

Voice Training: Desenvolvimento de aplicativo para treinamento de avaliação perceptiva-auditiva da voz.

Voice Training: Developing a mobile app for training auditory- perceptual assessment of the voice.

RESUMO

Adilson Franke Neia Júnior
adilsonfrankeia@gmail.com
Universidade Tecnológica Federal do Paraná, Cornélio Procópio, Paraná, Brasil

Maria Eugenia Dajer
medajer@utfpr.edu.br
Universidade Tecnológica Federal do Paraná, Cornélio Procópio, Paraná, Brasil

Nathan Antonio Guerreiro
nathanquerreiro@hotmail.com
Universidade Tecnológica Federal do Paraná, Cornélio Procópio, Paraná, Brasil

A análise perceptiva-auditiva é considerada uma das melhores maneiras para detectar características da voz, mas depende da experiência do avaliador. O objetivo desta pesquisa é desenvolver um aplicativo *mobile* que auxilie no treinamento de estudantes e profissionais fonoaudiólogos. O aplicativo possui dois tipos de treinamento, a classificação da predominância das características de tensão, sopro e rugosidade e a discriminação de 4 níveis de intensidade do desvio vocal. Foi utilizado um banco de vozes com 30 sinais previamente classificadas por um fonoaudiólogo especialista em voz. O aplicativo *voice training* foi desenvolvido para a utilização nos *smartphones*, sendo utilizado *Node.js*, *Mongodb* e *React Native*.

PALAVRAS-CHAVE: Plataforma de aprendizagem. Fonoaudiologia. Avaliação Perceptivo-auditiva. Voz

ABSTRACT

The auditory-perceptual analysis is considered one of the best ways to detect voice characteristics, but it depends on the evaluator's experience. The objective of this research is to develop a mobile application that assists the training of students and speech-language pathologists. The application has two types of training, the classification of the predominance of the characteristics of tension, breathiness, and roughness, and the discrimination of 4 levels of intensity of the vocal deviation. It used a database of 30 voice signals previously classified by a speech therapist. The voice training application was developed for use on smartphones, using Node.js, Mongodb and React Native

KEYWORDS: Learning Platform. Speech Therapists. Auditory-perceptual evaluation. Voice.

Recebido: 19 ago. 2020.

Aprovado: 01 out. 2020.

Direito autoral: Este trabalho está licenciado sob os termos da Licença Creative Commons-Atribuição 4.0 Internacional.



INTRODUÇÃO

Com o avanço da tecnologia e da acessibilidade, diversas pessoas estão tendo acesso à conteúdo e experiências que não seriam possíveis em outros tempos. Desde 2007 os dispositivos móveis inteligentes, *smartphones*, têm dominado o mercado ano após ano. Com esse avanço o mercado de aplicativos cresceu de forma desenfreada, e empresas como Google e Apple possuem inúmeros destes em suas lojas. Com isso os aplicativos voltados à saúde também entraram nesta conta, com mais de 31 mil aplicativos disponíveis para *download* (PAYNE,2015). Já em 2020, este número é imensamente maior, contando com cerca 315 mil nas lojas de aplicativo espalhadas pelo mundo (MOBILUSMD,2020).

O advento das tecnologias de informação e comunicação (TICs) trouxe uma nova relação a forma e ao alcance da educação a distância (EaD). No Brasil mais de 2,5 milhões de brasileiros realizavam um ensino a distância em 2008, sendo 79% uma graduação ou uma pós-graduação à distância. Com isso diversas plataformas de ensino estão surgindo, principalmente com a utilização de dispositivos móveis, potencializando o *mobile learning (m-learning)* e o *e-learning*. Segundo Yau e Joy (2010) a utilização destes dispositivos permite que os estudantes e professores extraiam os recursos oferecidos por estas plataformas, tornando o ensino mais agradável e acessível (FERNANDES et al., 2012).

Tendo estas informações em mente, este trabalho se propõe a criação de um aplicativo móvel para auxiliar o treinamento de profissionais e estudantes da área de fonoaudiologia a identificar características de rugosidade, soproidade e tensão de uma voz e os níveis de intensidade da característica. Esta ferramenta utilizará uma base de dados contendo 30 amostras de vozes classificadas por um fonoaudiólogo especialista em voz.

MATERIAIS E MÉTODOS

O aplicativo tem como objetivo facilitar o treinamento do usuário na avaliação perceptiva-auditiva da voz. O treinamento do usuário será dividido em duas formas, sendo uma oferecendo apenas a escolha da característica predominante, e outra sendo possível identificar 4 níveis de intensidade de cada característica. Em cada sessão de treinamento serão oferecidas 5 amostras aleatórias.

