

DisfagiaAPP: Aplicativo Multiplataforma para auxílio no rastreamento de risco da Disfagia Orofaríngea

Isac Velozo de Castro Aguiar
Pós-Graduando em Desenvolvimento de Aplicações e
Games para Dispositivos Móveis
Instituto Federal de Educação, Ciência e Tecnologia da
Bahia
Salvador-BA, Brasil
isacaguiar@gmail.com

Prof. Dr. Luiz Claudio Machado dos Santos
Departamento de Computação
Instituto Federal de Educação, Ciência e Tecnologia da
Bahia
Salvador-BA, Brasil
luizmachad@gmail.com

Resumo— Este artigo apresenta a especificação e implementação de uma solução que possui como núcleo um aplicativo multiplataforma para teste de rastreamento da disfagia orofaríngea, denominado DisfagiaApp. A proposta contempla a integração de três aplicações: aplicativo multiplataforma contendo informações sobre a disfagia orofaríngea e formulário para teste de rastreamento; página web contendo informações sobre a disfagia, formulário para teste de rastreio e resultado das pesquisas; e uma Interface de Programação de Aplicativos (API – *Application Programming Interface*).

Palavras-Chaves: *disfagia, aplicativo, dispositivo móvel, api, mhealth, multiplataforma, Flutter*

I. INTRODUÇÃO

O desenvolvimento global sustentável guiado pelos Objetivos de Desenvolvimento Sustentável (ODS), estabelecidos pela Organização das Nações Unidas (ONU), estimulam a criação de pesquisas em diversas áreas. Com a realização de pesquisas nas áreas da saúde e tecnologia surgiu a telemedicina por volta do século XIX, ampliado para *Mobile Health* ou saúde móvel, doravante *MHealth*, no século XX [1].

O *MHealth* tem como principal objetivo ampliar o acesso à informação e a serviços de saúde que promovem o bem-estar pessoal, cuidados preventivos e gestão de doenças crônicas [1]. Diversas tecnologias são envolvidas nesse processo, que variam desde serviços de mensagem até videogames ou aplicativos para *smartphones*.

Atualmente existem mais de 242 milhões de *smartphones* no Brasil [2]. A popularização dos dispositivos móveis fomenta o desenvolvimento de

soluções digitais que cabem na palma da mão. Na área da saúde estas soluções têm se mostrado uma excelente aliada no apoio à tomada de decisão, monitoramento, promoção de cuidados e maior adesão aos tratamentos da saúde.

As aplicações voltadas para o conceito *MHealth* descrevem a importância na qual os aplicativos, além de atender o comportamento funcional esperado relacionado ao tema proposto, devem oferecer uma boa experiência para o usuário, um bom desempenho e uma boa estabilidade, como também oferecer privacidade e segurança [3].

A adoção apropriada de padrões de interação e *design* visual devem proporcionar uma experiência ao usuário consistente e intuitiva, pois a usabilidade é considerada um fator de extrema importância e possui influência no nível de aceitabilidade do usuário em relação ao sistema [4]. É importante a utilização de princípios de *User Interface* ou interface de usuário (UI) e *User Experience* ou experiência de usuário (UX) para tornar a experiência mais agradável.

As características da usabilidade são importantes para a implementação de um sistema, mas outros critérios devem ser contemplados em um projeto de *software* para proporcionar um bom desempenho e uma estabilidade adequada durante o seu uso, é necessário a aplicação de tecnologias adequadas de acordo com a necessidade, assegurando sempre privacidade e segurança.

Diante desse contexto o objetivo deste trabalho é apresentar a especificação e implantação de um aplicativo multiplataforma na área de saúde para possibilitar o rastreamento da disfagia orofaríngea. Nos próximos parágrafos é definido o tema da disfagia aqui proposto.

A disfagia orofaríngea caracteriza-se por um distúrbio da deglutição ou qualquer dificuldade do trânsito do bolo alimentar da boca até o estômago, podendo estar associada a complicações, tais como:

desnutrição, desidratação, pneumonia aspirativa, penetração de saliva ou restos alimentares no vestíbulo laríngeo antes, durante ou após a deglutição [2].

O rastreamento da disfagia orofaríngea pode ser realizado através da aplicação do instrumento de rastreio chamado Rastreamento de Disfagia Orofaríngea em Idosos (RADI). Levando-se em consideração que a disfagia orofaríngea é caracterizada pela combinação de sinais e outros sintomas que colocam o indivíduo em risco nutricional, hídrico e pulmonar, este instrumento de rastreio é um método de triagem direcionado a identificação de indivíduos que reúnem os fatores preditivos para este desfecho e que necessitam de diagnóstico confirmatório [5]. Este artigo não contempla aspectos relacionados à área da saúde, maiores informações podem ser encontradas no trabalho da professora Torres (s.d.) [6].

