriminativa de formas; ou como o método com que se mede o reconhecimento da separação angular entre dois pontos no espaço (isto é, distância entre eles, relacionada ao primeiro ponto nodal do olho); ou da resolução (visual) de suas respectivas imagens sobre a retina, relacionadas ao segundo ponto nodal do olho (BICAS, 2002, p.376).

Zaparolli et al. (2009) citam que os dados históricos mais antigos encontrados sobre a medida da visão mostram que em 1843, o oftalmologista alemão Kuechler desenvolveu três tabela de medida para testes de acuidade visual. Porém, seu trabalho quase foi esquecido por completo. Jaeger, em 1854, publicou em Viena uma tabela de leitura para documentar a visão, sendo utilizada até hoje por muitas pessoas (ZAPAROLLI et al., 2009). Em 1862, o oftalmologista holandês Herman Snellen, com ajuda de outro especialista Donders, publicou sua famosa tabela baseada e definida em “optótipos” (ZAPAROLLI et al., 2009). A tabela, ilustrada na Figura 4, consiste em linhas de letras em que o examinado exposto frontalmente a ela tenta fazer a leitura para ver se consegue definir o que está escrito, caso não seja visto alguma linha, ela deve ser anotada e define o grau de acuidade visual do indivíduo. Este teste deve ser feito visualizando a imagem com os 2 olhos, depois tapando o esquerdo e depois tapando apenas o direito, pois cada olho pode manifestar um grau diferente de deficiência.

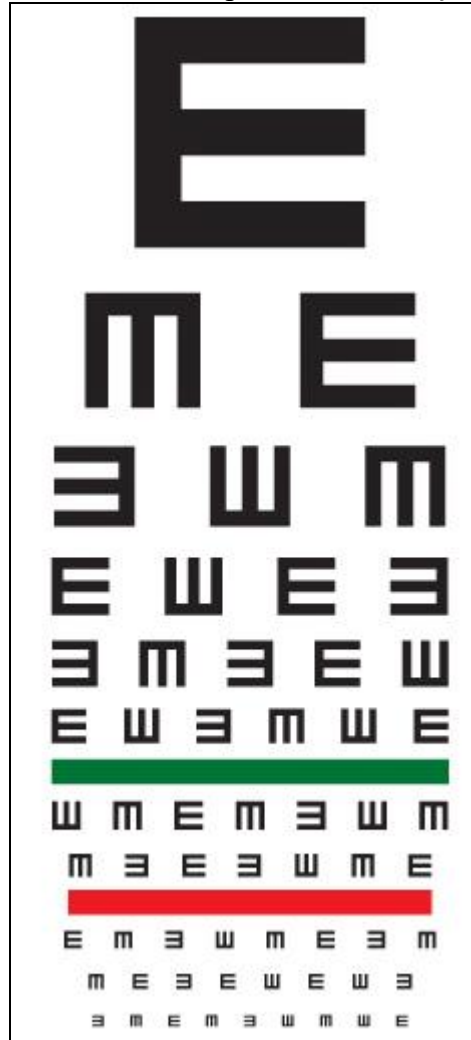
Figura 4 - Teste de Snellen



Fonte: Hospital Oftalmológico Visão Laser (2016).

Com o passar dos anos, a tabela vem sendo aprimorada e hoje pode ser encontrada de diversas formas. A Figura 5 ilustra um exemplo de Teste de Snellen, onde o indivíduo deve apontar para qual lado a letra “E” está apontando.

Figura 5 - Tabela de Snellen para indicar direção da letra “E”



Fonte: Soares (2009).

A partir dos resultados do teste é possível diagnosticar o grau da severidade da anomalia visual. Caso seja constatada a condição anormal da visão, deve-se encaminhar o examinado para o devido tratamento ou o uso de lentes corretivas para que o seu problema não afete o seu cotidiano ou o seu rendimento escolar.

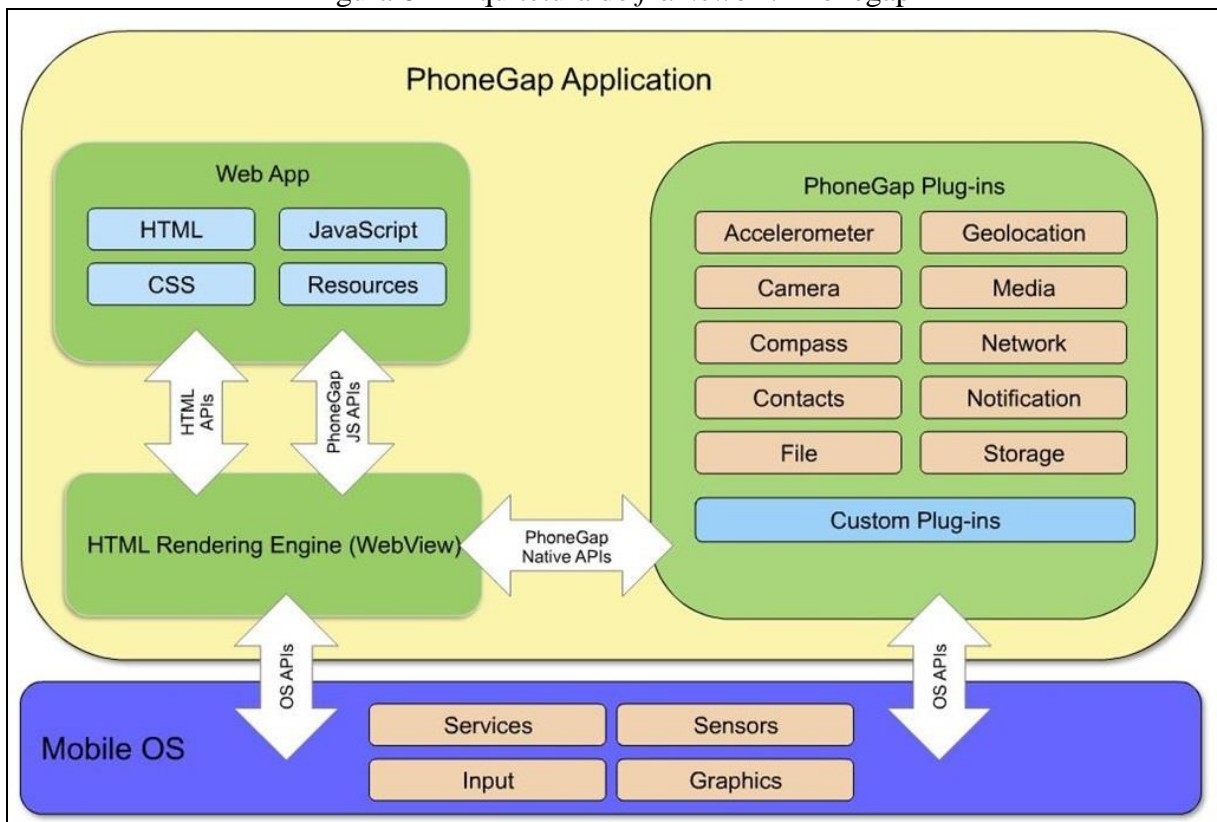
2.4 PHONEGAP

O Phonegap é um *framework* de desenvolvimento de aplicativos móveis concebido para solucionar um problema de reescrever a mesma aplicação para vários sistemas operacionais móveis. Muitos distribuidores de aplicações móveis acabam gerando aplicativos para esses sistemas operacionais diferentes a fim de aumentar sua abrangência no mercado acabam tendo

que investir valores extras para realizar a portabilidade. O Phonegap veio para sanar esta questão e simplificar o desenvolvimento, assim, apenas desenvolvendo um código-fonte em HTML, CSS e JavaScript, como se fosse uma página *web*, pode-se gerar aplicações instaláveis nos principais sistemas operacionais do mercado e facilmente reutilizar o código-fonte para utilizar em websites (TRICE, 2012).

A arquitetura do *framework* é exposta na Figura 6, onde o desenvolvedor constrói toda a sua aplicação como se fosse uma página web. O próprio Phonegap se encarrega de converter esse código-fonte para as *views* de uma aplicação móvel. O programador tem acesso aos *plugins* via chamadas da API nativa do *framework* que por sua vez acessa os recursos nativos do sistema operacional. Uma vantagem, além das expostas anteriormente, é a possibilidade de realizar o *deploy* da aplicação para depuração de forma simples e rápida para depuração em tempo real sem necessitar de ferramentas ou ambientes de desenvolvimento adicionais.

Figura 6 - Arquitetura do *framework* Phonegap



Fonte: Magni (2016).

Magni (2016) destaca algumas das desvantagens deste *framework* como a possível perda de performance da aplicação e ausência das interfaces de *widgets*. Considerando que a ideia de uso do Phonegap é construir um código-fonte e replicar pra várias plataformas, a aparência visual do aplicativo é idêntica em todos os sistemas operacionais, visto que Android e iOS possuem seus padrões de interface por exemplo. Outra desvantagem é o tamanho do

aplicativo gerado pelo *framework* ser consideravelmente maior do que se fosse construído desenvolvendo a mesma aplicação em código-fonte nativo.

O Phonegap é *open-source* e continua com atualizações frequentes e conta com uma comunidade bastante numerosa e ativa (MAGNI, 2016). Suporta os sistemas operacionais iOS, Android, Windows 8, Windows Phone 8, BlackBerry 10, Ubuntu, entre outros. A lista de compatibilidade de recursos do *framework* é ilustrado no Quadro 3.

Quadro 3 - Suporte de plataformas do Phonegap

	amazon- fireos	android	blackberry10	ios	Ubuntu	wp7 (Windows Phone 7)	wp8 (Windows Phone 8)	win8 (Windows 8)	tizen
cordova CLI	✓ Mac, Windows, Linux	✓ Mac, Windows, Linux	✓ Mac, Windows	✓ Mac	✓ Ubuntu	✓ Windows	✓ Windows	✓	X
Embedded WebView	✓ (see details)	✓ (see details)	X	✓ (see details)	✓	X	X	X	X
Plug-in Interface	✓ (see details)	✓ (see details)	✓ (see details)	✓ (see details)	✓	✓ (see details)	✓ (see details)	✓	X
Platform APIs									
Accelerometer	✓	✓	✓	✓	✓	✓	✓	✓	✓
Camera	✓	✓	✓	✓	✓	✓	✓	✓	✓
Capture	✓	✓	✓	✓	✓	✓	✓	X	X
Compass	✓	✓	✓	✓ (3GS+)	✓	✓	✓	✓	✓
Connection	✓	✓	✓	✓	✓	✓	✓	✓	✓
Contacts	✓	✓	✓	✓	✓	✓	✓	X	X
Device	✓	✓	✓	✓	✓	✓	✓	✓	✓
Events	✓	✓	✓	✓	✓	✓	✓	✓	✓
File	✓	✓	✓	✓	✓	✓	✓	✓	X
Geolocation	✓	✓	✓	✓	✓	✓	✓	✓	✓
Globalization	✓	✓	X	✓	✓	✓	✓	X	X
InAppBrowser	✓	✓	✓	✓	✓	✓	✓	(partial)uses iframe	X
Media	✓	✓	✓	✓	✓	✓	✓	✓	✓
Notification	✓	✓	✓	✓	✓	✓	✓	✓	✓
Splashscreen	✓	✓	✓	✓	✓	✓	✓	✓	X
Storage	✓	✓	✓	✓	✓	✓ localStorage & indexedDB	✓ localStorage & indexedDB	✓ localStorage & indexedDB	✓

Fonte: Phonegap Docs (2016).

Como pode ser visto no Quadro 3, o Phonegap suporta grande parte dos principais recursos nativos dos mais populares sistemas operacionais para dispositivos móveis do mercado atualmente. Algumas exceções percebidas como por exemplo o recurso de globalização no BlackBerry 10 e Windows 8. O Tizen, que é um sistema operacional *open-source* baseado no *kernel* do Linux feito em parceria com a Samsung (TIZEN, 2016), apresenta a menor compatibilidade com o Phonegap e quase todas as plataformas com exceção do Amazon FireOS, Android, iOS e Ubuntu não possuem a funcionalidade de

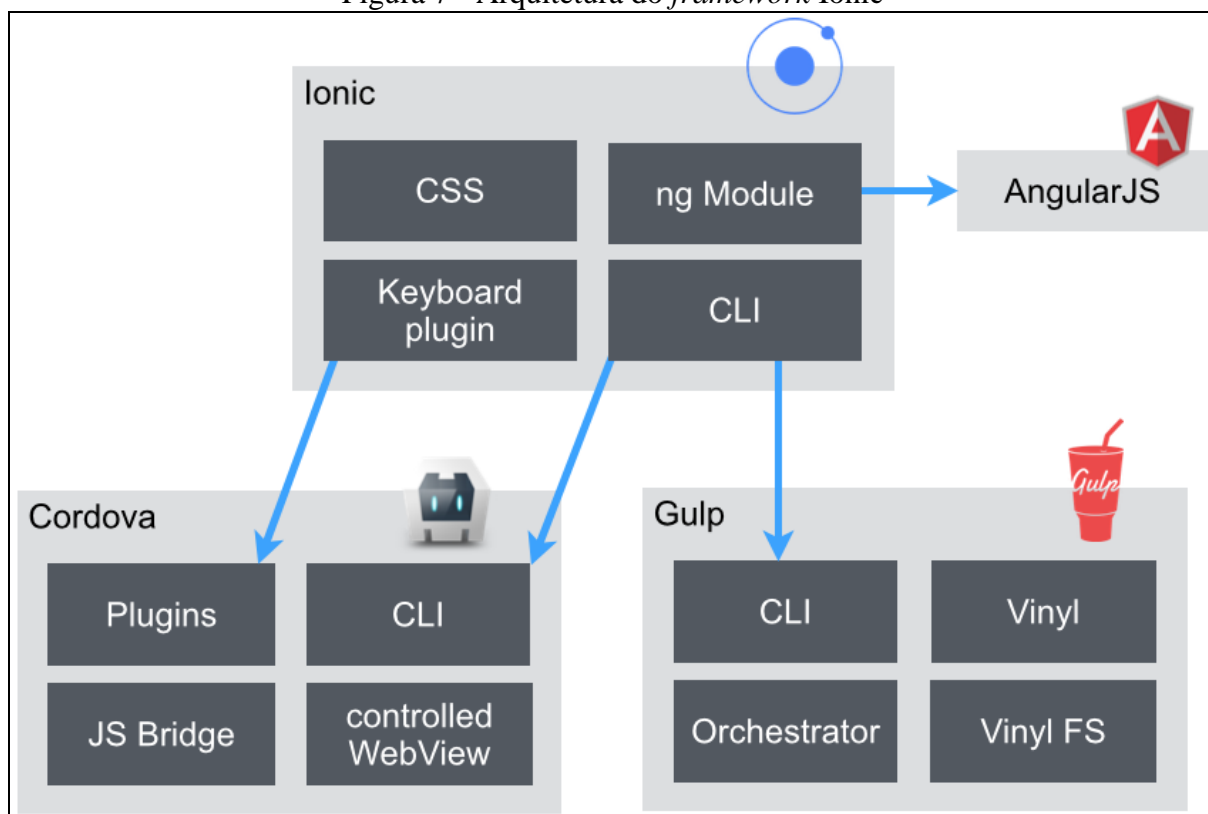
apresentar o conteúdo da aplicação como se fosse uma visão de página web embarcada do dispositivo.

2.5 IONIC

O Ionic é um *framework* construído sobre Phonegap com o propósito de facilitar e acelerar o desenvolvimento de aplicações móveis que proporciona ferramentas que auxiliam a distribuição e escalonamento de aplicativos (IONIC, 2016).

Um dos benefícios do uso do Ionic é a integração realizada entre este *framework* com o AngularJS, oferecendo recursos amplamente utilizados para desenvolvimento de websites que já conta com uma grande comunidade de desenvolvedores. Outros pontos positivos são a grande quantidade de componentes pré-construídos, inclusive oferecendo itens nativos dos sistemas operacionais Android e iOS. O *framework* conta também com uma funcionalidade denominada LiveReload, onde o desenvolvedor realiza alterações no código-fonte e já pode visualizar o aplicativo atualizado instantaneamente (IONIC, 2016).

A arquitetura do *framework* é ilustrada na Figura 7. A lógica é desenvolvida utilizando JavaScript e AngularJS como mostra o bloco `ng module`. Os recursos nativos do dispositivo móvel podem ser acessadas via Cordova ou Gulp conforme os blocos `keyboard plugin` e `cli`.

Figura 7 - Arquitetura do *framework* Ionic

Fonte: Codecentric (2016).

Assim como o Phonegap, o Ionic também é um *framework open-source*, sendo uma das principais opções hoje disponíveis para desenvolvimento de aplicações híbridas para dispositivos móveis. Por fim, o Ionic oferece mais de setenta funcionalidades nativas dos smartphones herdadas do uso do Phonegap como *bluetooth*, autenticação por digital, entre outras. Atualmente, o Ionic já ganhou uma segunda versão com mais funcionalidades porém não foi utilizada neste trabalho pois não houve necessidade e como ainda é nova, pode existir comportamentos inesperados, assim foi optado por desenvolver utilizando a primeira versão (IONIC, 2016).

2.6 TRABALHOS CORRELATOS

A seguir estão relacionados três trabalhos correlatos ao proposto. A seção 2.6.1 descreve uma proposta de aplicativo móvel *open source* para auxiliar no diagnóstico de indivíduos com daltonismo (PAULA JÚNIOR, 2014). Na seção 2.6.2 é apresentado um sistema de triagem visual e auditiva para crianças em idade escolar conectado a um banco de dados (SOARES, 2009). A seção 2.6.3 detalha o *Eye Test – Eye Exam* (HEALTHCARE4MOBILE, 2016), aplicativo móvel desenvolvido pela *healthcare4mobile* para diagnóstico de deficiências visuais. Ao final, a seção 2.6.4 mostra o aplicativo para

dispositivos móveis Peek Vision para diagnóstico de anomalias visuais (PEEK VISION, 2016).

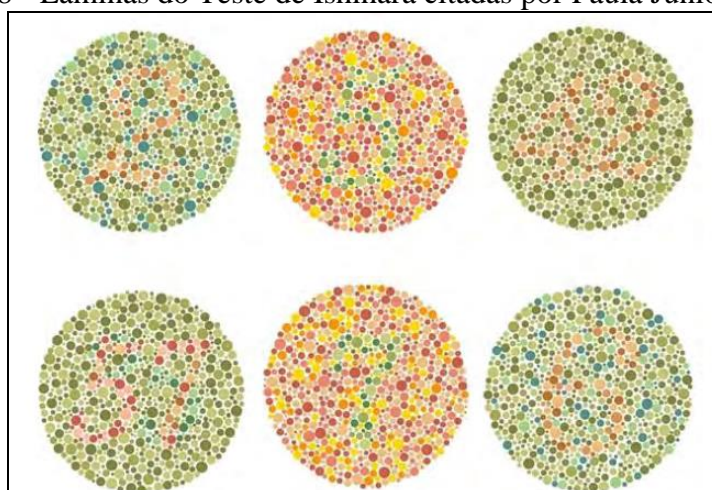
2.6.1 Proposta de um aplicativo móvel *Open-Source* em auxílio a indivíduos com discromopsia baseado em um estudo qualitativo

Paula Júnior (2014) propõe uma idéia de aplicativo móvel de código aberto e escalável com sistemas operacionais Android e iOS que, através de estudos e análises computacionais, permite diagnosticar rapidamente e automaticamente, o daltonismo dos usuários. Visto que não há ferramentas disponíveis gratuitamente que reúnam todas essas características no mercado.