Para o desenvolvimento do aplicativo foram utilizadas as seguintes ferramentas:

VSCode: Visual Studio Code é uma ferramenta leve e poderosa capaz de editar códigos, sendo seu sistema operacional Windows, MacOs ou Linux (VSCODE,2020).

Trello: Ferramenta utilizada para a separação das tarefas e melhor visualização do progresso do projeto (TRELLO,2020).

Figma: Ferramenta utilizada para a criação dos *layouts* e interfaces (Figma,2020).

MongoDb: é um banco de dados não relacional, ou seja, não possui relação entre tabelas. É um banco de documentos, o que significa que guarda os dados e realiza *queries* (buscas) no formato JavaScript Object Notation (JSON) (MONGODB,2020).

MongoDb Atlas: é um serviço em nuvem que facilita o uso e armazenamento de dados. Além de possuir a facilidade de implementar e gerenciar serviços em computação em nuvem (MongoDBAtlas,2020).

A seguir são apresentadas as linguagens de programação e *frameworks* (conceito que une códigos comuns entre várias aplicações) utilizados no projeto:

Javascript: é uma linguagem de programação interpretada e baseada em objetos de primeira classe, mais conhecida como a linguagem de *script* para páginas web, mas também utilizada em ambientes sem *browser*. É uma linguagem baseada em protótipos, multi-paradigma e dinâmica, suportando orientação a objetos, imperativos e declarativos. O padrão do JavaScript é o *ECMAScript* (Mozilla, 2020).

NodeJS: é um ambiente de execução utilizado no *server-sider* (lado do server) incluindo tudo que é necessário para a execução de um programa escrito em JavaScript. Para que isso possa ser realizado o *node* utiliza o "motor" V8 do Google Chrome, pegando seu código e convertendo este para um código de máquina. Com isso, esse apresenta um ótimo desempenho (NodeJs,2020).

React Native: é um *framework* do JavaScript para criar aplicativos mobiles nativos para IOS e Android. É baseado no React, um *framework* do Facebook para a criação de interfaces (EINSENMAN, 2015).

Expo: é um *framework* e uma plataforma para aplicações *React*. É um conjunto de ferramentas e serviços construídos em volta do *React Native* e plataformas nativas que ajudam no desenvolvimento, construção e lançamento (Expo,2020).

Processo de Criação:

No início do projeto foi feita uma análise visando quais seriam os propósitos e funções do sistema, sendo então realizada um levantamento de requisitos. Segundo a norma 90 do IEEE (IEEE, 1990) requisito de software pode ser definido como:

1. Uma capacidade que um usuário necessita para resolver um problema ou atingir um objetivo.
2. Uma capacidade que deve ser atendida ou possuída por um sistema ou componente de um sistema para satisfazer um contrato, padrão, especificação ou outro documento formalmente imposto.
3. O conjunto de todos os requisitos que formam a base para o desenvolvimento subsequente de um software ou componentes de um software.

Os requisitos de um sistema podem ser divididos em duas categorias, funcionais e não-funcionais. Os funcionais se referem às funções que o sistema deve possuir para atender a demanda. Os requisitos não-funcionais atribuem ao ambiente, sendo possivelmente: características de segurança, desempenho, modelagem dos dados utilizados (LIMA,2012).

Analisando os requisitos foi possível realizar o Quadro 1, contendo os requisitos funcionais do sistema e suas respectivas prioridades, sendo cada uma dividida por um código.

Quadro 1: Requisitos funcionais do sistema

Código	Nome	Prioridade
RF01	Criar usuário	Alta
RF02	Serviço de <i>login</i>	Baixa
RF03	Serviço de <i>logout</i>	Baixa
RF04	Treinamento por característica Predominante	Alta
RF05	Treinamento da intensidade da característica vocal	Alta

Fonte: Autoria própria (2020)

Os requisitos não-funcionais podem ser vistos na Quadro 2.