É importante destacar aqui, que não fez parte do escopo deste trabalho um teste com o usuário final. Foi utilizada a avaliação heurística [7], na qual 10 (dez) pessoas fizeram uso da ferramenta com o intuito de identificar possíveis melhorias, encontrar erros e sugerir novas ideias, visando agregar melhores formas de interação do usuário com a ferramenta. O objetivo aqui é apenas o desenvolvimento do aplicativo, ou seja, a parte técnica. Inclusive não é adentrado de forma detalhada o problema da disfagia.

O artigo está organizado em seções: I. Introdução relata o crescimento das pesquisas na área da saúde e tecnologia, a integração entre elas e o surgimento de aplicativos que atuam neste cenário e a solução proposta neste trabalho; II. Metodologia descreve os passos necessários para a elaboração deste projeto; III. Contexto *MHealth* aborda conceitos sobre o aplicativo e saúde; IV. Contexto Multiplataforma em que se apresenta conceitos que remetem a abordagem de aplicativos do tipo multiplataforma e o *framework Flutter*; V. DisfagiaAPP apresenta a arquitetura, especificação e implementação da solução proposta; e VI. Considerações Finais descreve as etapas e o processo de construção da ferramenta e, também, as lições aprendidas.

II. METODOLOGIA

Trata-se de um estudo descritivo e de produção tecnológica, norteado por princípios da engenharia de *software* para permitir a construção da solução proposta. Este projeto foi desenvolvido em parceria,

por profissionais e estudantes da área de saúde e tecnologia.

Segundo *Sommerville* [8], uma metodologia de desenvolvimento de *software* consiste numa abordagem sistemática da engenharia de *software* composta por processos e etapas bem definidos. Neste projeto foram utilizadas, basicamente, as seguintes etapas: comunicação, planejamento, modelagem, construção, entrega e incremento do *software*. As etapas foram aplicadas com base nos preceitos das metodologias ágeis, com adoção do fluxo incremental e evolucionário de maneira cíclica. A cada encontro realizado com os *stakeholders* o processo era iniciado. A Fig. 1 ilustra o fluxo do processo utilizado.

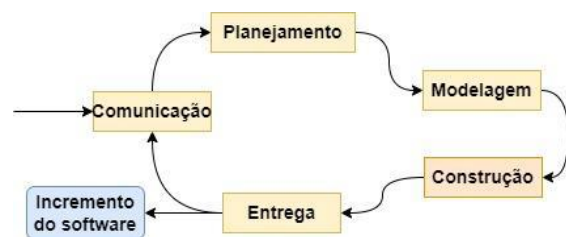


Fig. 1. Fluxo do processo.

Para um gerenciamento eficiente do processo, foi adotado o uso da metodologia *Kanban* [9]. O *Kanban* é um sistema visual de gestão de trabalho que busca conduzir cada tarefa por um fluxo pré-definido de trabalho. Ele é composto por um fluxo de valor, onde unidades de trabalho trafegam da esquerda para a direita. Cada etapa do processo adiciona mais valor ao item, sendo que quando ele chega ao final do processo, ele está “concluído”. Desta forma a cada nova interação ocorria um incremento do *software*.

Para auxiliar no acompanhamento do quadro *Kanban*, o *Trello* [10] foi a ferramenta utilizada. Nele, era possível visualizar o projeto em tarefas e ações, bem como a lógica do fluxo de tarefas baseada no *Kanban*. O quadro *Kanban* neste projeto foi composto por três colunas: “A Fazer”, “Em Progresso” e “Concluído”, conforme ilustração a seguir:

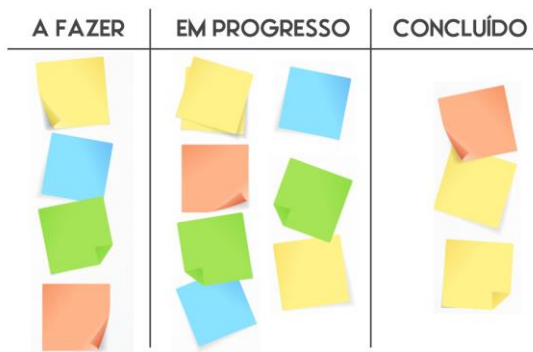


Fig. 2. Quadro Kanban.