Para atingir o seu objetivo, Paula Júnior (2014) primeiramente realizou pesquisas de problemas que afetam a visão para entender o contexto. A partir de suas pesquisas ele percebeu que tal condição atinge cerca de 10% dos homens e 1% das mulheres e é relacionada ao código genético ocasionado a genes recessivos ligados ao cromossomo X, que geralmente define a masculinidade.

Segundo Paula Júnior (2014), uma das formas de detectar a discromopsia é o Teste de Ishihara, que consiste em levar a pessoa a visualizar imagens compostas de pequenos pontos coloridos que, por contraste e diferença de cores, é possível enxergar um número, concordância do contrário, o indivíduo é considerado daltônico, conforme pode ser visto na Figura 8.

Figura 8 - Lâminas do Teste de Ishihara citadas por Paula Júnior (2014)



Fonte: Paula Júnior (2014).

A partir de sua pesquisa, Paula Júnior (2014) começou a fornecer questionários a pessoas sabidamente diagnosticadas com daltonismo. Seu objetivo é coletar informações de como este problema genético afetava seus cotidianos e também percebeu que há diferentes

níveis de gravidade deste distúrbio. Posteriormente, ele levou as informações à clínicas oftalmológicas afim de mapear os problemas relacionados e a existência de aplicativos relacionados. Foram analisados aplicativos relacionados ao daltonismo nas plataformas Android e iOS e, desta forma, foi possível comparar as funcionalidades disponíveis afim de mapear o que há no mercado e o que é carente, como mostra os quadros Quadro 5 e Quadro 6, sendo que as legendas destes quadros consta no Quadro 4. O autor ressalta que alguns dados não puderam ser elencados pois alguns aplicativos eram pagos e/ou possuem documentação insuficiente.

Quadro 4 - Siglas correspondentes às funcionalidades dos sistemas

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

Fonte: Paula Júnior (2014).

Quadro 5 - Comparativo dos aplicativos *Android* relacionados ao daltonismo

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

Fonte: Paula Júnior (2014).

Quadro 6 - Comparativo dos aplicativos *iOS* relacionados ao daltonismo

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

Fonte: Paula Júnior (2014).

Com essas informações, Paula Júnior (2014) cita em seu trabalho o planejamento do seu software. Porém, o aplicativo está em fase de concepção e por isso não foi possível levantar seus resultados.

2.6.2 Sistema de triagem visual e auditiva de crianças em idade escolar, conectado a um banco de dados

O trabalho de Soares (2009) tem como objetivo o desenvolvimento de uma ferramenta capaz de realizar a triagem visual e auditiva e, que possa ser aplicado em programas públicos. Diferentemente de outras formas de testes, esta ferramenta é uma alternativa que oferecesse recursos adicionais como armazenamento de dados e um tratamento estatístico para as avaliações realizadas em grande escala.

O sistema foi construído de maneira que comportasse subsistemas cada qual com sua função específica e que possibilitasse interessados a adicionar novas funcionalidades. Inicialmente foi implementada a função de triagem de acuidade visual que fez uso de um notebook para realizar o teste. Nela, o indivíduo deve indicar que letras ele está vendo em qual direção, conforme pode ser visto na Figura 9.

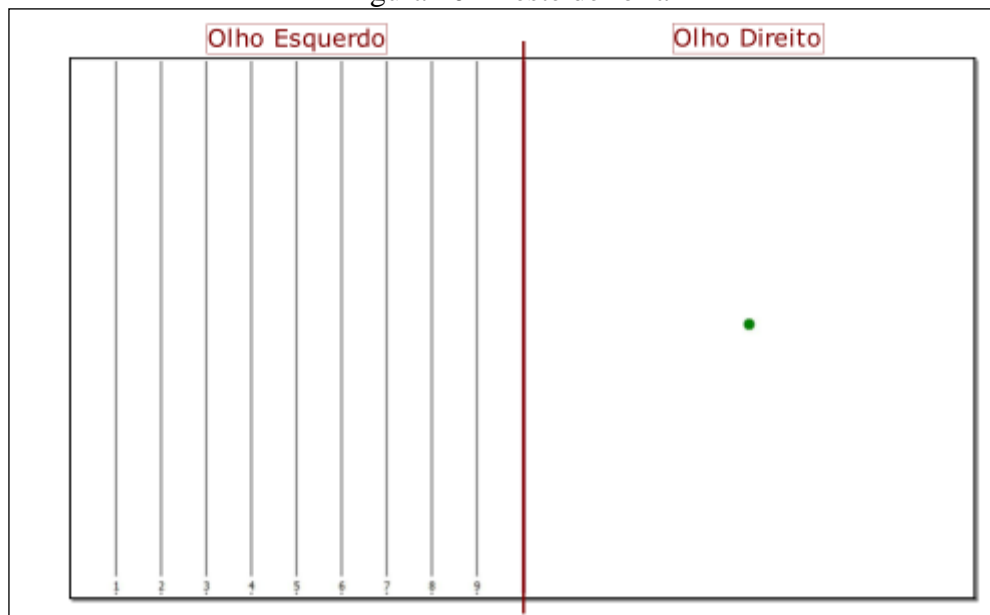
Figura 9 - Teste de acuidade visual



Fonte: Soares (2009).

Soares (2009), relata também o teste de sensibilidade ao contraste e de visão de cores, utilizados para detectar o daltonismo onde são utilizadas técnicas conforme Teste de Ishihara. Também foi disponibilizado o teste de foria, que consiste em outro exame visual onde o indivíduo visualiza a imagem apresentada na Figura 10. Ela é composta de retas horizontais ou verticais e um ponto, onde o exame consiste em perguntar ao examinado “Qual a linha mais próxima da bolinha”, de acordo com o número apresentado identifica-se a possível perda de alinhamento ocular.

Figura 10 - Teste de foria



Fonte: Soares (2009).

Abordando a audição, também é disponibilizada uma triagem do limiar auditivo. Onde, é emitido um sons com frequências de 250 Hz, 500 Hz, 1 kHz, 2 kHz, 3 kHz, 4 kHz, 6 kHz e 8kHz no lado esquerdo e direito separadamente. Se os sons com pressão acima de 25 cB não forem percebidos, considera-se que o indivíduo possui perda auditiva. E, por último, há a medição do desempenho de leitura, onde é necessário tanto um hardware especial como um aplicativo de comunicação com o banco de dados. A aferição do desempenho de leitura é realizada medindo o posicionamento do olho a cada instante, com essa finalidade, são utilizados emissores e receptores de luz infra-vermelha instalados em óculos especiais. A luz infra-vermelha emitida é refletida pelo olho e chega ao receptor de infra-vermelho e a intensidade dessa luz varia com a posição do olho, sendo possível mensurar o desempenho do indivíduo na leitura.

A integração dos subsistemas foi realizada em arquitetura cliente-servidor em uma aplicação *desktop*. A aplicação de Soares (2009) conta com funcionalidades de cadastro dos

examinados e comunicação com banco de dados guardando as informações coletadas para posterior análise.

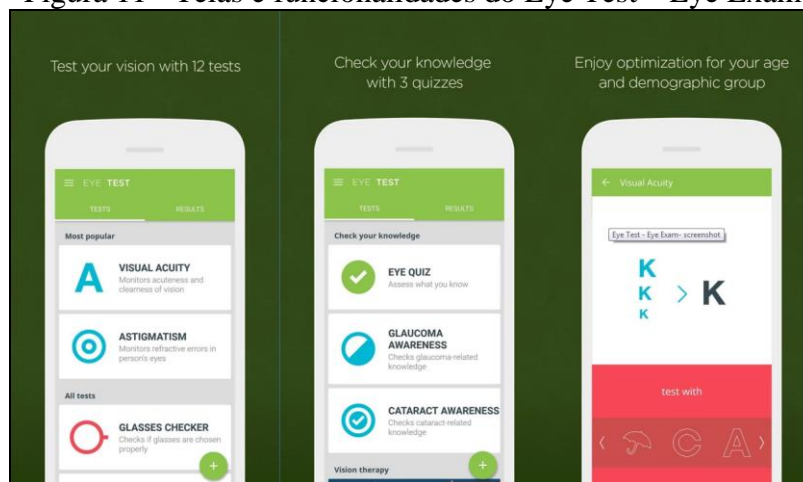
Não é comentado de forma exata os resultados do trabalho. Porém, o autor comenta que atingiu o objetivo, que basicamente era montar uma plataforma acoplável de funcionalidades que permitissem ter exames tanto visuais quanto auditivos, e que fosse barato e fácil de usar.

2.6.3 Eye Test – Eye Exam

O Eye Test – Eye Exam (HEALTHCARE4MOBILE, 2016) é um aplicativo móvel que tem por finalidade disponibilizar testes de visão para o usuário, somando 12 ao todo. Além de trazer informações adicionais de como funciona outras condições visuais como o glaucoma e catarata. O aplicativo também tem um quiz onde o indivíduo testa os seus conhecimentos sobre o olho, assim como disponibiliza informações para desmistificar o que realmente é prejudicial à visão ou não (HEALTHCARE4MOBILE, 2016).

O aplicativo também disponibiliza várias outras aplicações Android relacionadas à testes visuais ou prática de terapias visuais para a manutenção da saúde da visão, conforme mostra a principal tela do aplicativo apresentado na Figura 11.

Figura 11 - Telas e funcionalidades do Eye Test – Eye Exam



Fonte: Healthcare4mobile (2016).

Como pode ser visto na Figura 11, a aplicação possui uma interface amigável e de fácil uso. Ela traz exames de acuidade visual, astigmatismo, verificação se óculos estão apropriados para o usuário, olho seco, acomodação, daltonismo, deficiência visual de cores, teste de sensibilidade do contraste, teste do campo central de visão da pessoa, catarata, miopia e hipermetropia. O aplicativo retorna os resultados assim que os testes são finalizados,

fazendo um diagnóstico rápido com base no que o usuário respondeu nos questionamentos relacionados às imagens e testes propostos.

O time de desenvolvimento deixa claro que os exames realizados via dispositivo Android não substitui o diagnóstico feito por um médico especialista. Porém, sua intenção é proporcionar uma forma de diagnóstico simples para detectar previamente problemas que poderiam passar despercebidos caso não fosse dada a devida atenção.

2.6.4 Peek Vision

O Peek Vision é um aplicativo para dispositivos móveis que tem por finalidade executar testes de visão em diversas populações de indivíduos, sendo uma solução bem completa que possibilita também guardar os dados dos testes e realizar a análise sobre os dados obtidos.

O aplicativo conta com uma vasta gama de testes como teste de acuidade visual, teste de contraste de cores, teste de daltonismo e testes que envolvem imagens do fundo da retina que podem verificar diabetes e pressão alta sanguínea com o auxílio de um hardware acoplado ao smartphone com a função de um oftalmoscópio, conforme mostra a Figura 12 (PEEK VISION, 2016).

Figura 12 - Oftalmoscópio utilizado no aplicativo Peek Vision



Fonte: Peek Vision (2016).

O aplicativo foi construído com um time de desenvolvedores em conjunto com profissionais médicos da área. Possuem a missão de fortalecer os sistemas de saúde e dar mais poder aos indivíduos criando ferramentas e inteligência na área da saúde para aumentar o acesso ao diagnóstico de anomalias visuais em escala mundial. Seu uso é mais frequente em

populações sem muitas condições de acesso ao diagnóstico, principalmente África e Sudeste da Ásia conforme Figura 13 e Figura 14 (PEEK VISION, 2016).

Figura 13 - Mapa de localizações da aplicação dos testes do Peek Vision



Fonte: Peek Vision (2016).

Figura 14 - Triagem de anomalias visuais em escola do Condado de Trans Nzoia no Quênia



Fonte: Peek Vision (2016).

Além dos testes, o aplicativo também conta com programas de triagem de anomalias visuais em escolas, triagem de retinopatia diabética em pacientes e triagem em comunidades (PEEK VISION, 2016). Ele possibilita a análise dos dados e pode até recomendar o encaminhamento do indivíduo a um profissional adequado. Visando a colaboração mundial, é possível aos profissionais que aplicam os testes os pacientes, cadastrar as informações da

pessoa e o diagnóstico conclusivo, a fim de melhorar a base de análises do aplicativo para um diagnóstico mais assertivo e permitir que se possa acompanhar os resultados dos testes.

2.6.5 Comparativo e discussão das características dos trabalhos correlatos

A partir das informações elencadas com os trabalhos descritos nas seções anteriores, foi montado o Quadro 7 que apresenta um comparativo com as principais características dos trabalhos correlatos.

Quadro 7 - Comparação com os trabalhos correlatos

Características / trabalhos	Paula Júnior (2014)	Soares (2009)	Healthcare4 mobile (2016)	Peek Vision (2016)
Portável para dispositivos móveis	Sim	Não	Sim	Sim
Exame de acuidade visual	Não	Sim	Sim	Sim
Exame de daltonismo	Sim	Sim	Sim	Sim
Exame de glaucoma	Não	Não	Sim	Não
Exame de catarata	Não	Não	Sim	Não
Exame de retinopatia diabética	Não	Não	Não	Sim
Exame de pressão sanguínea via imagem de fundo do olho	Não	Não	Não	Sim
Público alvo	Crianças em idade escolar	Crianças em idade escolar	Todos	Todos
Relatórios detalhados de cada teste e evolução	Não	Não	Não	Sim
Encaminhamento automático para o profissional adequado	Não	Não	Não	Sim

Fonte: elaborado pelo autor.

A partir do Quadro 7, conclui-se que apenas o trabalho de Soares (2009) não possui portabilidade para dispositivos móveis. Já Paula Júnior (2014) aborda apenas o daltonismo, deixando de lado a questão da acuidade visual. Percebe-se que todos os trabalhos, com exceção do aplicativo Eye Test – Eye Exam (HEALTHCARE4MOBILE, 2016), tem um apelo mais social, visando auxiliar na detecção precoce de anomalias visuais nas escolas. O aplicativo de Healthcare4mobile (2016) é o único que possui testes de glaucoma e catarata. O Peek Vision é o aplicativo que mais se assemelha com a proposta em termos de testes e contribuição para a sociedade. Apenas este possui testes de retinopatia diabética e pressão sanguínea via imagens do fundo de olho. Ainda possui uma série de integrações com outros sistemas que possibilita o encaminhamento automático de pacientes com potencial diagnóstico de anomalia visual para os profissionais mais adequados.

3 DESENVOLVIMENTO

Este capítulo aborda os aspectos referentes à construção da plataforma. Na seção 3.1 encontram-se os requisitos funcionais e não funcionais. Na seção 3.2 são apresentados os diagramas de casos de uso, classes e atividades, especificando e detalhando o funcionamento do aplicativo. Ainda nesta seção, são apresentados os principais serviços que integram a Iris, mostrando a comunicação entre todos os componentes. Os principais trechos de código e utilização do aplicativo também são explicados nesta seção. Para encerrar o capítulo, a seção 3.4 expõe os resultados obtidos após o desenvolvimento do aplicativo e realiza a comparação entre trabalhos correlatos apresentados na seção 2.4.

3.1 REQUISITOS

Os requisitos funcionais e não funcionais da plataforma estão descritos conforme a lista abaixo:

- a) permitir que o usuário autentique seu usuário para ter acesso à plataforma (Requisito Funcional - RF);
- b) permitir o cadastro de novos usuários (RF);
- c) permitir que o usuário logado na plataforma possa alterar a sua senha (RF);
- d) ter um usuário administrador que tenha acesso à todas as funcionalidades da plataforma inclusive cadastrar novos usuários ou removê-los (RF);
- e) permitir o administrador cadastrar novos testes (RF);
- f) permitir que todos os usuários possam visualizar todos os testes cadastrados e suas respectivas questões e alternativas (RF);
- g) permitir o administrador remover testes, porém esta ação só pode ser realizada quando o teste não foi aplicado a nenhum aluno, caso contrário não deve permitir a remoção do mesmo (RF);
- h) permitir a todos os usuários cadastrarem novos alunos (RF);
- i) permitir aplicar um dos testes cadastrados a um aluno cadastrado na plataforma (RF);
- j) disponibilizar um relatório do resultado de um aluno em determinado teste e em determinada data (RF);
- k) disponibilizar um relatório de todos os resultados de um teste (RF);
- l) disponibilizar um relatório que mostre a evolução de determinado aluno em determinado teste (RF);
- m) ter controle de permissão onde o administrador tem acesso à todas as

funcionalidades e todos os outros usuários não podem cadastrar/alterar/remover testes e/ou usuários (RF);

- n) utilizar o *framework* Ionic (Requisito Não Funcional – RNF);
- o) deve ser desenvolvida no ambiente de desenvolvimento integrado Eclipse e o editor de textos Sublime Text (RNF);
- p) o servidor back-end deve ser desenvolvido utilizando a linguagem de programação Java (RNF) e utilizar a Web Api Jersey Restful;
- q) o cliente front-end deve ser desenvolvido utilizando as tecnologias web HTML, JavaScript, CSS (RNF);
- r) o front-end deve se comunicar com o servidor back-end por requisições web service Ajax do tipo REST utilizando a biblioteca AngularJS (RNF) enviando objetos em formato JSON;
- s) para o mapeamento de entidades e acesso a dados no back-end, deve-se utilizar a ferramenta MyBatis;
- t) utilizar o sistema de gerenciamento de banco de dados Oracle MySql (RNF);
- u) disponibilizar para dispositivos móveis que utilizem sistema operacional Android (RNF);
- v) funcionar apenas com acesso à internet ou ao menos na mesma rede na qual o servidor se encontra (RNF).