Quadro 2: Requisitos não-funcionais do sistema

Código	Nome	Prioridade
RNF01	Criar <i>layout</i> do aplicativo	Alta
RNF02	Inserção dos dados no banco de Dados	Alta
RNF03	Mudar as vozes para o formato mp3	Baixa

Fonte: Autoria própria (2020)

Com os requisitos separados, foi escolhida a metodologia de trabalho o Scrum. O Scrum é uma metodologia ágil para gestão e planejamento de software. Nele o projeto é dividido em ciclos chamados *sprints*. O *sprint* representa um conjunto de atividades que devem ser desenvolvidos em um tempo estipulado (SOMMERVILLE, 2019). Com isso foram feitas 3 *sprints* (Quadro 3) para a realização do projeto.

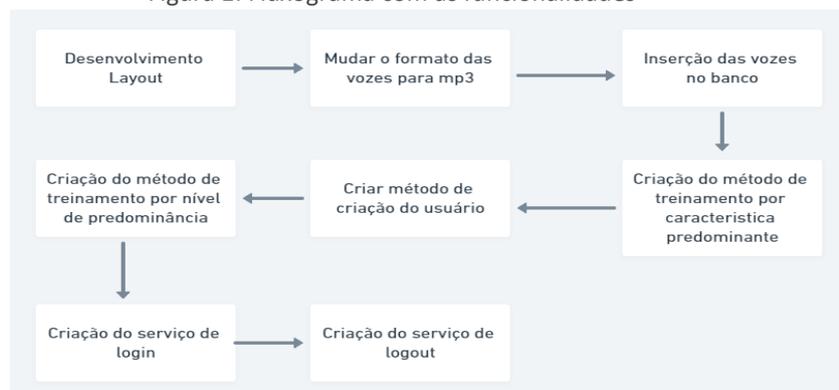
Quadro 3: *Sprints* realizadas

Número da sprint	Requisitos executados	Tempo estipulado
1	RNF01, RNF02, RNF03	02 semanas
2	RF04, RF01, RF05	04 semanas
3	RF03, RF02	02 semanas

Fonte: Autoria própria (2020)

O fluxograma (Figura 1) mostra ainda a ordem do desenvolvimento das funcionalidades.

Figura 1: Fluxograma com as funcionalidades



Fonte: Autoria própria (2020)

RESULTADOS

Aplicando as metodologias ágeis de desenvolvimento de software e as tecnologias demonstradas na seção de materiais e métodos, foi possível realizar a criação de um aplicativo nos moldes MVP, mínimo produto viável, para que assim possa ser apresentado um protótipo.

A base de dados utilizada neste trabalho contia 30 vozes, separadas por sexo e predominância da característica vocal de soproidade, tensão e rugosidade e intensidade do desvio vocal.

Na Figura 2 é possível observar o MongoDB Atlas, ferramenta onde está armazenado os dados referentes as vozes e aos usuários. As coleções foram divididas em dois esquemas, um referente aos dados dos usuários e outro contendo todas as informações das vozes, como o nível da intensidade-sonora e o sexo do paciente. Quando a requisição é feita através do servidor, uma consulta (*query*) é realizada, buscando assim os dados desejados.

Figura 2: Gerenciador de banco de dados AtlasDB

Collection Name	Documents	Documents Size	Documents Avg	Indexes	Index Size	Index Avg
userschemas	2	254B	127B	1	36KB	36KB
voiceschemas	31	4.43KB	147B	1	36KB	36KB

Fonte: Autoria própria (2020)

As consultas realizadas nos treinos foram feitas utilizando JSON, e não *SQL* (*Structured Query Language*), que é comum em bancos relacionais. Esta query vai então nos retornar 5 vozes, que são as necessárias para a realização de um treinamento por característica predominante ou intensidade da característica vocal. A *query* utilizada para este retorno é vista na Figura 3:

Figura 3: Query utilizada para a consulta das vozes

```
db.voiceschemas.find(
  {$or:[ { id:var1},{id:var2}, {id:var3}, {id:var4}, {id:var5}]}
)
```

Fonte: Autoria própria (2020)

O treinamento por característica predominante terá a interface mostrada na Figura 4. Quando o usuário entrar nesta interface será feita uma requisição *http*, *hypertext transfer protocol*, com um método solicitando uma voz aleatória. Caso ele queira ouvir novamente a voz, poderá clicar no símbolo de *play*. Após escutar e analisar, o fonoaudiólogo ou estudante deve responder com uma das opções (Rugosidade, soproidade ou tensão). Caso o usuário erre, este ainda terá mais uma chance de selecionar outra característica. Após o término da primeira voz, outra voz é reproduzida. O treinamento acaba quando 5 vozes tiverem sido escutadas e analisadas pelo usuário.