Para subsidiar o quadro durante os encontros, as necessidades eram apresentadas e após registradas no *Trello*, eram realizadas as ações necessárias para o incremento do *software*. Por exemplo, no primeiro encontro no qual os profissionais de saúde elencaram as suas necessidades, foi desenvolvido um protótipo, baseado na abordagem evolucionária que passou a ser utilizado nos encontros subsequentes até a conclusão do *software*. A figura a seguir apresenta algumas telas do primeiro protótipo do projeto.



Fig. 3. Primeiro protótipo do projeto.

Após o desenvolvimento do MVP [10], fruto da evolução do protótipo, o *software* foi disponibilizado para alguns usuários para avaliação de forma heurística [7], onde 10 usuários fizeram o uso do aplicativo com o intuito de encontrar erros e sugerir novas ideias, com o objetivo de agregar um maior valor para o aplicativo, bem como proporcionar uma melhor interação do usuário com a ferramenta.

III. CONTEXTO MHEALTH

Conforme discutido na seção I, é notório o crescimento de aplicações na área médica. Desde a telemedicina, surgem sistemas de monitoramento, gerenciamento e controle, sistemas de comunicação, *hardware* para sensoriamento de pacientes e exames clínicos, sistemas de apoio à decisão e/ou trabalho colaborativo que fomentam o surgimento de novas

pesquisas. Essas pesquisas visam contribuir com os serviços prestados, melhorando o atendimento e minimizando os riscos à saúde dos pacientes [1].

A telemedicina permite a prestação de serviços de saúde à distância por meio dos recursos de tecnologia da informação e comunicações [11], o aumento do número de dispositivos móveis e a evolução da telemedicina favorecem o surgimento de aplicativos voltados ao cenário *MHealth*.

A *MHealth* oferece serviços médicos ou de saúde pública com base no apoio tecnológico por meio de dispositivos móveis. A relação da saúde com a tecnologia da informação por meio dos aplicativos e recursos tecnológicos diversos auxilia no apoio à tomada de decisão, monitoramento, promoção de cuidados e maior adesão aos tratamentos de saúde [1]. Favorece, também, o acesso à informação e aos serviços de saúde que promovem o bem-estar pessoal, cuidados preventivos e gerenciamento de doenças crônicas, agenciando a eficiência no atendimento e práticas de gestão para melhorar a saúde da população.

Os aplicativos que englobam promoção, educação e cuidados com a saúde, devem ser fundamentados em evidências e teorias científicas, concebidas por profissionais médicos e concretizadas por profissionais de Tecnologia da Informação e Comunicação. De modo geral, as pesquisas acadêmicas com suas devidas credibilidades impulsionam a criação e evolução de aplicativos.

As pesquisas médicas servem como base para a construção dos aplicativos, e os profissionais de tecnologia da informação viabilizam a construção destes *softwares*. Os aplicativos podem ser desenvolvidos em plataforma nativa, *Android* (*Java* ou *Kotlin*) e *iOS* (*Objective-C* ou *Swift*), ou multiplataforma que possibilita a execução em ambas plataformas com um único código.

IV. APLICATIVOS MULTIPLATAFORMA

Os aplicativos multiplataforma permitem aos desenvolvedores implementarem apenas um código base e executarem em mais de uma plataforma, seja ela *Android* ou *iOS*. Esses tipos de aplicações possuem diversas vertentes, dentre elas, temos as *webapps*, as híbridas, as interpretadas e as *cross-compiled* [12].

Os *webapps*, aplicativos da web móveis, são baseados em *Hyper Text Markup Language* ou Linguagem de Marcação de Hipertexto (HTML), *Cascading Style Sheets* ou Folhas de Estilo em

V. DISFAGIAAPP

Cascata (CSS) e Linguagem de programação interpretada estruturada, doravante JavaScript, nesta vertente o *browser* do dispositivo é o ambiente de execução e, também, possui um comportamento e aparência semelhantes com as páginas *web*.

Aplicações híbridas são compiladas e executadas de forma nativa dentro da aplicação. Essas aplicações são consideradas uma combinação entre as *webapps* e as funcionalidades nativas de cada plataforma [13], possibilitando o acesso a recursos e funcionalidades não disponíveis no ambiente *web*. A *Apache Cordova* [14], *Ionic* [15] e *Adobe PhoneGap* [16] são exemplos de *frameworks* desta natureza.