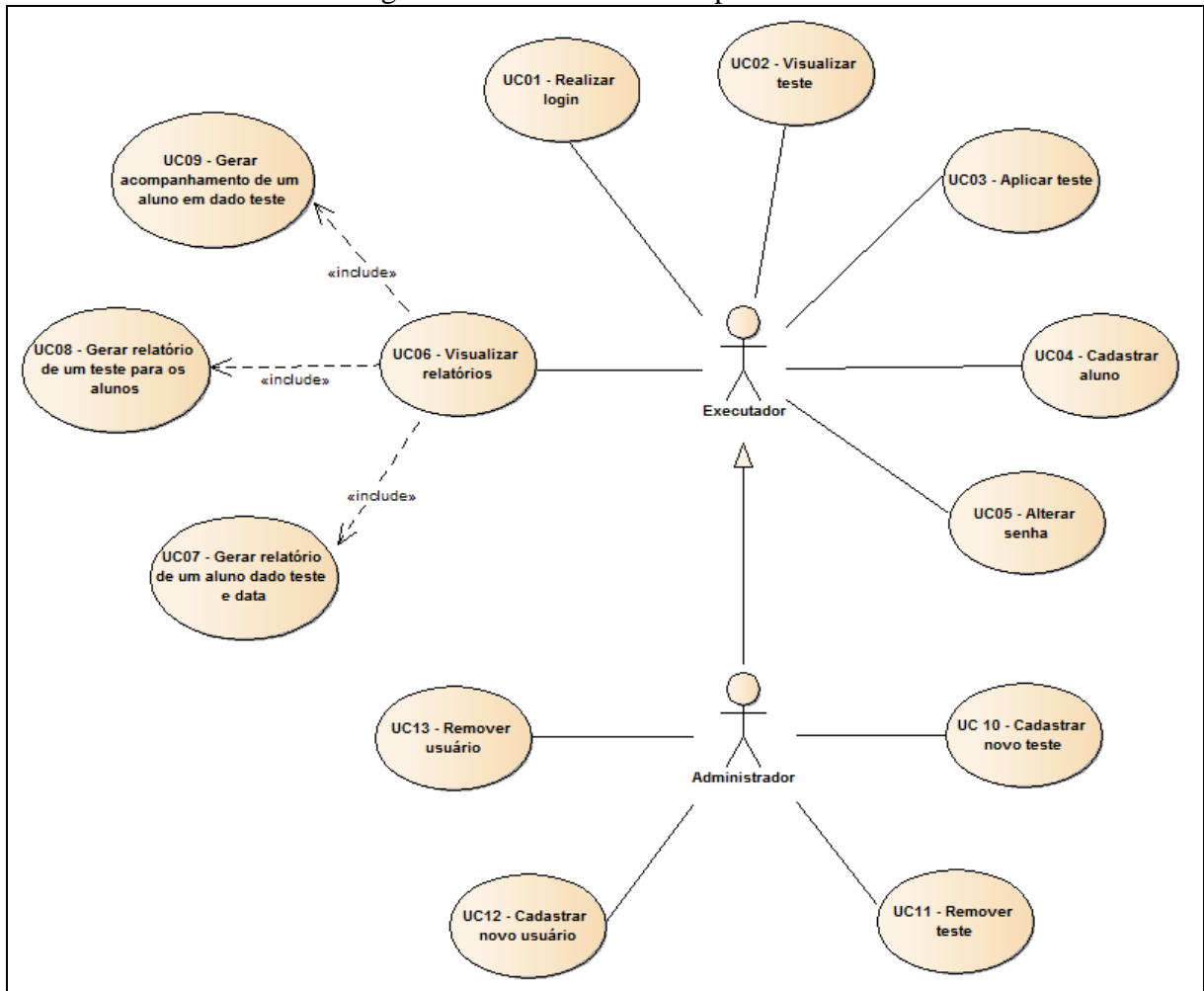
3.2 ESPECIFICAÇÃO

Para o entendimento das funcionalidades e arquitetura da plataforma, esta seção apresenta os diagramas de casos de uso, classes e atividades, elaborados com as ferramentas Enterprise Architect e Draw.io. Ao final desta seção é apresentada a arquitetura de comunicação entre a plataforma e o servidor.

3.2.1 Diagrama de casos de uso

A partir do levantamento dos requisitos da plataforma, foi desenvolvido o diagrama de casos de uso conforme ilustrado na Figura 15, que representam as funcionalidades disponíveis para os atores *Executador* e *Administrador*.

Figura 15 - Casos de uso da plataforma



Fonte: elaborado pelo autor.

Iniciado pelo Executador dos testes, o caso de uso UC01 - Realizar login o usuário deverá se autenticar na plataforma com nome do usuário e senha cadastrados, apenas assim terá acesso. No caso de uso UC02 - Visualizar teste é disponibilizada a visualização dos testes cadastrados pelo Administrador da plataforma. O caso de uso UC03 - Aplicar teste permite a todos os usuários da plataforma aplicar um teste a um aluno cadastrado e salvar os resultados ao final do processo. No caso de uso UC04 - Cadastrar aluno permite aos usuários cadastrarem novos alunos na plataforma. O caso de uso UC05 - Alterar senha é permitido ao usuário a todos os usuários alterar sua própria senha. No caso de uso UC06 - Visualizar relatórios é possível ter acesso a todos os relatórios disponibilizados. O caso de uso UC07 - Gerar relatório de um aluno dado teste e data permite a todos os usuários visualizar a performance de um aluno em um teste e data específica. No caso de uso UC08 - Gerar relatório de um teste para os alunos é possível conferir todos os resultados dos alunos em dado teste. Por fim, no caso de uso UC09

- Gerar acompanhamento de um aluno em dado teste possibilita que o usuário visualize a evolução do aluno em várias execuções do mesmo teste.

Em seguida, há o usuário `Administrador` que herda todas as funcionalidades do `Executador` e possui permissões para realizar algumas ações adicionais. O caso de uso UC10 - `Cadastrar novo teste` trata de cadastrar testes na plataforma, não necessariamente apenas testes de diagnóstico de anomalias visuais. O caso de uso UC11 - `Remover teste` permite ao `Administrador` excluir o teste selecionado, porém esta ação somente pode ser realizada se o teste não foi aplicado. No caso de uso UC12 - `Cadastrar novo usuário` é possível adicionar executadores de teste na plataforma que são distribuídos aos seus respectivos usuários. Por fim, o caso de uso UC13 - `Remover usuário` é responsável pela exclusão de usuários da plataforma restringindo o acesso caso julgue necessário.

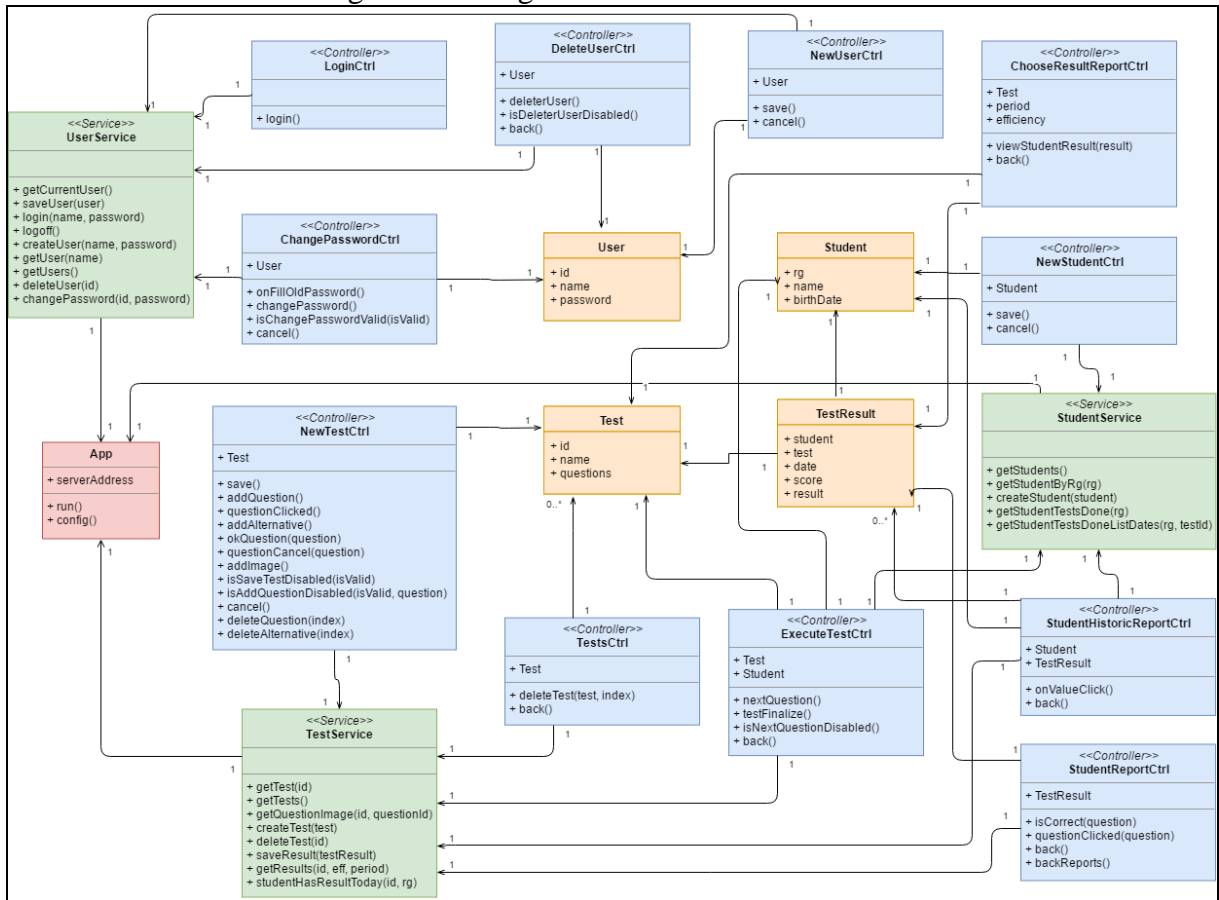
3.2.2 Diagrama de classes

O diagrama de classes apresenta uma visão de como as classes estão estruturadas e relacionadas. Nesta seção são descritas as classes necessários para o desenvolvimento desta plataforma, sendo que a aplicação está dividida em duas partes, o front-end e o back-end.

3.2.2.1 Diagrama de classes do front-end

A Figura 16 retrata a estrutura resumida de classes do front-end, desconsiderando menus que não lidam diretamente com dados. As classes estão divididas em 4 tipos diferentes: entidades (bege), controllers (azul), services (verde) e uma classe de inicialização (vermelho) e configuração de aplicação do Ionic.

Figura 16 - Diagrama de classes do front-end



Fonte: elaborado pelo autor.

Iniciando pelas entidades, que servem para representar um objeto oriundo de uma requisição ao back-end, transitam pelos controllers e representam os dados da manipulados no front-end da plataforma. A classe `User` contém as propriedades utilizadas pela plataforma para representar um usuário. A classe `Student` possui os atributos que modelam um aluno na plataforma. A classe `Test` retrata as características necessárias para representar na plataforma um teste. A classe `TestResult` contém os atributos que configuram um resultado de teste associado a um aluno e a um teste.

Os controllers permitem manipular os dados das entidades e controlar os comportamentos dos componentes que preenchem as páginas HTML da plataforma. A classe `LoginCtrl` representa o comportamento da tela inicial da plataforma e controla o login do usuário. A classe `NewUserCtrl` controla o cadastro de novos usuários e a remoção dos mesmo é controlado pela classe `DeleteUserCtrl`. A classe `ChangePasswordCtrl` disponibiliza a funcionalidade de alterar a senha do usuário logado na plataforma. A classe `NewStudentCtrl` permite ao usuário a inserção de novos alunos. A classe `NewTestCtrl` é responsável pelo cadastro de novos testes e também pela visualização de um teste já cadastrado. A classe `TestsCtrl` controla a listagem dos testes cadastrados na plataforma e

também disponibiliza a funcionalidade de remoção de algum dos testes cadastrados apenas para o administrador. A classe `ExecuteTestCtrl` controla a execução de um teste para um aluno e salva o resultado obtido ao final. A classe `StudentReportCtrl` disponibiliza um relatório com as informações da performance do aluno em um dado teste em certa data. A classe `StudentHistoricReportCtrl` permite acompanhar a evolução do aluno em um certo teste. Por fim, a classe `ChooseResultReportCtrl` é responsável por listar os resultados obtidos a partir de um teste escolhido pelo usuário para selecionar um dos resultados e assim mostrar o relatório.

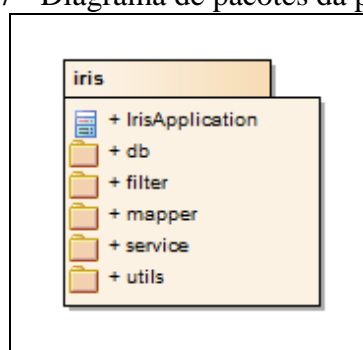
Em seguida, há os serviços que executam a função de comunicação com o back-end via requisições. A classe `UserService` contém as funções de comunicação com o servidor para acessar informações de usuários bem como o cadastro e remoção dos mesmos. A classe `TestService` fornece métodos, que através de requisições ao back-end, realiza manipulação dos testes e traz informações dos resultados dos alunos. A classe `StudentService` disponibiliza funções para criar alunos e retornar dados dos alunos cadastrados.

Por último, a classe `App` que é responsável por realizar configurações da aplicação na inicialização onde é realizado o mapeamento dos controllers de acordo com as páginas HTML e também configura o cache utilizado na plataforma.

3.2.2.2 Diagrama de pacotes do back-end

Para melhor organização do código, optou-se por separar as classes do aplicativo em diferentes pacotes, de acordo com sua especificidade. A Figura 17 exibe o diagrama de pacotes que compõem a plataforma Iris, são eles: `db`, `filter`, `mapper`, `service` e `utils`.

Figura 17 - Diagrama de pacotes da plataforma



Fonte: elaborado pelo autor.

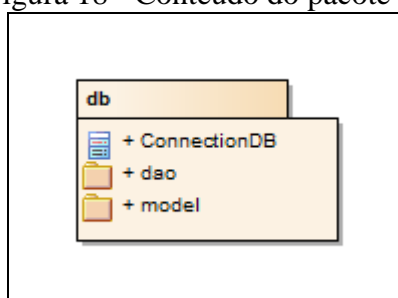
Como é possível ver no diagrama da Figura 17, os pacotes possuem dentro deles subpacotes que dividem as classes conforme sua especialidade. Há também a classe `IrisApplication` onde são feitas configurações de recursos para a aplicação e possui a anotação `@ApplicationPath("/")` que indica qual a URL base. As próximas seções

destinam-se ao detalhamento de cada pacote, destacando a responsabilidade de cada classe dentro da plataforma. A seção 3.2.2.2.1 descreve como o pacote `db` está estruturado e a função de cada classe dentro do mesmo. Na seção 3.2.2.2.2 é apresentado o pacote `filter` e a responsabilidade da classe nele contida. Na seção 3.2.2.2.3 é possível verificar o pacote `mapper` e seu conteúdo. A seção 3.2.2.2.4 explana como é o funcionamento das classes do pacote `service`. Ao final, a seção 3.2.2.2.5 mostra a finalidade do pacote `utils` e a classe contida no mesmo.

3.2.2.2.1 Pacote `db`

O pacote `db` é responsável pelas classes de acesso à base de dados e classes de modelos de dados. Conforme diagrama da Figura 18 é possível verificar que nesse pacote há a classe `ConnectionDB` que possui configurações de acesso à base de dados da plataforma e também o mapeamento de classes que realizam operações de manipulação de dados na base de dados. Este pacote ainda conta uma sub-divisão em dois pacotes: `dao` e `model`.

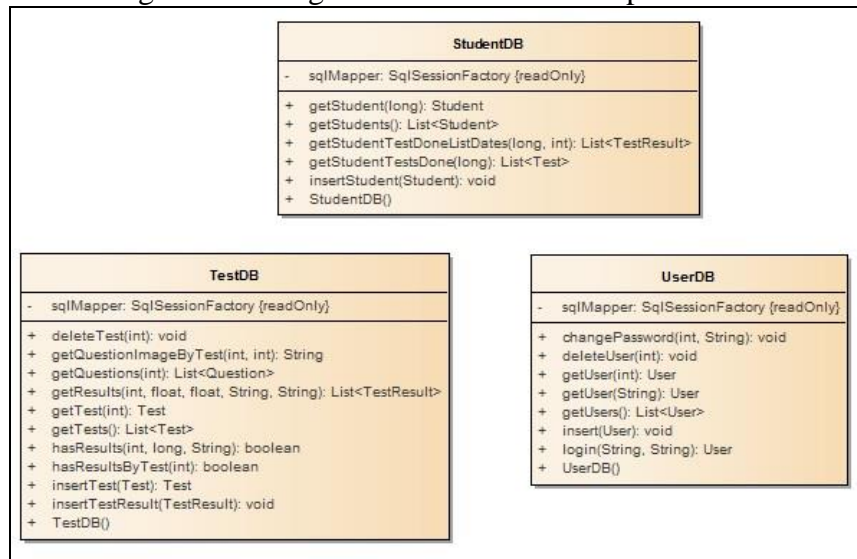
Figura 18 - Conteúdo do pacote `db`



Fonte: elaborado pelo autor.

O sub-pacote `dao`, como pode ser visto na Figura 19, contém lógica do tratamento dos dados obtidos da base de dados, fornecendo às classes do pacote `service` as informações a serem repassadas para o front-end.

Figura 19 - Diagrama de classes do sub-pacote dao

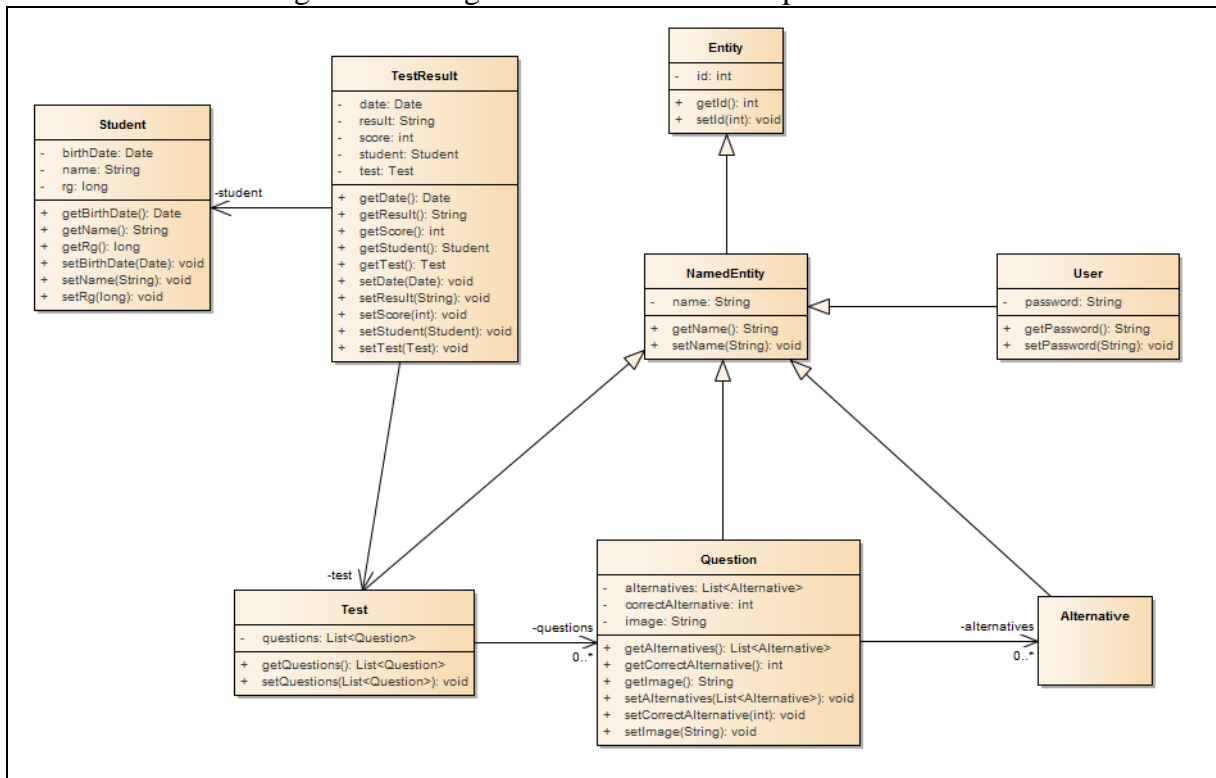


Fonte: elaborado pelo autor.