Figura 4: Interface treinamento por disfunção predominante



Fonte: Autoria Própria (2020)

Funcionalidades O treinamento por nível de intensidade da característica vocal terá a interface mostrada na Figura 5.

Figura 5: Interface para o treinamento por nível da característica vocal



Fonte: Autoria Própria (2020)

Esta interface também fará uma requisição *http* com um método, mas nessa etapa o usuário deverá selecionar o nível da voz de acordo com a sua característica e clicar no botão “próxima”, para continuar escutando as próximas vozes do treinamento. Caso o usuário não acerte, poderá fazer a tentativa mais uma vez.

Após a primeira voz, uma nova voz é reproduzida, somando um total de 5 vozes no final. Ao fim do treino, ele poderá visualizar as respostas.

CONCLUSÃO

Sabendo que o uso de aplicativos no âmbito da saúde vêm crescendo ano após ano, este trabalho teve como objetivo desenvolver um aplicativo mobile simples e objetivo para o treinamento de fonoaudiólogos interessados na área de voz. Sendo assim, foi possível criar tal aplicativo, mostrando que a junção de tecnologia e medicina podem sim ajudar muitas pessoas de diversas áreas.

Destaca-se ainda que para possíveis trabalhos futuros, planeja-se realizar testes com profissionais de fonoaudiologia que avaliarão as funcionalidades e a “usabilidade” do aplicativo e a obtenção de um banco de dados contendo mais vozes.

AGRADECIMENTOS

Agradeço muito à Universidade Tecnológica Federal do Paraná Campus Cornélio Procópio por proporcionar tal oportunidade para a realização desta iniciação científica.

Agradeço a grande paciência, ajuda e incentivo da minha orientadora que sempre se mostrou interessada e disposta a ajudar com todos os problemas existentes.

Agradecemos também a Dra. Glaucya Madazio do Centro de Estudos da Voz (CEV- SP) por compartilhar o banco de vozes classificado por predominância e intensidade do desvio vocal.

Agradeço a minha família, amigos e colegas de trabalho por todo o apoio que me foi proporcionado.

REFERÊNCIAS

EINSEMAN, Bonnie. **Learning React Native**. 1ª edição. O’Reilly, 22 de dezembro de 2015.

Expo. **Introduction to Expo**. Disponível em: <https://docs.expo.io/>. Acesso em 23 de setembro de 2020.

FERNANDES, Kleber Tavares et al. E-learning via dispositivos móveis no Brasil: Estado da Arte e Desafios a Luz do Acesso Participativo e Universal do Cidadão Brasileiro ao Conhecimento. In: ANAIS do Workshop de Desafios da Computação Aplicada a Educação. [S.l.: s.n.], 2012. p. 128–137.

Figma. **Design Features**. Disponível em: <https://www.figma.com/design/>. Acesso em 23 de setembro de 2020

IEEE Standard Glossary of Software Engineering Terminology. **IEEE Std 610.12-**

1990, p. 1–84, 1990

LIMA, Adilson da silva. **Especificações técnicas de software**. 1ª edição. Editora Érica, 22 de março de 2012

MOBILUS MD. **11 surprising health statics**. Disponível em: <https://www.mobius.md/blog/2019/03/11-mobile-health-statistics/>
Acesso em: 23 de setembro de 2020

MDN WEB DOCS. **Sobre JavaScript**. Disponível em: https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/About_JavaScript. Acesso em: 6 de Agosto de 2020.

Mongo DB. **Cloud Product**. Disponível em: <https://docs.mongodb.com/cloud/>. Acesso em 23 de setembro de 2020.

Mongo DB. **The MongoDB 4.4 Manual**. Disponível em: <https://docs.mongodb.com/manual/>. Acesso em 23 de setembro de 2020.

NodeJS. **API reference documentation**. Disponível em: <https://nodejs.org/en/docs/>. Acesso em 23 de setembro de 2020

PAYNE, Hannah E et al. Behavioral functionality of mobile apps in health interventions: a systematic review of the literature. **JMIR mHealth and uHealth**, JMIR Publications Inc., Toronto, Canada, v. 3, n. 1, e20, 2015.

SOMMERVILLE, Ian. **Engenharia de software**. 10ª edição. Pearson Universidades, 22 de abril de 2019.

Trello. **Página Inicial**. Disponível em: <https://trello.com/pt-BR>. Acesso em 23 de setembro de 2020.

VISUAL STUDIO CODE. **Getting started**. Disponível em: <https://code.visualstudio.com/docs>. Acesso em: 23 de setembro de 2020