Na abordagem interpretada, as aplicações se aproximam mais das aplicações nativas, quando consideradas questões de performance, interação e comportamentos. O código é interpretado em tempo de execução e sempre que necessário, faz chamadas às APIs nativas de cada sistema operativo [17], [18]. Os dispositivos *Android* e *iOS* possuem um interpretador *JavaScript*, denominado *JSCore* [19]. O *framework React Native* [20] é um exemplo de *framework* nesta categoria, ele possui acesso às APIs nativas dos dispositivos, permitindo assim criar aplicações com uma UI extremamente próxima de uma nativa. O *React Native* é baseado no *React*, a biblioteca *JavaScript* do *Facebook* [21].

As aplicações *cross-compiled* são escritas em uma linguagem diferente da nativa, em seguida, compilada em código nativo para cada plataforma móvel de suporte [12]. O *framework Flutter*, é um exemplo, ele é um *Software Development Kit* (SDK), isto é, um kit de desenvolvimento de software, do tipo *open-source* (que significa de código-fonte aberto) desenvolvida pelo *Google* [22]. Com o kit de ferramentas de UI do *Google*, é possível criar aplicativos bonitos e nativamente compilados para dispositivos móveis, *web*, *desktop* e incorporados a partir de uma única base de código.

O SDK do *Flutter* possui um conjunto de ferramentas e componentes que são utilizados para construir e compilar a aplicação. A ideia central do *framework* é permitir a construção da UI a partir de *widgets* [23]. Os *widgets* são componentes visuais utilizados para definir a interface de um aplicativo, estruturais, de plataforma e interativos.

Fazendo uma correlação com o LEGO (brinquedo cujo conceito se baseia em partes que se encaixam permitindo muitas combinações), um *widget* é equivalente a uma peça e um conjunto dessas peças representa uma interface completa.

A solução DisfagiaAPP contempla um aplicativo *MHealth* para orientações e teste de rastreamento da disfagia orofaríngea e uma página *web* que além do teste de rastreamento apresenta dados compilados das pesquisas. O rastreio da disfagia orofaríngea é realizado através da aplicação do instrumento de rastreio *RADI*.

O aplicativo para dispositivos móveis é multiplataforma, isto é, compilado para as plataformas *Android* e *iOS*, desenvolvido utilizando o *framework Flutter*. A página *web* segue as Diretrizes de Acessibilidade para Conteúdo *Web* (WCAG) 2.1 [24]. Ambos possuem interface intuitiva baseado em princípios de UI e UX, visando atender a acessibilidade inclusive facilitada também para os idosos.

A ideia é oferecer ao usuário uma experiência agradável durante o uso do aplicativo e da página *web* independente do seu nível/perfil, desmistificando a disfagia orofaríngea com informações acerca do assunto, bem como auxiliando no rastreamento. A ilustração a seguir apresenta os requisitos mínimos para o alcance dos seus objetivos.

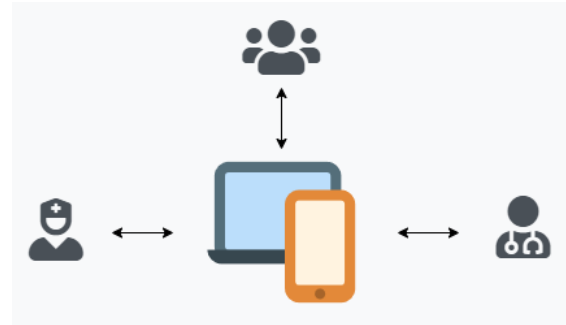


Fig. 4. Objetivo do sistema.

A solução proposta contempla três partes: aplicativo multiplataforma sobre a disfagia orofaríngea com formulário para teste de rastreamento; página *web* acessível contendo informações sobre disfagia orofaríngea, formulário para teste de rastreamento e resultado com base nos testes realizados; e uma API para armazenamento dos dados, que fornecem serviços para o aplicativo e a página *web* permitindo, assim, a utilização do conceito de micro serviço.

A Fig. 5 ilustra a arquitetura geral da solução, que apresenta as partes da solução proposta e suas interações.