A classe `StudentDB` que é responsável por realizar operações que lidam com os dados obtidos da base de dados relevantes ao aluno. Esta classe possui métodos como `getStudent` que tem a função de recuperar dados de alunos cadastrados e também `getStudentTestsDone` que traz os testes que o aluno realizou, afim de filtrar e mostrar apenas opções válidas no front-end. A classe `TestDB` provém dados pertinentes aos testes e resultados dos mesmos para os serviços disponibilizados para o front-end. Ela contém métodos como `deleteTest` e `insertTest` que tratam da criação e remoção de testes e métodos que salvam e trazem informações dos resultados como `insertTestResult` e `getTestResult`. A classe `UserDB` engloba funções para manipular dados provenientes dos usuários, como `insert`, `deleteUser` e `changePassword` que tratam do cadastro, remoção e troca de senha respectivamente.

A Figura 20 mostra como as classes do sub-pacote `model` estão relacionadas as quais contém informações necessárias para a persistência dos dados no banco de dados. Todas elas, com exceção das classes `Student` e `TestResult`, herdam a classe `Entity` as quais possuem o campo `id` que é utilizado como chave primária. A classe `NamedEntity` é utilizada pelas classes que possuem algum tipo de descrição como campo, que seriam as classes `Alternative`, `Question` e `Test`. A classe `TestResult` representa o resultado de um teste, sendo assim, guarda uma referência para as classes `Test` e `Student`, e ainda possui o campo `data` de realização do teste como chave primária de acordo com a modelagem da base de dados.

Figura 20 - Diagrama de classes do sub-pacote model

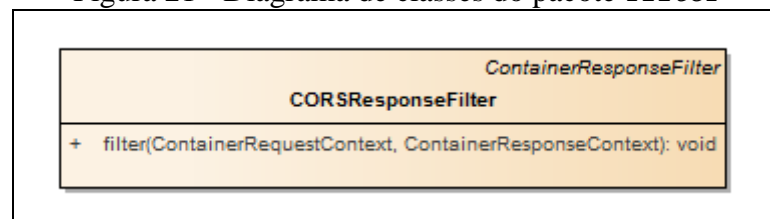


Fonte: elaborado pelo autor.

3.2.2.2.2 Pacote filter

O pacote `filter` é responsável por realizar configurações no header da resposta do servidor para o browser aceitar respostas de todos os tipos de requisições realizadas pelo front-end. JavaScript continua com uma implementação de segurança que já vem de muitos anos atrás que juntamente com os browsers só aceitam respostas do servidor se forem da mesma origem do protocolo + host + porta que o servidor, caso contrário pode rejeitar as respostas e assim a plataforma não consegue ter acesso aos dados que necessita. Dessa forma foi criada a classe `CORSResponseFilter`, de acordo com a Figura 21, para dar acesso às respostas e sua codificação será descrita na seção de implementação.

Figura 21 - Diagrama de classes do pacote filter

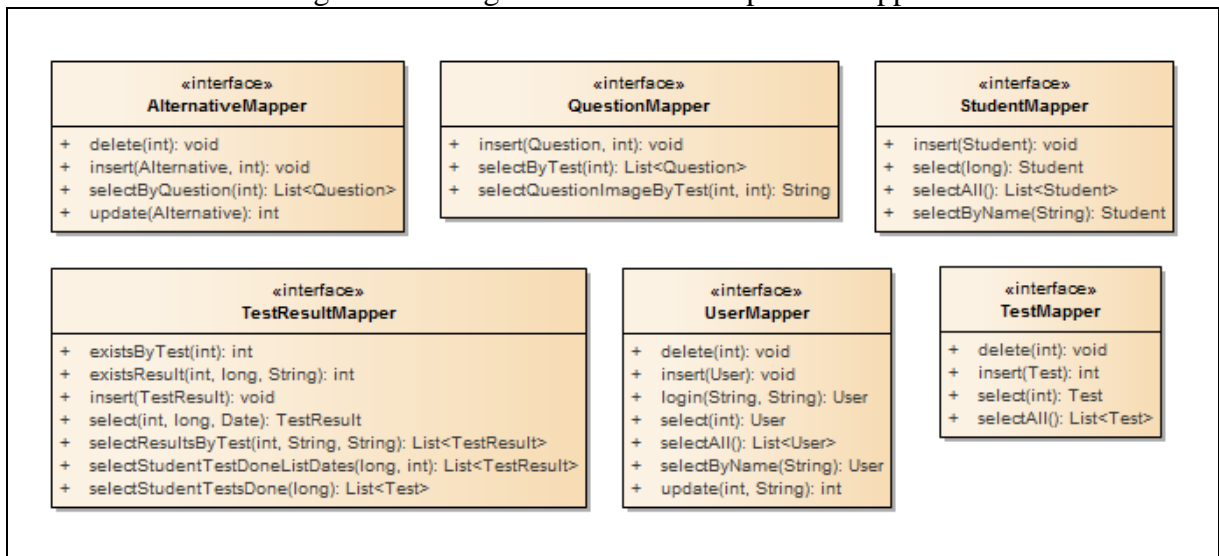


Fonte: elaborado pelo autor.

3.2.2.2.3 Pacote mapper

O conteúdo do pacote `mapper` é ilustrado de acordo com a Figura 22. Ele conta com classes mapeadoras de comandos SQL para métodos Java em interfaces com acesso simplificado no código e mantendo todo acesso ao banco isolado. Este pacote contém as interfaces `AlternativeMapper`, `QuestionMapper`, `StudentMapper`, `TestResultMapper`, `UserMapper` e `TestMapper` que realizam a função citada acima e são todas chamadas pelas classes do pacote `db.dao` para retornar os dados e manipulá-los para retornar ao front-end.

Figura 22 - Diagrama de classes do pacote mapper

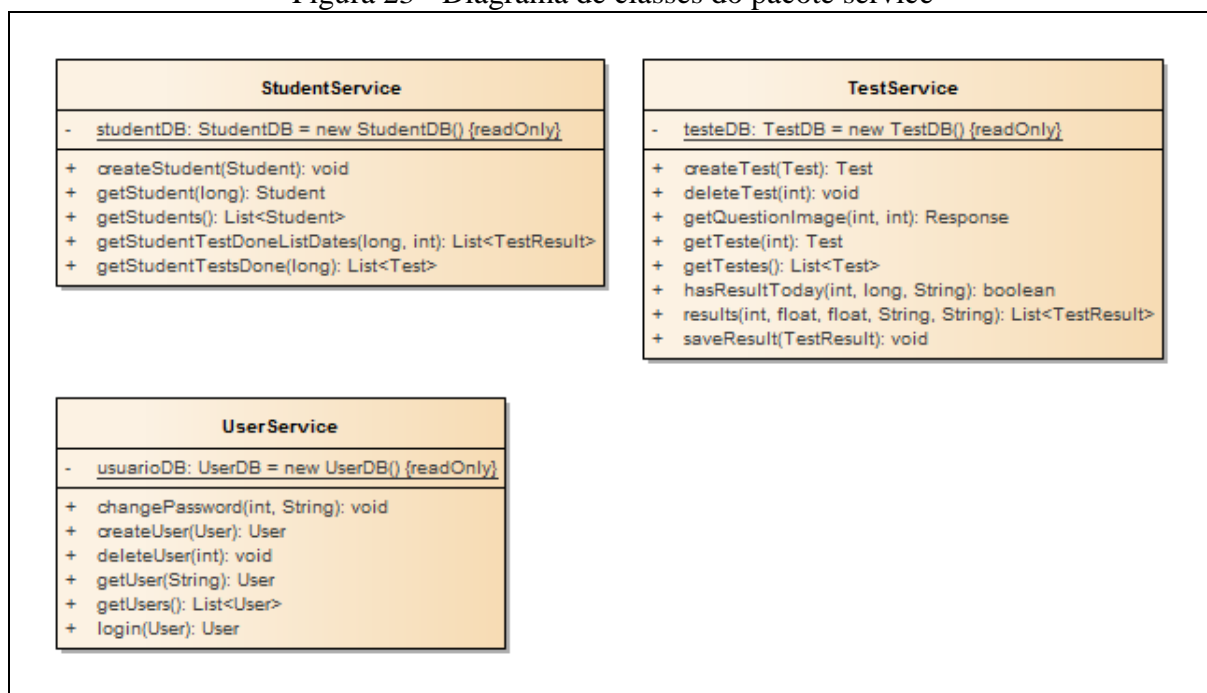


Fonte: elaborado pelo autor.

3.2.2.2.4 Pacote service

A Figura 23 apresenta o diagrama de classes do pacote `service`. Os web services foram construídos no estilo arquitetural REST utilizando o *framework* Jersey Restful. A URL base do sistema já foi configurada na classe `IrisApplication` no pacote principal.

Figura 23 - Diagrama de classes do pacote service



Fonte: elaborado pelo autor.

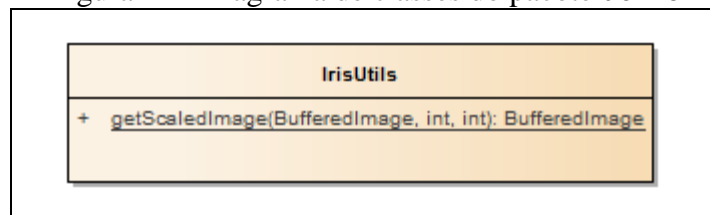
A classe `StudentService` possui a anotação `@Path("/student")` que determina o contexto deste web service e os métodos contidos nessa classe tem uma anotação `@Path("...")` que determina a URL pela qual o front-end deve mandar uma requisição para conseguir o retorno dos valores desejados, por exemplo o método `createStudent` que tem a anotação `@Path("/create")` e é responsável pelo cadastro de um novo aluno na plataforma. Assim, se forma a URL utilizando a URL base da aplicação que é configurada na classe `IrisApplication` e se concatena com a URL do contexto do aluno e mais a URL de método, constituindo uma URL como `.../student/create`. Na classe `TestService` há a anotação `@Path("/test")` que define o contexto deste web service e pode-se encontrar os métodos de comunicação com o front-end referentes a cadastro e remoção de testes e informações referentes aos resultados dos testes aplicados. A classe `UserService` contém a anotação `@Path("/user")` e dispõe de métodos que fornecem ao front-end condições de manipular dados relevantes aos usuários da plataforma e controla o login dos mesmos.

Esses *web services* não possuem lógica de programação, eles apenas repassam informações para o front-end ou salvam dados na base de dados através de referências às classes `TestDB`, `StudentDB` e `UserDB` que eles contém.

3.2.2.2.5 Pacote `utils`

A Figura 24 ilustra o conteúdo do pacote `utils`. Ele conta apenas com a classe `IrisUtils` que contém o método `getScaledImage` que tem função utilitária de redimensionar imagens maiores que 400 *pixels* de altura por 400 *pixels* de largura para obedecer este tamanho máximo sem haver distorções, com objetivo de não haver muita diferença no tamanho das imagens entre a execução de um teste em um *smartphone* e um *tablet* por exemplo. Diminuindo a imagem também diminui o tráfego de dados pela internet e deixa a plataforma mais rápida e responsiva.

Figura 24 - Diagrama de classes do pacote `utils`

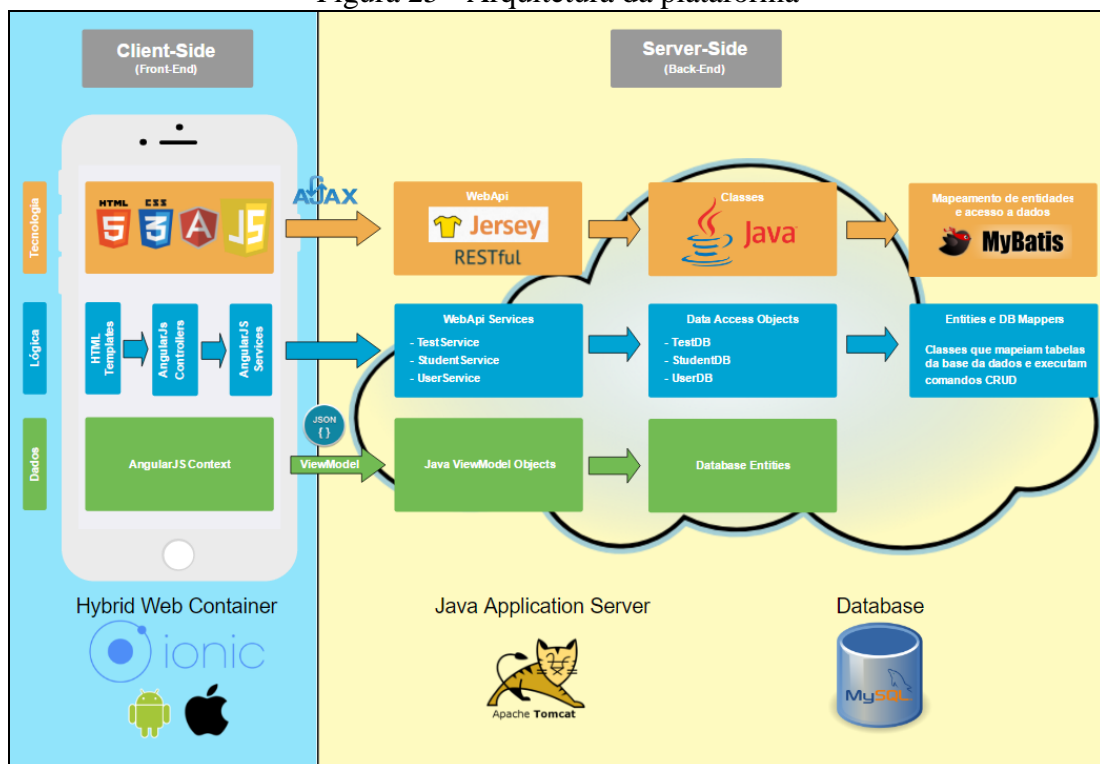


Fonte: elaborado pelo autor.

3.2.3 Arquitetura e integrações do aplicativo

Para o funcionamento deste trabalho, vários serviços e ferramentas de terceiros foram utilizados. A Figura 25 mostra a arquitetura macro criada com a integração de todas estas ferramentas.

Figura 25 - Arquitetura da plataforma



Fonte: elaborado pelo autor.

A camada `client-side` representa o dispositivo móvel do usuário com a plataforma Iris instalada. Este por sua vez, é um `Hybrid Web Container` desenvolvido com o *framework* Ionic e que pode ser construído para dispositivos com sistemas operacionais Android e iOS. A camada do Ionic encapsula o que seria uma implementação de um cliente de um web site, que envolve o uso de diversas tecnologias retratadas na figura mencionada que são utilizadas amplamente para o desenvolvimento de páginas de internet. Essas tecnologias orquestram a visualização das páginas, que seriam HTML e CSS, e a manipulação dos dados pelos `controllers` e `services` desenvolvidos em AngularJS e a lógica que trabalha com esses dados utilizando JavaScript. Os dados no front-end são mantidos pelo contexto do AngularJS. Por fim, o front-end se comunica com o servidor via requisições Ajax utilizadas internamente pelo AngularJS e enviam os dados em formato JSON.

Na camada `server-side` encontra-se a implementação do back-end desenvolvido utilizando a linguagem de programação Java que trabalha com requisições REST utilizando a Web Api Jersey e mapeamento de entidades e acesso a dados é realizado pelo *framework* MyBatis. Nesta camada é realizada a persistência dos dados e busca de informações da base de dados Oracle MySQL. Esses dados são manipulados no back-end tanto para salvar informações no SGBD como também retornar informações para o front-end disponibilizar para o usuário. Para comportar o servidor, é utilizado o servidor de aplicações Java, Apache Tomcat, que é responsável pelo gerenciamento do acesso à aplicação do `server-side`.

3.3 IMPLEMENTAÇÃO

Nesta seção são apresentadas as técnicas e ferramentas utilizadas no trabalho e as etapas seguintes para o desenvolvimento da plataforma Iris, dividido em front-end e back-end para facilitar o entendimento. Para finalizar esta seção são apresentadas as telas e funcionalidades do protótipo desenvolvido.

3.3.1 Técnicas e ferramentas utilizadas

O desenvolvimento deste trabalho está dividido em quatro estágios. O primeiro deles é a criação do servidor back-end utilizando a linguagem de programação Java, no ambiente de desenvolvimento Eclipse e em conjunto com a ferramenta para automação de compilação e gerenciamento de dependências Maven. Para as operações com banco de dados foi utilizado o *framework* MyBatis que permitiu utilizar um mapeament objeto-relacional para aumentar a abstração na manipulação de tabelas e atributos. Para a camada de serviço foi utilizada a Web

API Jersey, permitindo através de anotações no código-fonte, mapear as chamadas web services para os métodos correspondentes.

O segundo estágio é o desenvolvimento do front-end, onde o usuário tem acesso direto à plataforma, podendo assim cadastrar e aplicar testes aos alunos e conferir os seus resultados. Para a construção do código-fonte, utilizou-se as ferramentas padrões de desenvolvimento web HTML, CSS, Javascript, com auxílio do editor de textos Sublime Text. Foram utilizados os *frameworks* Bower e NPM para gerenciamento de dependências. O NPM foi utilizado para o gerenciamento de módulos como o Ionic e Cordova, já o Bower foi utilizado para gerenciar componentes que são utilizados no código do front-end, como o AngularJS. O Ionic foi utilizado para construir a aplicação para dispositivos com sistema operacional Android, empacotando os elementos do front-end e formando um arquivo instalável e executável em celulares, por exemplo. Cordova foi necessário para ter acesso ao diretório da biblioteca de imagens do Android na plataforma, possibilitando ao usuário inserir imagens nos testes que estão nos diretórios de imagens do seu dispositivo Android. O *framework* AngularJS foi empregado para aumentar a abstração do código Javascript, utilizando diretivas que facilitaram a integração entre os dados de domínio e as páginas HTML e organizando o código de forma simples e legível. O AngularJS também foi utilizado para realizar as chamadas web service do tipo RESTful, utilizando o padrão de anotação de objetos JSON para enviar e receber dados do servidor, permitindo manipular de forma acessível os objetos.

O terceiro estágio é a adição de alguns plugins no front-end: um para trabalhar com listas drop-down e pesquisa de registros (*ui-select*); um para trabalhar com sliders para definir períodos de datas (*rzModule*); um para trabalhar com a plotagem de gráficos (*chart*); um para trabalhar com mensagens de erro disponibilizadas ao usuário no preenchimento de formulários (*ng-messages*); um para trabalhar com textareas para torná-los com tamanhos dinâmicos de acordo com o preenchimento do seu conteúdo (*ng-elastic*); um para acessar a biblioteca de imagens do Android (*cordova-plugin-camera*); e o último para permitir a manutenção de caches de testes, alunos e imagens na plataforma (*angular-cache*). Foi utilizado o módulo *ngCordova*, conforme citado anteriormente, também para acessar os plugins no formato de injeção de dependências utilizando módulos do AngularJS.