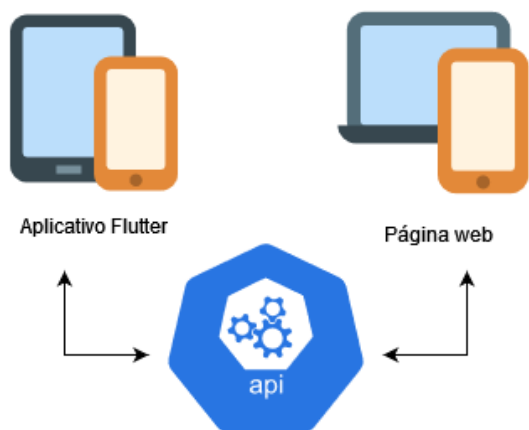


Fig. 5. Arquitetura Geral da Solução.

A página *web* está disponível para acesso no endereço da internet <https://disfagia.isacaguair.com.br/> e o aplicativo pode ser baixado na loja *Google Play* no endereço da internet <https://play.google.com/store/apps/details?id=br.com.disfagia> ou via *link* direto na página *web* através do endereço da internet <https://disfagia.isacaguair.com.br/web/download/disfagia-app.apk>. Ambos acessam a API para obter os dados para apresentação.

A. Aplicativo Multiplataforma

O *framework Flutter* permite a construção de aplicativos para dispositivos móveis multiplataforma *cross-compiled* [12]. O *Flutter* utiliza a linguagem *Dart* [25], posteriormente compilada para plataforma nativa (*Android* e *iOS*).

Além da vantagem do desenvolvimento de um único código, o kit de ferramentas de UI do *Google* oferece recursos de interfaces padronizados que aceleram e simplificam o desenvolvimento [22].

O uso de interfaces mais adequadas e recursos gráficos pode auxiliar na demonstração das informações que são relevantes ao usuário. Estes recursos conseguem tornar mais eficiente o processo de acesso e recuperação da informação [26].

O processo de desenvolvimento da interface foi elaborado a partir de aspectos voltados aos usuários, visando facilitar a compreensão das ações disponíveis na ferramenta até a etapa de conclusão das tarefas, para agregar maior fundamentação teórica foi utilizado o Guia para o Desenvolvimento de Aplicações Móveis Acessíveis [24], desenvolvido e oferecido pelo *Samsung* Instituto de Desenvolvimento para Informática (SIDI) [27].

A interface possui estrutura comum a outros aplicativos, desta forma facilita a compreensão/utilização por parte dos usuários. O *Flutter* permite a criação de uma estrutura básica de layout visual para as telas do aplicativo, neste caso o *Material Design*. “O *Material Design* é um sistema adaptável de diretrizes, componentes e ferramentas que apóiam as melhores práticas de design de interface de usuário” [28].

O *Material Design* permite a criação da estrutura básica de *layout* visual por meio do *widget Scaffold*, além deste *widget* são utilizados outros 2 (dois), 1 (um) para barra de aplicativos, utilizando o *AppBar* e o outro para geração da barra de rolagem no conteúdo apresentado na tela, utilizado o *SingleChildScrollView*.

A tela inicial, como as outras telas do projeto, apresenta a estrutura básica em conjunto com outros 4 (quatro) *widjets*, a logomarca do aplicativo, textos informativos e seus respectivos campos de entrada de dados e botão para confirmação dos dados e navegação no uso do aplicativo.

O menu de navegação é acionado ao clicar no ícone de menu na barra superior esquerda do aplicativo. As opções de navegação apresentam para cada item de navegação um ícone seguido do texto informativo.

A Fig. 6 apresenta a tela inicial do aplicativo e o menu de navegação, ambos representados por um conjunto de *widjets* que compõem as telas.



Fig. 6. Tela inicial e tela de entrada.

A cor base do aplicativo é o branco e o azul e a utilização de cores claras tem o intuito de causar a sensação de leveza [29]. Os *widjets*, imagens, botões, texto e ícones utilizam a cor azul, oferecendo

um padrão visual comum em todas as telas do aplicativo.

As imagens utilizadas no aplicativo foram elaboradas especificamente para o aplicativo, criando assim um padrão visual particular com uma maior relação com o tema abordado. A Fig. 7 apresenta as telas de orientações sobre a deglutição e a disfagia orofaríngea com as imagens relacionadas a cada termo visando facilitar a compreensão por parte dos usuários. As imagens possuem um papel bastante importante na comunicação.



Fig. 7. Telas de orientação.

A tela do teste de rastreio da disfagia orofaríngea não possui imagem, o conteúdo informativo é textual, as perguntas são apresentadas em formato de formulário para possibilitar a entrada dos dados e a realização do rastreio. As perguntas são destacadas em azul com o texto branco, com as opções de respostas que variam de acordo com o tipo do dado. Os tipos podem ser texto, número, seleção ou múltipla escolhas. A Fig. 8 apresenta a tela de rastreio:



Fig. 8. Tela de rastreio.