O último estágio para o desenvolvimento da plataforma é a criação do aplicativo para dispositivos móveis Iris, construído utilizando o *framework* Ionic na plataforma Phonegap. Para construir o aplicativo para o sistema operacional android para testes e sem publicar no Google Play Store, basta via linha de comando acessar o diretório da aplicação Ionic e executar o comando `ionic build android`, empacotando a aplicação em um arquivo `.apk`

que é possível de instalar no dispositivo Android. Para depuração local tanto em emuladores Android como dispositivos reais conectados via USB no computador basta executar o comando `ionic run android`.

3.3.2 Principais implementações da plataforma

Nesta seção encontram-se os trechos de códigos mais relevantes da plataforma, com seus devidos detalhamentos. A seção 3.3.2.1 apresenta o cache de dados da plataforma, utilizado para diminuir o consumo de internet e aumentar o desempenho. A seção 3.3.2.2 mostra como foi desenvolvido o acesso à biblioteca de imagens do dispositivo móvel. Por fim, a seção 3.3.2.3 detalha a implementação dos mapeadores SQL, abstraindo e facilitando o trabalho com o banco de dados em servidores Java.

3.3.2.1 Cache de dados

Para trabalhar com o cache de dados no front-end, primeiramente é necessário importar o plugin `angular-cache`. Feito este procedimento é necessário declará-lo nas configurações do AngularJS como apresentado na Quadro 8, onde na linha 2 é realizada a extensão do cache do cache padrão do angular e como configuração inicial é determinado que o tempo máximo de vida de cada item adicionado no cache é de 15 minutos, passado esse período é descartado e será preenchido novamente com informações do back-end caso haja uma nova requisição para tal.

Quadro 8 - Configuração inicial do cache

1	<code>.config(function (CacheFactoryProvider) {</code>
2	<code>angular.extend(CacheFactoryProvider.defaults, { maxAge: 15 * 60 *</code>
3	<code>1000 });</code>
4	<code>});</code>

Fonte: elaborado pelo autor.

Para utilização desse cache, foram adicionadas instâncias nas classes `StudentService` e `TestService`, onde são armazenadas informações dos alunos, testes e imagens de forma separada. O Quadro 9 ilustra as configurações dos caches `testCache` e `imageCache`, com as propriedades `maxAge` definida para valores entre quinze e vinte minutos no descarte de cada item adicionado, sobrescrevendo a configuração inicial, conforme mostra o Quadro 8. Há também as propriedades `cacheFlushInterval` definido para uma hora, que é o descarte do cache inteiro independente do tempo restante que o item ainda tem vivo no cache e a propriedade `deleteOnExpire` define que os objetos no cache serão removidos na expiração do cache.

Quadro 9 - Configuração das instâncias dos caches

```

1 Iris.service('TestService', ['$http', '$q', '$cordovaFile',
2 'CacheFactory', function($http, $q, $cordovaFile, CacheFactory) {
3
4     var urlBase = serverAddress + '/test/';
5
6     CacheFactory('testCache', {
7         maxAge: 15 * 60 * 1000,
8         cacheFlushInterval: 60 * 60 * 1000,
9         deleteOnExpire: 'aggressive'
10    });
11
12    CacheFactory('imageCache', {
13        maxAge: 20 * 60 * 1000,
14        cacheFlushInterval: 60 * 60 * 1000,
15        deleteOnExpire: 'aggressive'
16    });

```

Fonte: elaborado pelo autor.

A utilização do cache foi restrita apenas aos serviços do front-end para aumentar abstração do código e evitar que fique espalhado por todo o front-end possibilitando a perda de controle e um código de difícil leitura, assim o cache é apenas acessado quando é chamado um serviço. O Quadro 10 apresenta usos do cache na classe `TestService` e percebe-se que se assemelha muito com implementações de mapas que estão disponíveis em várias plataformas e linguagens, trabalhando com cache e valor e com métodos `put`, `get` e `remove`.

Quadro 10 - Utilização dos caches

```

1     this.createTest = function(test) {
2         var deferred = $q.defer();
3         var testCache = CacheFactory.get('testCache');
4
5         $http.post(urlBase + "create", test).success(function(test) {
6             for(i = 0; i < test.questions.length; i++) {
7                 test.questions[i].image = null;
8             }
9             testCache.put(test.id.toString(), test);
10            deferred.resolve(test);
11        });
12
13        return deferred.promise;
14    };
15
16    this.deleteTest = function(testId) {
17        var deferred = $q.defer();
18        var testCache = CacheFactory.get('testCache');
19
20        $http.delete(urlBase + testId).success(function() {
21            testCache.remove(testId.toString());
22            deferred.resolve(true);
23        }).error(function(data) {
24            deferred.resolve(false);
25        });
26
27        return deferred.promise;

```

Fonte: elaborado pelo autor.

3.3.2.2 Acessando o diretório da biblioteca de imagens do dispositivo móvel

Na criação de testes, o administrador tem a opção de adicionar imagens às questões dos testes para ilustrar. Foi utilizado o plugin `cordova-plugin-camera` com a extensão `ngCordova` para acessar o diretório da biblioteca de imagens do dispositivo. O Quadro 11 apresenta o código-fonte utilizado, onde na linha 3 é definido o objeto `options` com as opções disponíveis no plugin. Neste objeto estão configuradas algumas propriedades como na linha 4 `destinationType` que define como a image será enviada ao destino, no caso foi escolhido que serão mandados os dados dela em base64. A origem da imagem é proveniente do atributo `sourceType` que é especificado para utilizar a biblioteca de imagens do dispositivo. O `allowEdit` está configurado para não permitir a edição das imagens logo antes de escolhê-las e o `encodingType` foi determinado que gerasse imagens do tipo JPEG. Ao selecionar a imagem com sucesso, ela é passada como parâmetro para o método `onImageSuccess` na linha 12 que posteriormente é enviada ao servidor back-end.

Quadro 11 - Utilizando a biblioteca de imagens do dispositivo móvel

```

1  $scope.addImage = function() {
2
3      var options = {
4          destinationType: Camera.DestinationType.DATA_URL,
5          sourceType: Camera.PictureSourceType.PHOTOLIBRARY,
6          allowEdit: false,
7          encodingType: Camera.EncodingType.JPEG
8      };
9
10     $cordovaCamera.getPicture(options).then(function(imageData) {
11
12         onImageSuccess(imageData);
13
14         function onImageSuccess(dataURL) {
15             $scope.selectedQuestion.image = dataURL;
16             $scope.currentImage = "data:image/jpeg;base64," +
dataURL;
17         }
18     }, function(err) {
19         console.log(err);
20     });
21 }

```

Fonte: elaborado pelo autor.

3.3.2.3 Mapeadores de comandos SQL

Primeiramente é necessário configurar a conexão com o banco de dados, onde as credenciais de acesso estão armazenadas na linha 5 do arquivo `iris/db/configuration.xml`, conforme mostra o Quadro 12. Entre as linhas 15 e 20 é feito o mapeamento das interfaces que contém métodos que realizam as operações na base de dados.

Quadro 12 - Configurando os mapeadores

```

1  public class ConnectionDB {
2
3      private static SqlSessionFactory sqlMapper = null;
4      private static final String configurationsFile =
5  "iris/db/configuration.xml";
6
7      public static SqlSessionFactory getSqlMapper() {
8          if (sqlMapper == null) {
9              Reader reader = null;
10             try {
11                 reader =
12 Resources.getResourceAsReader(configurationsFile);
13                 sqlMapper = new
14 SqlSessionFactoryBuilder().build(reader, "desenvolvimento");
15                 sqlMapper.getConfiguration().addMapper(UserMapper.class);
16                 sqlMapper.getConfiguration().addMapper(TestMapper.class);
17                 sqlMapper.getConfiguration().addMapper(QuestionMapper.class);
18                 sqlMapper.getConfiguration().addMapper(AlternativeMapper.class);
19                 sqlMapper.getConfiguration().addMapper(StudentMapper.class);
20                 sqlMapper.getConfiguration().addMapper(TestResultMapper.class);
21             } catch (final Throwable t) {
22                 t.printStackTrace();
23             }
24             return sqlMapper;
25         }
26     }
27 }

```

Fonte: elaborado: pelo autor.

O Quadro 13 expõe a implementação de um das interfaces mapeadores do back-end para exemplificar o processo de desenvolvimento utilizando a biblioteca de persistência de dados MyBatis. É necessário criar os métodos Java, passando parâmetros caso necessário, e adicionar anotações como `@Select`, `@Insert`, `@Update`, `@Delete` para os respectivos comandos SQL. Caso o comando seja um `select`, por vezes o objeto retornado pode trazer referências para outros objetos, como na linha 5, onde um teste pode ter uma lista de questões, por isso é necessário mapear as questões conforme foram mapeadas no objeto modelo da base de dado `Test`. Isso agiliza o desenvolvimento do back-end e isolando a camada de negócio, com todas essas regras concentradas em um só ponto para fácil manutenção.

Quadro 13 - Mapeamento de comandos SQL para métodos Java

```

1  public interface TestMapper {
2
3      @Select("SELECT id, name FROM test WHERE id = #{id}")
4      @Results({ @Result(property = "id", column = "id"),
5                  @Result(property = "questions", column = "id",
6 javaType = List.class, many = @Many(select =
7 "iris.mapper.QuestionMapper.selectByTest")) })
8      Test select(@Param("id") int id);
9
10     @Select("SELECT * FROM test")
11     @Results({ @Result(property = "id", column = "id"),
12                 @Result(property = "questions", column = "id",
13 javaType = List.class, many = @Many(select =
14 "iris.mapper.QuestionMapper.selectByTest")) })
15     List<Test> selectAll();
16
17     @Insert("INSERT INTO test (name) VALUES (#{name})")
18     @Options(useGeneratedKeys = true, keyProperty = "id")
19     int insert(Test test);
20
21     @Delete("DELETE FROM test WHERE id = #{id}")
22     void delete(int id);
23 }

```

Fonte: elaborado pelo autor.

3.3.3 Diagrama de atividades

A

Figura 26 apresenta um diagrama de atividades onde são representadas as etapas e ações do usuário na utilização da plataforma. Quando a plataforma é aberta, o usuário se depara com a tela de login onde precisa preencher suas credenciais de forma correta para poder ter acesso à plataforma. Por padrão, a plataforma disponibiliza apenas um usuário cadastrado que é o do administrador, com usuário chamado “admin” e senha com valor de “admin” e a partir deste usuário pode-se inserir novos, porém sem acesso total ao sistema. Cada usuário pode alterar a senha apenas sua própria senha. Realizada a autenticação com sucesso, a plataforma apresenta uma série de opções no menu principal, mas neste momento preferiu-se abordar o fluxo mais utilizado na plataforma. O usuário seleciona a opção de aplicar o teste e necessita selecionar o aluno e o teste dentre os cadastrados. Caso o aluno não esteja cadastrado ou o cache de alunos esteja vazio, o usuário cadastra este aluno na plataforma. Para os testes, apenas o administrador tem acesso de cadastro, então cabe os outros usuários apenas escolher um dos testes disponibilizados. Realizado o cadastro do novo aluno, a plataforma insere o aluno no cache do front-end e carrega novamente os alunos e testes do cache e o usuário pode executar o teste. Ao final do teste o resultado é salvo e o usuário pode acessar o relatório do teste recém aplicado com informações da porcentagem de

acertos e cada resposta de cada questão. Os resultados dos testes não possuem cache, pois não é mandatório que o usuário sempre acessará os resultados, porém é mantido um cache de imagens para cada teste, caso tente-se acessar as respostas dos alunos, onde é possível visualizar a questão e todas as informações bem como a alternativa selecionada.

Figura 26 - Diagrama de atividades



Fonte: elaborado pelo autor.

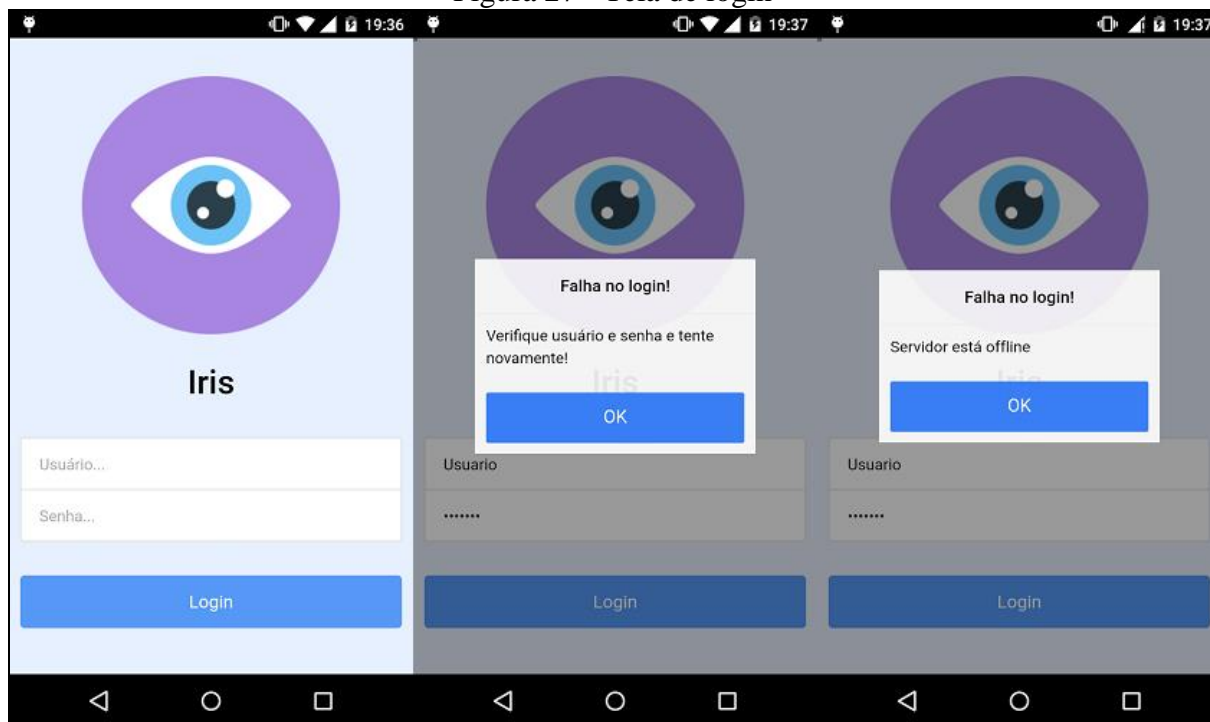
3.3.4 Operacionalidade da implementação

Nesta seção é apresentada a operacionalidade da plataforma desenvolvida, através da interface gráfica disponibilizada ao usuário. As imagens abaixo foram obtidas utilizando um dispositivo Motorola Moto X modelo XT1058 com Android 5.1.

3.3.4.1 Login e menu principal da aplicação

Ao inicializar a plataforma no dispositivo móvel, o usuário tem acesso a tela de login, onde ele pode realizar a autenticação. Foram adicionadas mensagens de notificação ao usuário quando há erro na autenticação do usuário e quando o servidor está offline conforme apresentado na Figura 27.

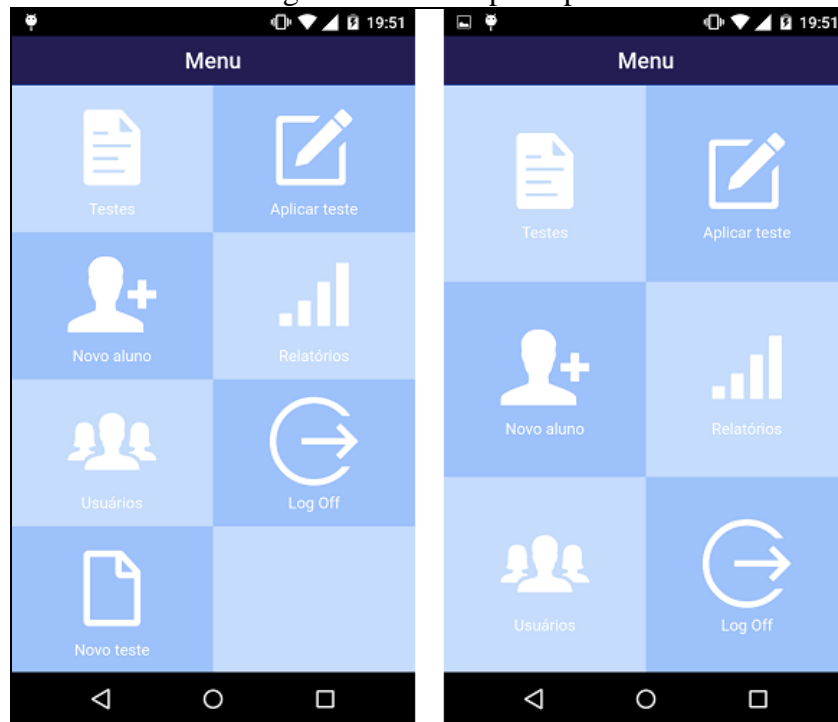
Figura 27 - Tela de login



Fonte: elaborado pelo autor.

Realizada a autenticação, o usuário se depara com o menu principal que conta com funções de cadastros de alunos, testes e usuários e visualização de relatórios. De acordo com a Figura 28, a visão do administrador é a tela que se encontra na esquerda e conta com uma função adicional de cadastro de novos testes que não está presente para o restante dos usuários conforme tela da direita.

Figura 28 - Menu principal

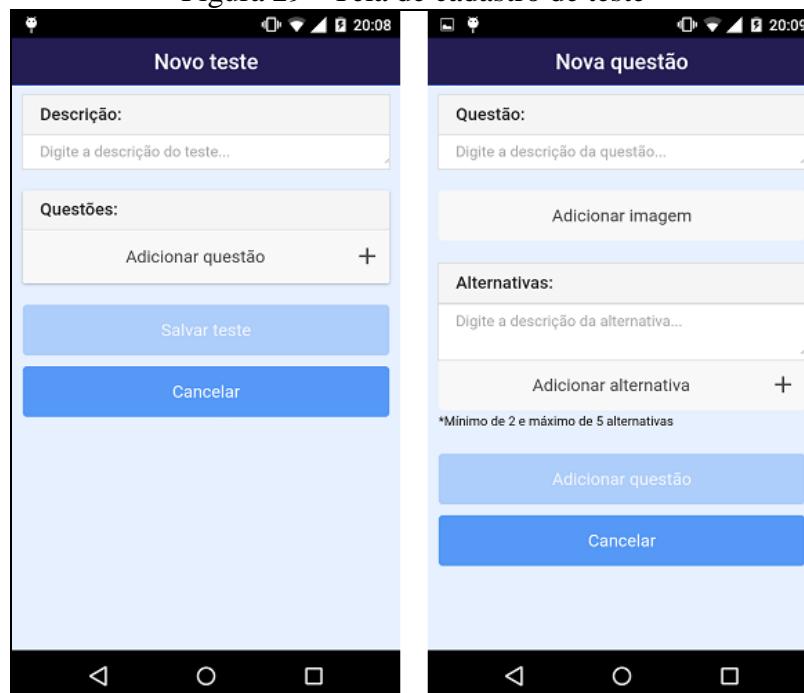


Fonte: elaborado pelo autor.

3.3.4.2 Testes

Para cadastrar novos testes, o administrador conta com a interface apresentada na Figura 29. À esquerda é a tela inicial e visão geral do teste e ao clicar para adicionar uma questão é mostrada a tela da direita.

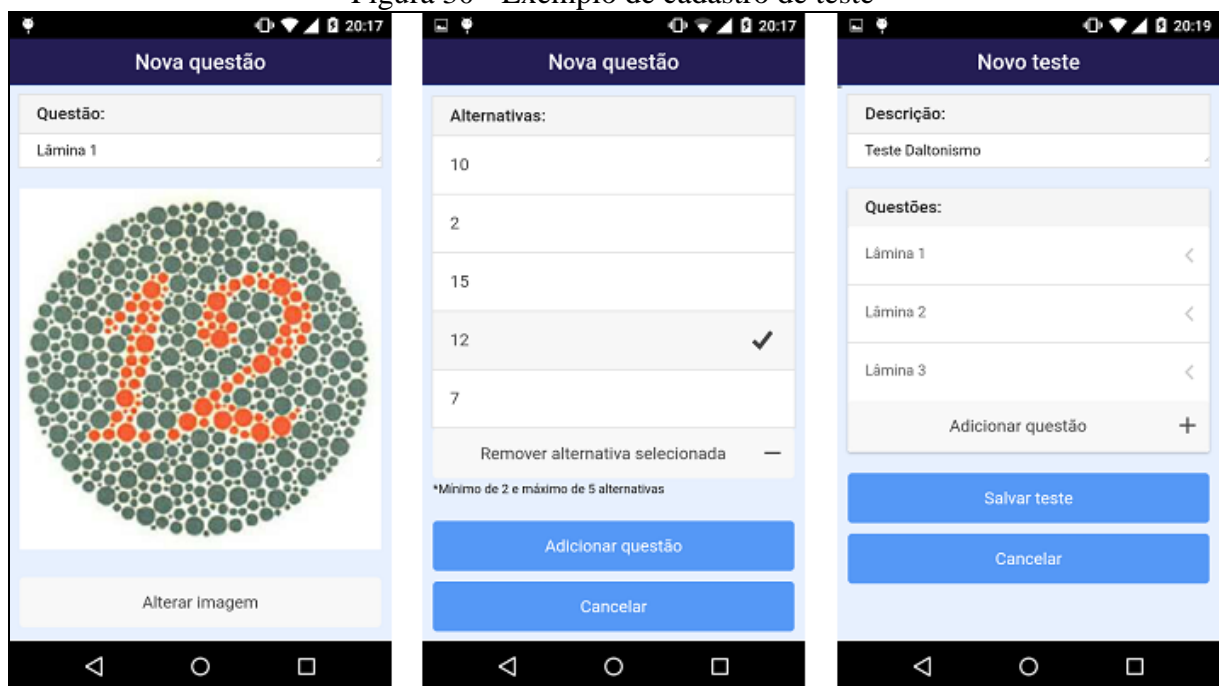
Figura 29 - Tela de cadastro de teste



Fonte: elaborado pelo autor.

Para um teste ser cadastrado é obrigatório que ele tenha pelo menos uma questão e uma descrição com no mínimo cinco caracteres e para adicionar uma questão é necessário que possua descrição com no mínimo cinco caracteres e que ela possua no mínimo duas e no máximo cinco alternativas. Após adicionar as alternativas o usuário tem que selecionar a correta e assim completa-se o cadastro daquela questão. A Figura 30 ilustra esse processo, onde o usuário insere uma questão no Teste de Daltonismo com uma figura e ao final atualiza a lista de questões, porém até este momento todos os dados estão em memória que são apenas salvos ao clicar o botão de `Salvar Teste`.

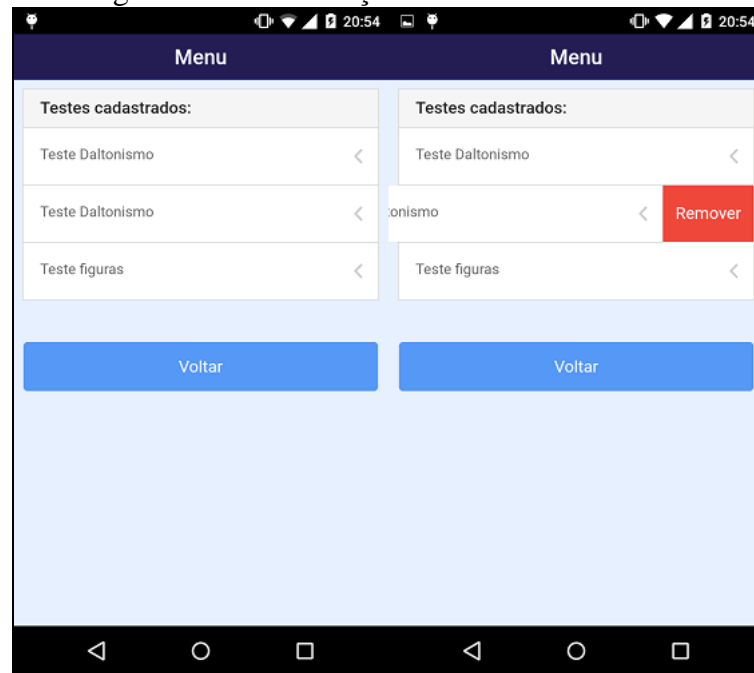
Figura 30 - Exemplo de cadastro de teste



Fonte: elaborado pelo autor.

Para visualizar os testes cadastrados, foi disponibilizada a tela ilustrada pela Figura 31. Todos os usuários podem acessar a tela de visualização de testes cadastrados, porém apenas administradores tem a permissão de remover testes, mas para realizar a remoção de um teste é necessário que ele não tenha sido aplicado a nenhum aluno, caso contrário o usuário será informado e a operação cancelada. A Figura 31 exibe a tela em questão, que lista os testes cadastrados e possibilita que o usuário selecione um deles e visualize seu conteúdo nos mesmos modos que a seção interior do cadastro do teste, embora todas as informações estejam em formato read-only.

Figura 31 - Visualização dos testes cadastrados

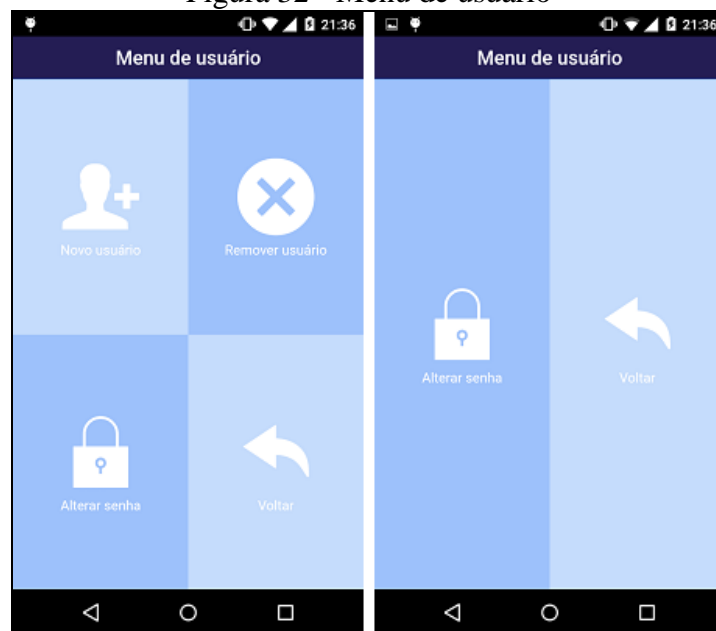


Fonte: elaborado pelo autor.

3.3.4.3 Opções de usuário

A Figura 32 apresenta as opções disponibilizadas ao usuário logado na plataforma, no lado esquerdo é a visão do administrador, que pode adicionar e remover usuários, e à direita é a visão dos outros usuários que contam apenas com a função de alteração de senha.

Figura 32 - Menu de usuário

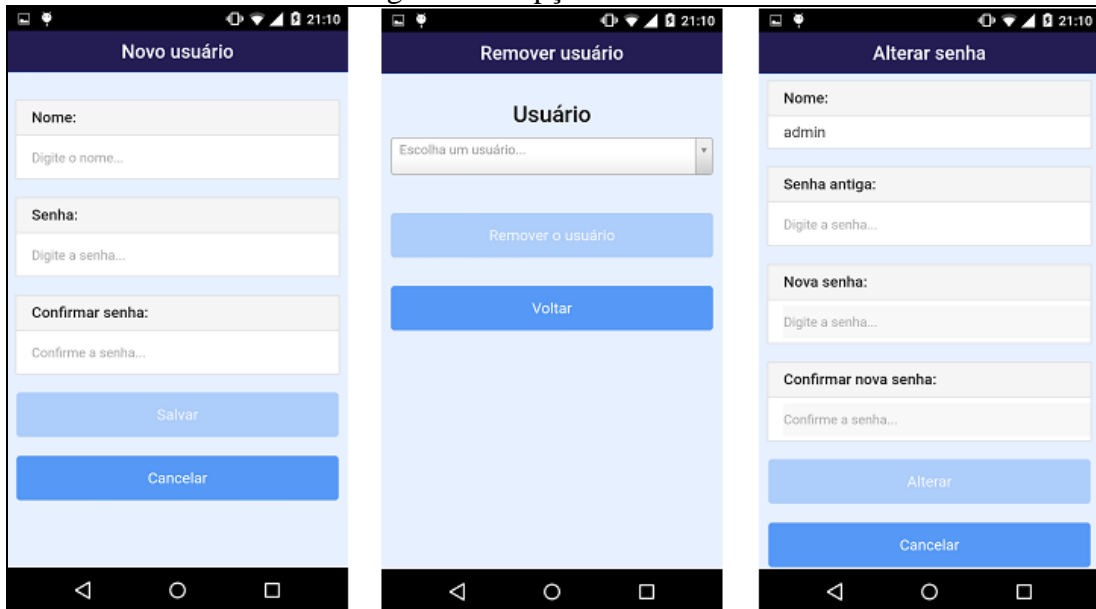


Fonte: elaborado pelo autor.

Na Figura 33 é possível ver com detalhes as telas de cada opção de usuário, à esquerda é o cadastro de novos usuários, ao centro é a remoção de usuários existentes na plataforma e à

direita é a alteração da senha, onde deve-se preencher a senha antiga e cadastrar a nova para que seja salvo com sucesso.

Figura 33 - Opções de usuário

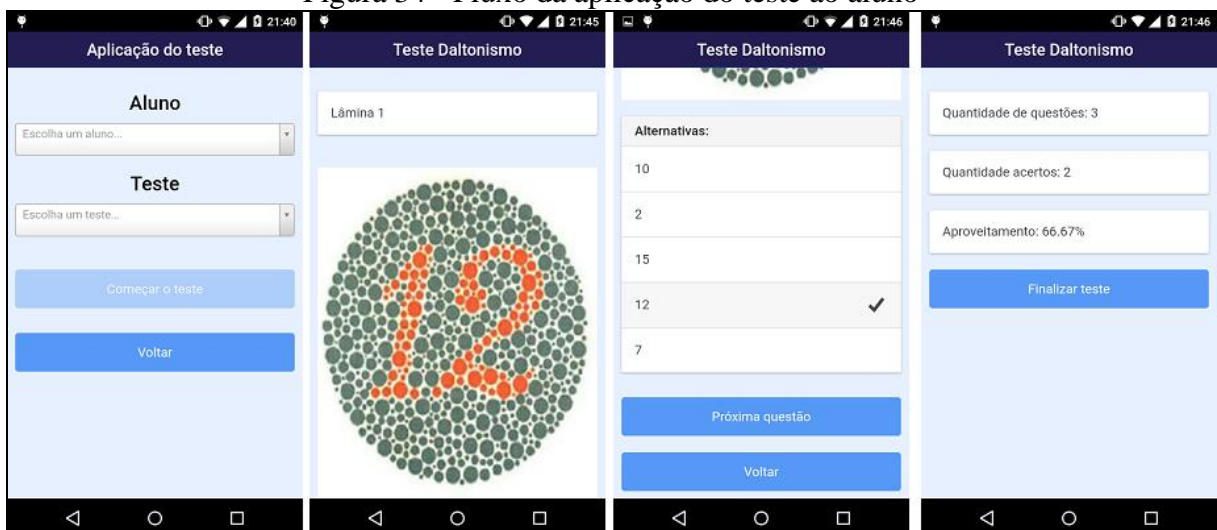


Fonte: elaborado pelo autor.

3.3.4.4 Aplicação de teste a um aluno

A plataforma permite que todos os usuários possam aplicar testes aos alunos cadastrados. A Figura 34 mostra as telas envolvidas no processo, começando com a seleção do aluno e do teste a ser executado e depois iterando por todas as questões onde o aluno responde cada questão com a alternativa que considerar mais adequada. Ao final são apresentados dados como quantidade de questões do teste, quantidade de acertos do aluno e o percentagem de aproveitamento do aluno no teste, gravando as informações do resultado.


Figura 34 - Fluxo da aplicação do teste ao aluno



Fonte: elaborado pelo autor.

De acordo com a primeira tela da Figura 34, caso o usuário perceba que o aluno não esteja cadastrado, foi disponibilizada a tela de cadastro do estudante. O cadastro do aluno pode ser realizado por todos os usuários. Todos os campos são obrigatórios e podem ser vistos na Figura 35. Não é permitido o cadastro de RGs duplicados.

Figura 35 - Cadastro do aluno



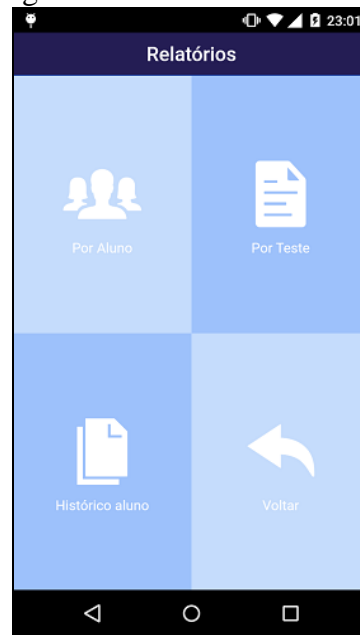
A imagem mostra uma tela de um aplicativo móvel intitulada "Novo aluno". No topo, há uma barra de status com ícones de bateria, Wi-Fi e sinal de celular, e o horário 21:08. Abaixo, o título "Novo aluno" está em uma barra azul escura. O formulário principal tem um fundo branco e contém três campos de entrada: "Nome:" com o placeholder "Digite o nome...", "RG:" com o placeholder "Digite o rg...", e "Data de nascimento:". Abaixo dos campos, há dois botões: "Salvar" (em azul claro) e "Cancelar" (em azul escuro). Na base da tela, há uma barra de navegação com ícones de voltar, home e recentes.

Fonte: elaborado pelo autor.

3.3.4.5 Relatórios

Por fim, há a funcionalidade de listar relatório conforme o menu ilustrado pela Figura 36. Todos os usuários possuem a permissão para visualizar todos esses relatórios.

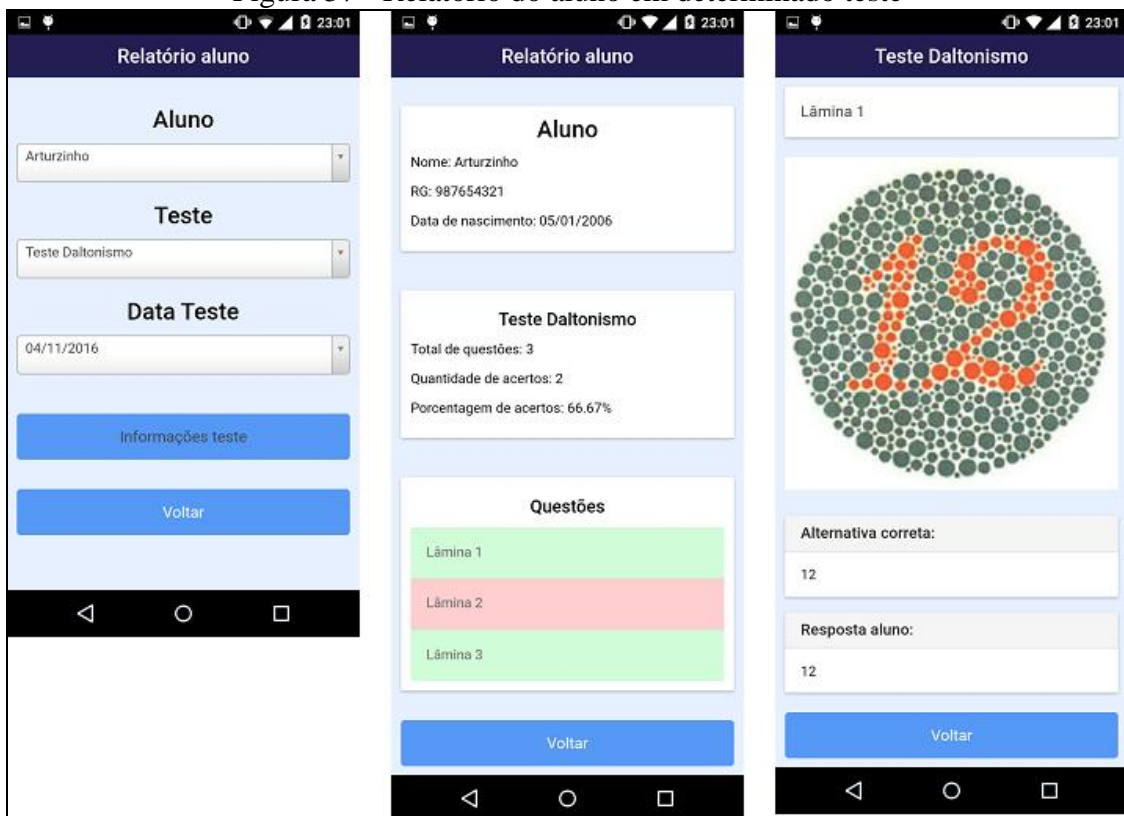
Figura 36 - Menu de relatórios



Fonte: elaborado pelo autor.

A opção do relatório por aluno mostra as informações de um aluno, em certo teste e determinada data, conforme exposto pela Figura 37. Assim é possível visualizar todas as informações do aluno, desempenho no teste e respostas das questões.