B. Página Web

Como alternativa ao aplicativo, também, foi desenvolvida a página *web*, que além das opções disponíveis no aplicativo, permite a visualização gráfica analítica com base dados oriundos dos testes de rastreio.

Para o desenvolvimento da página, foram utilizadas as linguagens de programação PHP [30] em conjunto com *framework bootstrap* [31], que permite o desenvolvimento de sites e aplicações *web* utilizando HTML, CSS e *Javascript*.

PHP é uma linguagem de *script* de uso geral popular, especialmente adequada para o desenvolvimento da *web* [30], ela permite a geração de conteúdo dinâmico na *web*.

O *framework bootstrap* é um *framework* de código aberto para desenvolvimento de componentes de interface e *front-end* para sites e aplicações *web* usando HTML, CSS e *Javascript*, baseado em modelos de *design* para a tipografia, melhorando a experiência do usuário com um *site* responsivo.

Um *site* responsivo pode ser acessado de um computador, *notebook*, *smartphone*, *tablet* ou televisão. A Fig. 8 apresenta a tela inicial da página *web* com a opção de responsividade em computadores e dispositivos móveis.



Fig. 8. Visualização responsiva da página inicial da página web.

A página web também apresenta orientações sobre a deglutição e disfagia orofaríngea, como o aplicativo. Foram utilizadas as mesmas imagens para padronização, bem como para uma maior familiarização. A Fig. 9 apresenta a tela que contém as informações da deglutição.



Fig. 9. Visualização da tela com orientações sobre a deglutição.

O uso de interfaces mais adequadas e recursos gráficos podem auxiliar na demonstração das informações que são relevantes ao usuário. Estes recursos conseguem tornar mais eficiente o processo de acesso e recuperação da informação.

Vale ressaltar que as imagens utilizadas na página web adotam as Diretrizes de Acessibilidade para Conteúdo Web [32], desenvolvido pela W3C [33], organização internacional de padrões que desenvolve os pilares de tecnologias, neste caso para tornar o conteúdo acessível, principalmente, para os idosos.

O formulário para o rastreio da disfagia orofaríngea da página web é muito semelhante ao do aplicativo. Os dados são imputados a partir do formulário e assim é respondido o teste. A Fig. 10 apresenta a tela em formato de formulário.



Fig. 10. Página do Teste de Rastreamento.

Como complemento da solução apresentada na página, contempla-se a análise dos dados obtidos nas pesquisas, contendo as informações compiladas dos testes realizados. Os dados são apresentados em formato gráfico *pizza*, levando em consideração a faixa etária, o gênero e a idade. Desta forma é possível identificar os grupos que são mais vulneráveis de acordo com estas classificações. A Fig. 11 apresenta a página que apresenta os gráficos contendo os dados compilados.

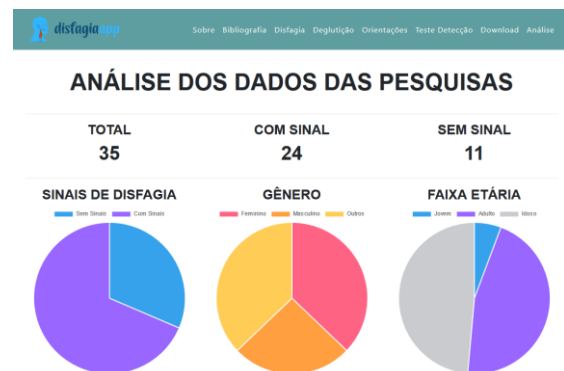


Fig. 11. Tela de visualização dos dados compilados.

C. API – Application Programming Interface

A API foi desenvolvida utilizando a linguagem de programação PHP com banco de dados MySQL, ela fornece serviços para o aplicativo e a página web.

Os dados são computados após o preenchimento e submissão dos formulários, seja através do aplicativo ou da página web. São computados e armazenados no banco de dados a idade, o gênero e as opções que permitem o rastreio da disfagia orofaríngea, nenhum outro dado do usuário é armazenado. A partir dos dados obtidos e armazenados na realização dos testes é possível a apresentação os dados compilados na página web.

VI. CONSIDERAÇÕES FINAIS

Nesta seção, apresentam-se as considerações sobre o desenvolvimento de uma solução *MHealth*, fruto de uma pesquisa interdisciplinar para rastreamento da disfagia orofaríngea, princípios de qualidade foram levados em consideração para propiciar ao usuário uma experiência, principalmente os idosos. A seguir serão apresentados os resultados e as contribuições.

A interdisciplinaridade foi um ponto positivo para a elaboração deste trabalho, no qual a professora Torres (s.d.) realizou o estudo sobre a disfagia e propôs em seu trabalho a criação de um aplicativo para a rastreamento da disfagia orofaríngea, permitindo, assim, que neste trabalho o foco fosse na elaboração do aplicativo em conjunto com a solução proposta da criação da página *web* e da API para disponibilidade dos dados obtidos durante o teste de rastreamento.

A utilização da metodologia *Kanban* foi extremamente importante para guiar as etapas e auxiliar o gerenciamento do projeto, pois a qualquer momento, era possível identificar o andamento e evolução do projeto.

Outro ponto importante foi a construção do aplicativo utilizando o *Flutter*, pois este permite a construção de aplicativos multiplataforma de maneira mais rápida contendo recursos de UI gráficas padronizados para oferecer uma interação mais amigável.

A correlação com a qualidade em relação aos sistemas computacionais, faz-se referência, automaticamente, à questão da usabilidade, pois os aspectos que indicam uma melhor usabilidade defendem questões como: a facilidade de aprendizado do sistema, facilidade de uso, satisfação dos usuários, inclusive, a adoção de questões de acessibilidade que foram aplicadas no aplicativo e na página *web* com o intuito de propiciar a utilização de maneira agradável por parte principalmente dos idosos.

Pode-se constatar, também, que através das avaliações heurísticas realizadas com as pessoas que fizeram o uso do aplicativo, foi um fator importante que resultou em algumas modificações até a versão atual do aplicativo aqui apresentada.

Para continuidade e ampliação da temática do trabalho são sugeridas as seguintes abordagens:

- Acompanhamento dos dados e identificação de possíveis novos itens de validação;

- Evolução da ferramenta incluindo inclusive possibilidade de envio dos testes por e-mail e/ou *whatsapp*.

VII. REFERÊNCIAS

- [1] MEDEIROS, Rodrigo Azevedo de et al., M-Health: definição, interesses, desafios e futuro, In: LEITE, Cícilia Raquel Maia; ROSA, Suelia Rodrigues Fleury (org.), “Novas tecnologias aplicadas à saúde: integração de áreas transformando a sociedade”, Mossoró, RN: EDUERN, 2017, p. 107-122. Disponível em: <<http://www.sbeb.org.br/site/wp-content/uploads/LivroVersaoFinal15-07-2017.pdf>>, acessado em 15/10/2021.
- [2] MEIRELES, Fernando de Souza, “Pesquisa do Uso de Tecnologia da Informação”, Fundação Getúlio Vargas de São Paulo (FGV-SP). Disponível em <<https://eaesp.fgv.br/producao-intelectual/pesquisa-anual-uso-ti>>, acessado em 16/10/2021.
- [3] ANDROID, “Principais critérios de qualidade do app”, atualizado 10/02/2021. Disponível em <<https://developer.android.com/docs/quality-guidelines/core-app-quality?hl=pt-br>>, acessado em 16/10/2021.
- [4] KORTUM, P.; SORBER, M. (2015), “Measuring the Usability of Mobile Applications for Phones and Tablets”, International Journal of Human-Computer Interaction, v. 31, n. 8, p. 518–529.
- [5] MAGALHÃES JUNIOR, HV; PERNAMBUCO, LA. Screening for oropharyngeal dysphagia. CoDAS, 2015; 27 (2). Disponível em <<https://www.scielo.br/j/codas/a/F6LRfmQRzBxB6kV89yHyf3M/?lang=en>>, acessado em 06/02/2022.
- [6] TORRES, Ana Catarina Moura, “Aplicativo para Detecção de Risco de Disfagia: Desenvolvimento e Aplicação”, s.d., Universidade Federal da Bahia, Instituto de Ciência da Saúde. Salvador-BA.
- [7] NIELSEN’S, Jakob, “10 Usability Heuristics for User Interface Design”, 24/04/1994, atualizado em 15/11/2020. Disponível em <<https://www.nngroup.com/articles/ten-usability-heuristics/>>, acessado em 16/10/2021.
- [8] SOMMERVILLE, Ian, “Software Engineering”, Boston: Pearson Education, v.9 edition, 011. ISBN-13:978-0-13-703515-1.
- [9] KANBAM, “O que é Kanban? Definição e Detalhes Explicados”, disponível em <<https://kanbanize.com/pt/recursos-kanban/primeiros-passos/o-que-e-kanban>>, acessado em 10/01/2022.
- [10] TRELLO. “O Trello ajuda os times a agilizar o trabalho”, disponível em <<https://trello.com/pt-BRhttps://trello.com/pt-BR>>, acessado em 20/10/2021.