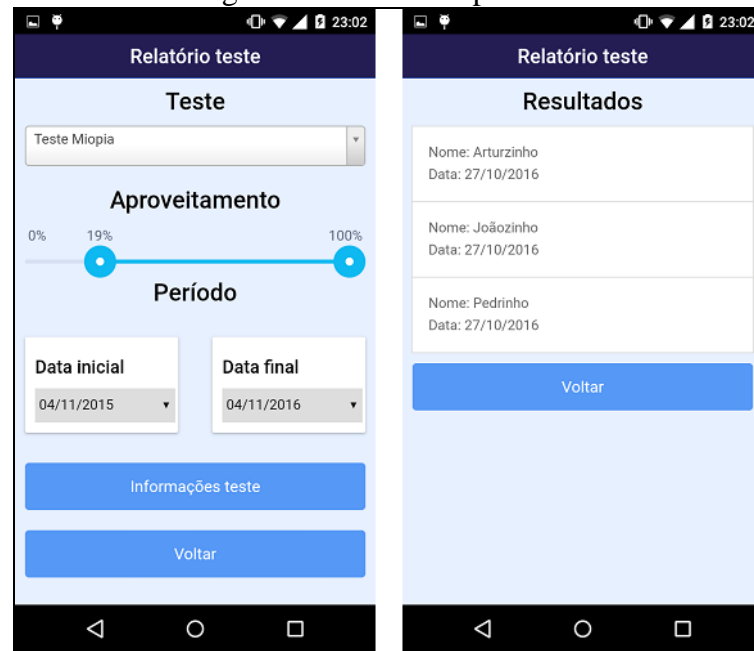
Figura 37 - Relatório do aluno em determinado teste



Fonte: elaborado pelo autor.

Em seguida, há o relatório por teste, retratado pela Figura 38. Neste relatório o usuário seleciona um teste, determina qual a faixa de aproveitamento que quer visualizar e o período de datas. Por padrão, a data final vem preenchida com o dia de hoje e a data inicial de um ano anterior à data atual. A partir desses filtros, serão listados todos os resultados, sendo possível selecionar cada teste para visualizar as informações conforme o relatório do aluno comentado anteriormente.

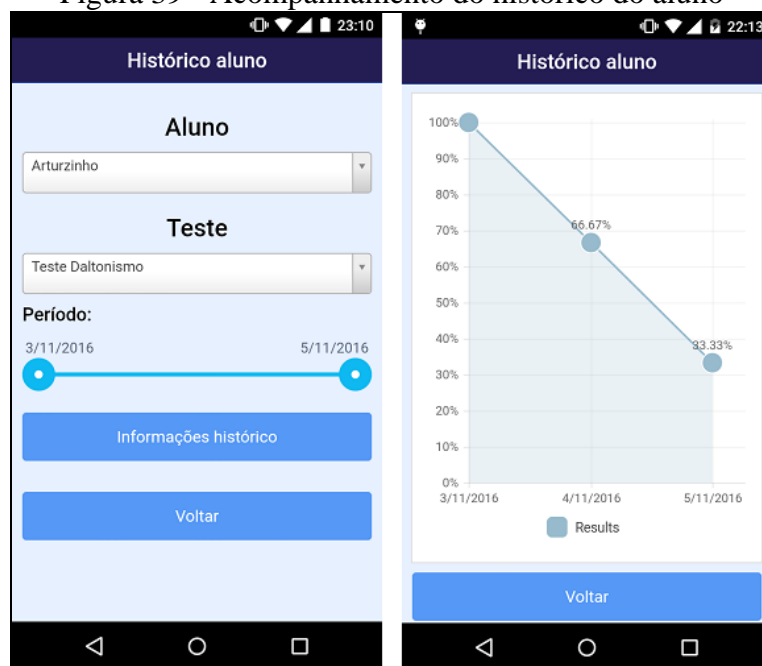
Figura 38 - Relatório por teste



Fonte: elaborado pelo autor.

A última opção de relatório é a de acompanhamento do histórico do aluno em certo teste, para mapear a evolução do mesmo a fim de detectar melhorias ou pioras no desempenho. A Figura 39 apresenta o relatório de acompanhamento, onde o usuário seleciona um aluno, um teste que ele realizou e a faixa de datas que deseja conferir e em seguida a plataforma disponibiliza uma visualização da evolução do aluno na forma de gráfico. É possível clicar nos pontos do gráfico para ver os detalhes do teste conforme o primeiro relatório do aluno mencionado.

Figura 39 - Acompanhamento do histórico do aluno



Fonte: elaborado pelo autor.

3.4 ANÁLISE DOS RESULTADOS

Nesta seção serão apresentados os experimentos realizados com a plataforma. Na seção 3.4.1 é feita a análise dos resultados de uma lista de tarefas enviadas aos usuários. A seção 3.4.2 aborda a análise quanto à usabilidade da plataforma. Na seção 3.4.3 são demonstrados testes de compatibilidade da plataforma com diferentes dispositivos móveis e versões do Android, também foram verificados possíveis problemas de layout da plataforma e, por fim, executados experimentos de carregamento de dados do servidor. Na seção 3.4.4 é realizada a comparação das funcionalidades da plataforma com os trabalhos correlatos.

3.4.1 Análise dos resultados da lista de tarefas

Após a análise de perfil dos usuários, foi realizada a avaliação dos resultados obtidos a partir da lista de tarefas (Apêndice A), aplicados a onze pessoas. Todos os usuários realizaram a instalação do aplicativo com sucesso. O perfil dos usuários consistiu de maioria de homens e apenas uma mulher. Dos avaliados, 36% tinham abaixo de dezoito anos, 36% entre dezoito e trinta anos de idade e 28% tinham acima de quarenta e cinco anos. A maioria dos usuários tem acesso diário a um smartphone com algumas exceções, dos que possuem quarenta e cinco anos ou mais.

A aplicação dos testes foi feita por todos os voluntários, alguns sugeriram melhorias como adicionar um link para a tela de cadastro de aluno caso o usuário não ache o mesmo no campo de busca. Houve alguns problemas percebidos na aplicação do teste onde havia

demora no carregamento das imagens devido à má qualidade da internet no local, o que por vezes, impossibilitou a execução do teste.

A visualização dos testes pré-cadastrados na plataforma ocorreu assim como a aplicação dos testes. Eles compartilham da mesma apresentação das informações dos testes que são apresentadas na aplicação dos mesmos e por isso algumas vezes apresentaram ligeira lentidão no carregamento de imagem por conta do plano de internet do usuário. A possibilidade de verificar o conteúdo dos testes com todas suas informações foi em geral bem recebida pelos usuários, conforme ilustrado no Quadro 14.

Quadro 14 - Respostas às tarefas aplicadas aos administradores e executadores de testes

Tarefas/Respostas	Sim	Sim, mas com dificuldade	Não
Instalação do aplicativo	100%		
Realizar login	100%		
Cadastrar um novo teste	66%	33%	
Acessar um teste cadastrado	81%	19%	
Acessar opções de usuário	100%		
Cadastrar novo aluno	91%	9%	
Aplicar teste	80%	10%	10%
Consultar relatórios e resultados	81%	19%	

Fonte: elaborado pelo autor.

Com relação às opções de usuário, que envolve apenas a alteração de senha para os executadores, não houveram problemas. Para administradores há opções extras de criar e remover usuários, nestes casos também não houveram relatos de contratemplos.

O cadastro de pessoas na plataforma ocorreu sem problemas evidentes. Há a ressalva de que alguns usuários levantaram, que entendem que seria benéfico a inserção de fotos de perfil para o avaliado utilizando a câmera do dispositivo, que também facilitaria na identificação dos indivíduos na hora de selecionar para operação de aplicação de teste e visualização de relatórios.

A criação de testes por parte do administrador teve êxito em todos os casos, conforme Quadro 14. Alguns pontos foram assinalados, pelos 33% que afirmaram ter certa dificuldade, como permitir ao usuário realizar alterações nos testes após o cadastro, ou alterar a ordem das questões caso julgasse necessário. Também foi sugerido a utilização da câmera do dispositivo para adicionar as imagens as testes.

A aplicação dos testes funcionou corretamente em 90% dos casos e o restante teve problema de internet que impossibilitou o prosseguimento do teste. Nesse quesito, os usuários deram várias ideias para melhorar o processo, começando por filtrar alunos apenas de uma certa instituição de ensino que o usuário pertence, opção que não foi implementada na plataforma, mas que poderia melhorar a experiência no uso da aplicação móvel. Outra

solicitação dos usuários foi que a imagem fosse mostrada por alguns segundos utilizando a tela inteira do dispositivo, minimizando a questão do tamanho pequeno de tela de alguns dispositivos. Outra questão sugerida foi inserir a funcionalidade de realizar o mesmo teste para cada olho, pois é possível que o aluno tenha anomalias visuais em apenas um dos olhos, ou cada olho possuir um distúrbio diferente. A forma utilizada pela plataforma foi julgada de forma geral como adequada, mas que seria possível realizar melhorias, conforme levantado acima.

Os testes de acuidade visual foram executados com os usuários não utilizando óculos, os avaliados acima de quarenta e cinco anos tiveram uma certa dificuldade em questões que apresentavam símbolos menores e obtiveram uma taxa de aproveitamento entre 70% e 80%. Outro usuário mais jovem, na faixa entre dezoito e trinta anos realizou o teste de acuidade visual duas vezes, uma com e outra sem óculos, e teve aproveitamento de 100% nas duas ocasiões. Vale ressaltar que o teste de daltonismo foi aplicado a um usuário que possui esse distúrbio, que evidenciou um certo nível da anomalia onde conseguiu uma taxa de acerto de 76%. Este usuário acabou apontando um detalhe que não foi levado em consideração na criação dos testes, que seria o cadastro de alternativas “nenhuma das opções”, pois o que conseguia visualizar não condizia com nenhuma das alternativas. Outro ponto levantado é a opção de não trabalhar com alternativas nas questões, dar opção ao avaliado de ter uma resposta aberta.

Por fim, os usuários verificaram os relatórios disponibilizados e a maioria conseguiu visualizar todos os resultados desejados, com apenas algumas exceções devido à lentidão no carregamento de figuras. Alguns usuários sugeriram que a plataforma seja capaz de gerar um pré-diagnóstico baseado nos resultados obtidos e que essas informações possam ser enviadas por email para um responsável para encaminhar para profissionais especializados.

3.4.2 Análise quanto à usabilidade e funcionalidades

Foi aplicado um questionário fechado a onze pessoas, três administradores e oito executadores de testes, com sete perguntas relacionadas à usabilidade do aplicativo e oito perguntas referentes às funcionalidades e interesse dos usuários pelo diagnóstico de anomalias visuais. O Quadro 15 apresenta as respostas das perguntas referentes à usabilidade.

Quadro 15 - Avaliação de usabilidade da plataforma

Perguntas / Critérios de avaliação	Concordo totalmente	Concordo parcialmente	Não concordo nem discordo	Discordo parcialmente	Discordo totalmente
1. Foi fácil encontrar as informações que você precisou	72%	18%		10%	
2. O design da interface do aplicativo é atraente	80%	10%	10%		
3. A organização dos menus e comandos de ação (como botões e links) é lógica, permitindo encontrá-los facilmente na tela	81%	19%			
4. Foi fácil navegar nos menus e telas do aplicativo	72%	19%		9%	
5. É fácil lembrar como fazer as coisas no aplicativo	91%	9%			
6. Os símbolos e ícones são claros e intuitivos	80%	10%	10%		
7. O tamanho da tela é suficiente para realizar a avaliação	36%	27%	18%	10%	9%

Fonte: elaborado pelo autor.

Conforme as respostas do Quadro 15, a maioria dos usuários achou fácil encontrar as informações necessárias e acharam o design da interface atraente. Concordaram que foi simples navegar nos menus e telas. Nem todos os usuários possuem contato diário com smartphones e os testes da plataforma ocorreram sem instruções de uso além das que constavam nos formulários de questionamentos enviados aos testadores. Um detalhe apenas ressaltado por alguns usuários, é que gostariam de um maior contraste de cores nos menus, para facilitar a leitura das opções oferecidas.

A organização dos menus e a lógica de navegação em sua maioria teve grande aceitação assim como os ícones que são considerados claros e intuitivos. A maioria dos usuários concorda que uma vez realizado o processo inteiro de testes e visualização de relatórios, lembrar dos passos executados é considerado fácil.

A ressalva percebida foi em questão ao tamanho da tela. Alguns testes foram executados em celulares com telas de quatro polegadas que pode ocasionar um certo comprometimento da experiência de uso da plataforma. Alguns usuários concordam que o tamanho também pode influenciar caso a mesma pessoa seja avaliada por dois dispositivos de tamanhos diferentes. A maioria entende que é possível realizar testes desta forma, porém um teste realizado por esta plataforma não poderia servir como um diagnóstico inteiramente seguro e que necessita de um supervisionamento de um profissional para que o caso do avaliado seja melhor verificado.

Na sequência são analisadas as respostas das questões relacionadas às funcionalidades do aplicativo e sua eficácia em relação ao diagnóstico de anomalias visuais. O Quadro 16 exibe as respostas obtidas.

Quadro 16 - Respostas relacionadas às funcionalidades e à eficácia do aplicativo

Perguntas / Critérios de avaliação	Concordo totalmente	Concordo parcialmente	Não concordo nem discordo	Discordo parcialmente	Discordo totalmente
1. Você conseguiu compreender que os testes servem para diagnosticar e acompanhar os resultados do aluno	72%	18%	10%		
2. Você se sente mais motivado para fazer o acompanhamento dos resultados no aluno?	72%	10%	18%		
3. Você achou importante ver os resultados do aluno de maneira detalhada?	91%		9%		
4. Às vezes eu não sei o que fazer com este aplicativo			9%	55%	36%
5. Você precisaria do apoio de uma pessoa para usar este aplicativo	9%	9%	27%	9%	44%
6. O aplicativo em algum momento parou inesperadamente	18%		18%		64%
7. Você acha que os relatórios apresentam informações suficientes para identificar anomalias visuais	27%	18%	9%	36%	9%
8. Você acha que este aplicativo serve como alternativa para identificar anomalias visuais	27%	46%	27%		

Fonte: elaborado pelo autor.

A maioria dos usuários conseguiu compreender a utilidade da plataforma para realizar diagnósticos de anomalias visuais. Quase na mesma proporção, os usuários se sentem mais motivados para realizar os testes e acompanhar o resultado dos alunos. Onde, 91% dos usuários entendem a importância de visualizar os relatórios com detalhamento dos resultados e sugerem também dados extras como o usuário que aplicou o teste e em qual dispositivo foi realizado.

Houve relatos de 18% dos usuários de que a plataforma parou de responder, em carregamento de imagens e acredita-se que tenha sido pela falta de qualidade de sinal de internet. Os usuários tiveram opiniões bastante divergentes quanto à suficiência ou não de informações nos relatórios para diagnosticar uma anomalia visual. Eles entendem que os testes fornecem dados para um ponto de início de investigação mas ainda não é uma informação conclusiva.

Ao final, 73% dos usuários acreditam que a plataforma venha trazer benefícios e que serve como opção para diagnosticar anomalias visuais. Mesmo que não seja um diagnóstico conclusivo, porém oferece insumo suficiente para filtrar alunos que possuem anomalias mais evidentes.

A aplicação dos testes de hipermetropia seguiram regras de distância máxima de 50 cm e testes de miopia tiveram distância mínima de 2 metros, para 9 dos 11 casos testados. Para os testes foi requisitado que os usuários que usassem óculos os retirassem. Não foram evidenciados casos de hipermetropia. Já para o teste de miopia, foram evidenciados 2 casos, onde os usuários já faziam uso de óculos, que totalizaram de 50 a 70% de acertos. Houve testes de daltonismo que seguiram a recomendação de Ishihara (1972) também em 9 de 11 casos, com distância média de 75 cm. Desses não foi possível verificar casos de usuários que possuíam este distúrbio. Dos outros 2 casos foi percebido que um usuário que já sabia de seu distúrbio que realizou um auto teste, teve um aproveitamento no teste de 72%. Em todos os testes, buscou-se estar em ambiente bem iluminado para que isso não fosse um fator que comprometesse a experiência e resultados dos testes.

3.4.3 Testes de carregamento de dados, layout e compatibilidade

O experimento de carregamento de dados foi realizado em dispositivo móvel com conexão à internet via 3G e com a plataforma recém instalada. Na base de dados já haviam testes e alunos cadastrados. Um dos casos de testes mais comuns da aplicação é o usuário realizar o login e aplicar um teste a um aluno. Realizando este procedimento no dispositivo na primeira vez, foi percebido uma pequena lentidão na renderização de cada questão entre 1 a 2 segundos na maioria dos casos, principalmente pela qualidade da conexão da internet. Realizando este mesmo teste em conexão que pertence à mesma rede do servidor esta renderização ocorreu quase que instantaneamente. Ao realizar o mesmo teste novamente sem fechar o aplicativo da plataforma, com um aluno diferente, por exemplo, as questões eram imediatamente carregadas sem espera, isso porque todas as informações necessárias para executar o teste já se encontravam no cache da plataforma. Questões de testes que não possuem imagens cadastradas são carregadas com muita rapidez. Assim conclui-se que a conexão à internet e o fato dos testes possuírem imagens são fatores determinantes na fluidez, pelo menos, na primeira execução de um teste e que o cache contribui de maneira significativa para a aplicação dos testes e um menor consumo de dados de internet.

Já o experimento de compatibilidade foi realizado utilizando oito dispositivos Android,

possibilitando avaliar a compatibilidade da plataforma em diferentes hardwares e versões do sistema operacional Android. Os dispositivos utilizados estão listados no Quadro 17. A maioria eram smartphones, sendo apenas um *tablet*.

Quadro 17 - Dispositivos utilizados no experimento

Dispositivo	Versão do Android	Tamanho da tela	Resolução
LG Optimus L3 II	4.1.1	3.2"	(320x240)
Motorola Moto X 2013 (dispositivo de desenvolvimento)	5.1	4.7"	(1280x720)
Motorola Xoom	4.1.1	10.1"	(1280x800)
Nexus 4	4.4.4	4.7"	(1280x768)
Nexus 5	5	4.95"	(1920x1080)
Samsung Galaxy S5	4.4.4	5.1"	(1920x1080)
Samsung Galaxy S6	6.0	5.1"	(2560x1440)
Sony Xperia Z	4.3	5"	(1920x1080)

Fonte: elaborado pelo autor.

O aplicativo da plataforma foi instalado com sucesso em todos os aparelhos e todas as funcionalidades puderam ser utilizadas. Entretanto, foi possível perceber alguns problemas de layout relacionados aos menus do sistema na versão instalada no tablet Motorola Xoom, que possui tela maior, mas que continuou funcional permitindo executar todas as ações dos menus. Outro ponto de atenção é que no smartphone LG Optimus L3 II, que possui resolução de tela muito baixa comparada aos outros dispositivos, as figuras mostradas nas questões dos testes que possuem altura maior que 280 pixels acabam não aparecendo inteiramente na tela, obrigando o usuário a scrollar para baixo para visualizar o resto da imagem. Por fim, a plataforma mostrou-se compatível com versões do Android desde a versão 4.1.1 até 6.0.

3.4.4 Comparação com os trabalhos correlatos

O Quadro 18 apresenta um comparativo entre a plataforma desenvolvida e os trabalhos correlatos. Os dados utilizados para comparação são baseados nos experimentos realizados.