- [11] MALDONATO, José Manuel Santos de Varge; MARQUES, Alexandre Barbosa; CRUZ, Antonio, “Telemedicina: desafios à sua difusão no Brasil”, Cad. Saúde Pública, Rio de Janeiro, 32 Sup 2:e00155615, 2016.
- [12] ZAMORA, Antonio Julián Alférez, “Estudio comparativo de frameworks multiplataforma para desarrollo de aplicaciones móviles”, E.T.S.I. Telecomunicación (UPM), 2019.
- [13] STARK, J., “Building iPhone Apps with HTML, CSS, and JavaScript”, O’Reilly Media, Janeiro, 2010.
- [14] APACHE CORDOVA, “The Apache Software Foundation”, disponível em <<https://cordova.apache.org/>>, acessado em 12/11/2021.
- [15] DRIFTY CO., “Cross-Platform Mobile App Development”, 2020. Disponível em <<https://ionicframework.com/>>, acessado em 12/11/2021.
- [16] PHONEGAP, “Adobe Systems Inc.”, disponível em <https://phonegap.com/>, acessado em 12/11/2021.
- [17] GONSALVES, M, “Evaluating the Mobile Development Frameworks Apache Cordova and Flutter and Their Impact on the Development Process and Application.
- [18] SCHARFSTEIN, M. e GAURF, “OTT Video-Oriented Mobile Applications Development Using Cross-Platform UI Frameworks”, J. Chem. Inf. Model., vol. 53, n. 9, pp. 1689–1699, 2013, doi: 10.1017/CBO9781107415324.004.
- [19] APPLE DEVELOPER, “Framework, JavaScriptCore”, disponível em <<https://developer.apple.com/documentation/javascriptcore>>, acessado em 12/11/2021.
- [20] REACT NATIVE, “JavaScriptCore”. Disponível em <<https://reactnative.dev/>>, acessado em 11/11/2021.
- [21] EISENMAN, Bonnie, “Learning React Native: Building Native Mobile Apps with JavaScript”, O’Reilly Media, 2016.
- [22] FLUTTER, “Build apps for any screen”, disponível em <<https://flutter.dev/>>, acessado em 17/11/2021.
- [23] FLUTTER, “Introduction to widgets”, disponível em <<https://flutter.dev/docs/development/ui/widgets-intro>>, acessado em 15/11/2021.
- [24] SIDI, “Guia para o Desenvolvimento de Aplicações Móveis Acessíveis”, disponível em <<https://www.sidi.org.br/guiadeacessibilidade/>>, acessado em 20/11/2021.
- [25] DART, “Paint your UI to life”, disponível em <<https://dart.dev/>>, acessado em 13/11/2021.
- [26] WATERS, D. J., “Projeto conservação preventiva em bibliotecas e arquivos. Do microfilme à imagem digital”, Rio de Janeiro: Arquivo Nacional, 1997.
- [27] SIDI, “Samsung Instituto de Desenvolvimento para Informática”. Disponível em <<https://www.sidi.org.br/>>, acessado em 20/11/2021.
- [28] MATERIAL DESIGN, “Create intuitive and beautiful products with Material Design”, disponível em <<https://material.io/design>>, acessado em 20/11/2021.
- [29] GUIMARÃES, Luciano, “A Cor como Informação”, 2.ed. São Paulo: Annablume, 2000, 160p.
- [30] PHP, “Hypertext Preprocessor”, disponível em <<https://www.php.net/>>, acessado em 17/11/2021.
- [31] BOOTSTRAP, “The most popular HTML, CSS, and JS library in the world”, disponível em <<https://getbootstrap.com/>>, acessado em 17/11/2021.
- [32] W3C, “Diretrizes de Acessibilidade para Conteúdo Web (WCAG) 2.1”, disponível em <<https://www.w3c.br/traducoes/wcag/wcag21-pt-BR/>>, acessado em 15/11/2021.
- [33] W3C BRASIL, “World Wide Web Consortium Escritório Brasil”, disponível em <<https://www.w3c.br/>>, acessado em 17/11/2021.