Quadro 18 - Comparação com os trabalhos correlatos

Características / trabalhos	Iris (2016)	Paula Júnior (2014)	Soares (2009)	Healthcare4 mobile (2016)	Peek Vision (2016)
Portável para dispositivos móveis	Sim	Sim	Não	Sim	Sim
Open Source	Sim	Sim	Não	Não	Não
Exame de acuidade visual	Sim	Não	Sim	Sim	Sim
Exame de daltonismo	Sim	Sim	Sim	Sim	Sim
Exame de glaucoma	Não	Não	Não	Sim	Não
Exame de catarata	Não	Não	Não	Sim	Não
Exame de retinopatia diabética	Não	Não	Não	Não	Sim
Exame de pressão sanguínea via imagem de fundo do olho	Não	Não	Não	Não	Sim
Público alvo	Crianças em idade escolar	Crianças em idade escolar	Crianças em idade escolar	Todos	Todos
Permite cadastrar novos testes	Sim	Não	Não	Não	Sim
Relatórios detalhados de cada teste e evolução	Sim	Não	Não	Não	Sim
Encaminhamento automático para o profissional adequado	Não	Não	Não	Não	Sim

Fonte: elaborado pelo autor.

A plataforma Iris possui uma distribuição para dispositivos móveis assim como a maioria dos trabalhos correlatos com exceção de Soares (2009). O diferencial deste trabalho é que se pode cadastrar novos testes, inclusive com um estudo de caso pode ser possível cadastrar testes de glaucoma e catarata como o aplicativo de Healthcare4mobile (2016), e esses testes não necessariamente devem ser relacionadas à deficiências visuais. Outra vantagem é o acompanhamento por vários relatórios diferentes dos resultados dos alunos nos testes, podendo visualizar a evolução do avaliado ao longo do tempo, porém não possibilita ainda o encaminhamento automático de usuários para médicos especializados como o Peek Vision (2016) executa.

4 CONCLUSÕES

Segundo Toledo et al. (2010) as deficiências visuais podem ser consideradas um problema de saúde pública, apesar de não haver um grande interesse político no controle e prevenção, elas afetam diretamente a saúde e o rendimento das pessoas, podendo limitar ou até tornar uma pessoa inválida no caso de cegueira completa. Toledo et al. (2010) ainda citam

A implementação dos programas de detecção de baixa acuidade visual e de prevenção de problemas oftalmológicos em países desenvolvidos mostram que os custos dessas ações são incomparavelmente menores do que aqueles representados pelo atendimento a portadores de distúrbios oculares”, ressaltando que a prevenção além de ser extremamente benéfica para o paciente, que terá melhor saúde da visão, o Estado economiza muitos gastos que teria para tratar uma pessoa que apresenta deficiência visual (TOLEDO et al., 2010, p. 416).

Em relação aos trabalhos correlatos, percebe-se uma preocupação em desenvolver soluções para dispositivos móveis, dada a popularidade desses itens, onde cada vez mais se desprende da arquitetura de *software* estilo *desktop* do trabalho de Soares (2009) e direcionando cada vez mais para arquiteturas móveis como a ferramenta de Paula Júnior (2014) e o Eye Test – Eye Exam (HEALTHCARE4MOBILE, 2015). Vale lembrar que nenhum desses aplicativos substitui uma consulta com um oftalmologista, mas oferecem um pré-diagnóstico que direcionam o examinado para um médico especializado caso o *software* detecte uma anomalia.

Os trabalhos de Soares (2009) e Peek Vision (2016) possuem uma certa similaridade com a plataforma desenvolvida neste trabalho, sendo a segunda uma solução já concreta e sendo utilizada amplamente em comunidades carentes da África e sudeste da Ásia. Iris vem com uma proposta de melhorar o acompanhamento da evolução de possíveis anomalias em crianças de idade escolar e necessita de uma pós avaliação por parte de um profissional especializado, enquanto o Peek Vision (2016) é capaz de realizar pré-diagnósticos e encaminhar o avaliado a um oftalmologista.

Os resultados dos testes da plataforma apontam que há ainda grandes melhorias que podem tornar a solução mais completa e confiável. Em geral, a plataforma possui boa usabilidade e se pode trabalhar para aprimorar certos pontos. A maioria dos usuários concorda que a plataforma auxilia, porém não deve servir como única fonte de informações para concluir que um aluno possui um diagnóstico adverso. É necessário o acompanhamento da área médica para que seja possível lidar com os casos mais evidentes.

Com o desenvolvimento dessa plataforma, espera-se oferecer amplo suporte às escolas auxiliando na detecção precoce de distúrbios da visão e permitindo que os dados coletados sejam armazenados para análise e, que crianças tenham um tratamento antecipado,

minimizando a insuficiência escolar relacionada a essas condições e proporcionando melhor saúde à população.

4.1 EXTENSÕES

Sugerem-se as seguintes extensões para trabalhos futuros:

- a) realizar agendamento e envio de emails, por exemplo uma vez por mês, para os usuários ou administrador avisando que alguns alunos que possuem rendimento abaixo do esperado;
- b) melhorar compatibilidade com tablets e desenvolver um website com o conteúdo do aplicativo móvel da plataforma;
- c) implementar aspectos de Gamificação para aumentar a motivação e o engajamento dos alunos e usuários;
- d) adicionar cadastro de instituições de ensino e fazer com que usuário possa ver apenas os alunos da própria instituição de ensino;
- e) disponibilizar aplicativo da plataforma para o sistema operacional Android, visto que o código-fonte é o mesmo e basta ter um dispositivo com sistema operacional Apple e uma licença de desenvolvedor Apple;
- f) disponibilizar versão da plataforma na web, para ser utilizado via browsers em computadores convencionais;
- g) permitir alterar os testes, realizando um versionamento do mesmo a fim de manter a versão original para fins de conferência dos resultados dos alunos;
- h) permitir remover os testes mesmo com resultados cadastrados, onde seria adotada a abordagem de escondê-los da lista de testes passíveis de serem aplicados, mas seriam mantidos para realizar a posterior conferência nos relatórios dos alunos;
- i) adicionar informação do sexo do aluno, pois é uma informação relevante para testes de daltonismo.

REFERÊNCIAS

- BICAS, Harley E. A. et al. Acuidade visual. Medidas e notações. **Arquivos Brasileiros de Oftalmologia**, São Paulo, v. 65, n. 3, p. 375-384, jun., 2002.
- CLÍNICA DE OLHOS OFTALMOVILLAS. **Problemas de visão afetam o rendimento escolar**. 2016. Disponível em: <<http://www.ofthalmovillas.com.br/problemas-de-visao-afetam-o-rendimento-escolar/>>. Acesso em: 02 abr. 2016.
- CODECENTRIC. **Ionic: An AngularJS based framework on the rise**. [S. I.], 2016. Disponível em: <<https://blog.codecentric.de/en/2014/11/ionic-angularjs-framework-on-the-rise/>>. Acesso em: 15 nov. 2016.
- COLBLINDOR. **Ishihara's Test for Colour Deficiency: 38 Plates Edition**. 2016. Disponível em: <<http://www.visaolaser.com.br/saude-ocular/teste-de-visao/teste-de-snellen>>. Acesso em: 02 abr. 2016.
- HEALTHCARE4MOBILE. **Eye Examination Tests and Training - Apps**. 2016. Disponível em: <<http://www. http://www.eyexamtest.com/>>. Acesso em: 02 abr. 2016.
- HOSPITAL OFTALMOLÓGICO VISÃO LASER. **Teste de Snellen**. 2016. Disponível em: <<http://www.visaolaser.com.br/saude-ocular/teste-de-visao/teste-de-snellen>>. Acesso em: 02 abr. 2016.
- IONIC. **Ionic Framework**. [S. I.], 2016. Disponível em: <<http://ionicframework.com/>>. Acesso em: 06 nov. 2016.
- ISHIHARA, Shinobu. **Tests For Colour-Blindness**. Tokyo - Kyoto: Kanehara Shuran, 1972. 33 p.
- MAGNI, Matteo. **Building Mobile Applications with HTML/Js/CSS and Cordova**. 2016. Disponível em: <<http://magni.me/presentation-cordova-phonegap-course/>>. Acesso em: 15 nov. 2016.
- MELO, Débora G. et al. Os "daltônicos" e suas dificuldades: condição negligenciada no Brasil?. **Physis: Revista de Saúde Coletiva**, Rio de Janeiro, v. 24, n. 4, p. 1229-1253, out./dez., 2014.
- MINISTÉRIO DA SAÚDE. Secretaria de Atenção à Saúde. Departamento de Ações Programáticas Estratégicas. **Diretrizes de atenção à saúde ocular na infância: detecção e intervenção precoce para prevenção de deficiências visuais**. Brasília, 2013.
- PAULA JÚNIOR, Carlos A.. **Proposta de um aplicativo móvel em auxílio a indivíduos com discromopsia baseado em um estudo qualitativo**. 2014. 87 f. Proposta de TCC (Graduação) - Curso de Sistemas de Informação, Faculdade Católica Salesiana do Espírito Santo, Vitória, 2014.
- PEEK VISION. **Peek Beta**. [S. I.], 2016. Disponível em: <<http://www.peekvision.org/>> Acesso em: 15 nov. 2016.
- PHONEGAP DOCS. **Phonegap Documentation**. [S. I.], 2016. Disponível em: <<http://docs.phonegap.com/>>. Acesso em: 15 nov. 2016.
- ROCHA, Maria N. A. M. et al. Prevalência de doenças oculares e causas de comprometimento visual em crianças atendidas em um Centro de Referência em Oftalmologia do centro-oeste do Brasil. **Revista Brasileira de Oftalmologia**, Rio de Janeiro, v. 73, n. 4, p. 225-229, jul./ago., 2014.

SEDPD. **Resultados Preliminares da Amostra | Censo 2010**. 2010. Disponível em: <<http://www.pessoacomdeficiencia.gov.br/app/indicadores/censo-2010>>. Acesso em: 02 abr. 2016.

SOARES, Fabrício C. **Sistema de triagem visual e auditiva de crianças em idade escolar, conectado a um banco de dados**. 2009. 140 f. Tese (Doutorado) - Curso de Engenharia Mecânica, Universidade Federal de Minas Gerais, Belo Horizonte, 2009.

TALEB, Alexandre et al.. **As Condições de Saúde Ocular no Brasil**. São Paulo: Conselho Brasileiro de Oftalmologia, 2012. 35 p.

TELLER, Davida Y. First glances: the vision of infants: The Friedenwald lecture. **Investigate Ophthalmology Visual Science**, [S.l.], v. 38, n. 4, p. 2183-2203, out., 1997.

TIZEN. **Tizen, Connect Everything**. [S. l.], 2016. Disponível em: <<https://www.tizen.org/>>. Acesso em: 15 nov. 2016.

TOLEDO, Carolina C. et al. Detecção precoce de deficiência visual e sua relação com o rendimento escolar. **Revista da Associação Médica Brasileira**. São Paulo, v. 56, n. 4, p. 415-419, 2010.

TRICE, Andrew **PhoneGap Explained Visually**. [S.l.], 2012a. Disponível em <<http://phonegap.com/2012/05/02/phonegap-explained-visually/>>. Acesso em 04 nov. 2016.

ZAPAROLLI, Marcio et al. Avaliação da acuidade visual Snellen. **Arquivos Brasileiros de Oftalmologia**, São Paulo, v. 72, n. 6, p. 783-788, nov./dez., 2009.

APÊNDICE A – Lista de tarefas e questionário de avaliação de usabilidade

Neste apêndice constam os questionários de perfil de usuário, a lista de tarefas executadas pelos voluntários e o questionário de usabilidade aplicado ao final do experimento.

Quadro 19 - Questionário de perfil do usuário

Iris – Plataforma para triagem de deficiências visuais em crianças com idade escolar

Olá!

Você está sendo convidado a participar de uma avaliação do aplicativo de triagem de deficiências visuais em crianças com idade escolar Iris. Este aplicativo tem como principal objetivo, disponibilizar uma forma de diagnosticar anomalias visuais e possibilitar o acompanhamento da evolução do distúrbio nos alunos.

Os testes de deficiências visuais são utilizados pelos oftalmologistas para acompanhar a saúde da visão das pessoas, permitindo o diagnóstico precoce de distúrbios como Daltonismo, Miopia, Hipermetropia, entre outros. Essas patologias podem afetar no rendimento de uma pessoa e se não tratado pode se agravar comprometendo a visão do indivíduo.

Gostaríamos de sua contribuição para avaliar o aplicativo que foi desenvolvido. Para tanto, pedimos que você responda ao questionário de perfil de usuário e, em seguida, realize as tarefas listadas informando se foi possível ou não executar cada uma delas. Ao final, é apresentado um questionário para você realizar uma avaliação geral do aplicativo.

PERFIL DE USUÁRIO

Sexo: () Feminino () Masculino

Idade:

() Tenho menos de 18 anos

() Tenho entre 30 e 45 anos

() Tenho entre 18 e 30 anos

() Tenho mais de 45 anos

Profissão: _____

Você utiliza celular com qual frequência?

() todos os dias

() A cada 15 dias

() 2 a 3 vezes por semana

() 1 vez ao mês

() 1 vez por semana

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

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

Quadro 20 - Lista de tarefas a serem executadas pelo administrador

LISTA DE TAREFAS DO ADMINISTRADOR

A seguir, é apresentada uma lista com 8 atividades que têm por objetivo avaliar o aplicativo desenvolvido. Por gentileza, realize o que é solicitado em cada tarefa, indicando se a mesma foi concluída ou não.

1) Instalação do aplicativo

Baixe o aplicativo e faça sua instalação.

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

2) Realizar login

Necessário estar conectado à internet e, caso seja o primeiro acesso, autenticar-se no aplicativo utilizando como credenciais de usuário e senha o termo “admin”.

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

3) Cadastrar um novo teste

Ao abrir o aplicativo, navegue pela tela inicial e inicie o cadastro de um novo teste. Adicione questões ao teste, cada uma com imagens opcionais oriundas do dispositivo do usuário e suas respectivas alternativas, assinalando uma delas como a correta. Ao cadastrar as questões é possível removê-las se necessário, na tela que mostra todas as questões pode-se deslizar a questão para a esquerda revelando a opções de remover. Por fim, salvar o teste pelo botão.

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

4) Acessando o teste cadastrado

Após o fim do cadastro do teste, navegue pela tela inicial e acesse o teste que foi criado na tarefa anterior. Visualizar as informações do teste, bem como questões, suas alternativas corretas e imagens.

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

5) Opções de usuário

Através do menu principal do aplicativo, acesse a opção “Usuários” e realize as opções de cadastrar usuário, remover usuário e alterar sua senha.

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

6) Cadastrar novo aluno

Na tela principal do aplicativo, acesse a opção Novo aluno e adicione um aluno ao aplicativo preenchendo os campos necessários.

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

7) Aplicar teste.

A partir do menu principal, selecionar a opção “Aplicar teste”, escolher o aluno e teste cadastrados e iniciar o processo. Mostrar as questões ao aluno e selecionar as alternativas corretas de cada questão apresentada e seguir o processo até o fim, quando serão mostradas informações sobre o aproveitamento do aluno no teste e feito isso, salvar o resultado.

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

8) Consultar relatórios de resultados

Na tela principal do aplicativo, acesse a opção “Relatórios” e escolher qual modalidade de relatório deseja conferir. Na opção “Por aluno” é possível visualizar um resultado individual de um teste em uma data. Na opção “Por teste” é possível verificar todos os resultados de todos os testes que respeitem os parâmetros definidos pelo usuário. Por fim a opção “Histórico aluno” onde é possível acompanhar a evolução do aluno caso tenha dois ou mais resultados cadastrados para um determinado teste.

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

Quadro 21 - Lista de tarefas a serem executadas pelos executadores

LISTA DE TAREFAS DO EXECUTADOR

A seguir, é apresentada uma lista com 7 atividades que têm por objetivo avaliar o aplicativo desenvolvido. Por gentileza, realize o que é solicitado em cada tarefa, indicando se a mesma foi concluída ou não.

1) Instalação do aplicativo

Baixe o aplicativo e faça sua instalação.

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

2) Realizar login

Necessário estar conectado à internet e se autenticar no aplicativo utilizando as credenciais distribuídas pelo administrador.

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

3) Acessando o teste cadastrado

Após o fim do cadastro do teste, navegue pela tela inicial e acesse o teste que foi criado na tarefa anterior. Visualizar as informações do teste, bem como questões, suas alternativas corretas e imagens.

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

4) Opções de usuário

Através do menu principal do aplicativo, acesse a opção “Usuários” e realize a alteração de senha, informando a senha antiga e preencher a nova senha com a confirmação.

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

5) Cadastrar novo aluno

Na tela principal do aplicativo, acesse a opção Novo aluno e adicione um aluno ao aplicativo preenchendo os campos necessários.

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

6) Aplicar teste.

A partir do menu principal, selecionar a opção “Aplicar teste”, escolher o aluno e teste cadastrados e iniciar o processo. Mostrar as questões ao aluno e selecionar as alternativas corretas de cada questão apresentada e seguir o processo até o fim, quando serão mostradas informações sobre o aproveitamento do aluno no teste e feito isso, salvar o resultado.

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

7) Consultar relatórios de resultados

Na tela principal do aplicativo, acesse a opção “Relatórios” e escolher qual modalidade de relatório deseja conferir. Na opção “Por aluno” é possível visualizar um resultado individual de um teste em uma data. Na opção “Por teste” é possível verificar todos os resultados de todos os testes que respeitem os parâmetros definidos pelo usuário. Por fim a opção “Histórico aluno” onde é possível acompanhar a evolução do aluno caso tenha dois ou mais resultados cadastrados para um determinado teste.

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

Quadro 22 - Questionário de usabilidade

QUESTIONÁRIO DE AVALIAÇÃO DO APLICATIVO

Após a utilização do aplicativo, você está convidado a responder um questionário de avaliação do mesmo. As respostas deverão ser feitas na tabela abaixo observando às impressões obtidas com a utilização do aplicativo. Você deve responder preenchendo uma das alternativas. Após o questionário com perguntas objetivas, é apresentado um espaço para comentários gerais sobre a ferramenta e sugestões de melhorias.

Perguntas / Critérios de avaliação	Concordo totalmente	Concordo parcialmente	Não concordo nem discordo	Discordo parcialmente	Discordo totalmente
Foi fácil encontrar as informações que você precisou					
O design da interface do aplicativo é atraente					
A organização dos menus e comandos de ação (como botões e links) é lógica, permitindo encontrá-los facilmente na tela					
Foi fácil navegar nos menus e telas do aplicativo					
É fácil lembrar como fazer as coisas no aplicativo					
Os símbolos e ícones são claros e intuitivos					
O tamanho da tela é suficiente para realizar a avaliação					
Você conseguiu compreender que os testes servem para diagnosticar e acompanhar os resultados do aluno					
Você se sente mais motivado para fazer o acompanhamento dos resultados no aluno?					
Você achou importante ver os resultados do aluno de maneira detalhada?					
Às vezes eu não sei o que fazer com este aplicativo					
Você precisaria do apoio de uma pessoa para usar este aplicativo					
O aplicativo em algum momento parou inesperadamente					
Você acha que os relatórios apresentam informações suficientes para identificar anomalias visuais					
Você acha que este aplicativo serve como alternativa para identificar anomalias visuais					

Qual é a sua opinião sobre o aplicativo quanto ao seu uso e funcionalidades? Fique a vontade para fazer críticas e sugestões.
