UNIVERSIDADE FEDERAL DE SANTA MARIA
CENTRO DE TECNOLOGIA
PROGRAMA DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

Frederico Hansel dos Santos Gassen

SISTEMA DE IDENTIFICAÇÃO DIGITAL E REFATORAÇÃO APLICADAS EM UM APLICATIVO PARA AVALIAÇÕES FONOLÓGICAS

Santa Maria, RS
2021

Frederico Hansel dos Santos Gassen

SISTEMA DE IDENTIFICAÇÃO DIGITAL E REFATORAÇÃO APLICADAS EM UM APLICATIVO PARA AVALIAÇÕES FONOLÓGICAS

Trabalho de Conclusão de Curso apresentado ao Bacharelado em Ciência da Computação da Universidade Federal de Santa Maria (UFSM, RS), como requisito parcial para a obtenção do grau de **Bacharelado em Ciência da Computação**

Orientador: Prof. Dr. João Carlos Damasceno Lima

Co-orientador: Prof. Dr. Celio Trois

478

Santa Maria, RS

2021

Hansel dos Santos Gassen, Frederico

Sistema de Identificação Digital e Refatoração aplicadas em um Aplicativo para Avaliações Fonológicas / por Frederico Hansel dos Santos Gassen. – 2021.
55 f.: il.; 30 cm.

Orientador: João Carlos Damasceno Lima
Co-orientador: Celio Trois
Trabalho de Conclusão de Curso - Universidade Federal de Santa Maria, Centro de Tecnologia, Bacharelado em Ciência da Computação, RS, 2021.

1. Identificação digital. 2. Voz. 3. Fonoaudiologia. I. Damasceno Lima, João Carlos. II. Trois, Celio. III. Título.

E-mail: fhgassen@inf.ufsm.br

Frederico Hansel dos Santos Gassen

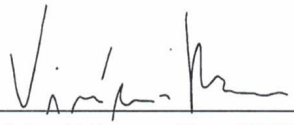
SISTEMA DE IDENTIFICAÇÃO DIGITAL E REFATORAÇÃO APLICADAS EM UM APLICATIVO PARA AVALIAÇÕES FONOLÓGICAS

Trabalho de Conclusão de Curso apresentado ao Bacharelado em Ciência da Computação da Universidade Federal de Santa Maria (UFSM, RS), como requisito parcial para a obtenção do grau de **Bacharelado em Ciência da Computação**

Aprovado em 12 de fevereiro de 2021:

João Carlos Damasceno Lima, Dr. (UFSM)
(Presidente/Orientador)

Maria Helena Franciscatto, M.ª (UFSM)



Vinícius Maran, Dr. (UFSM)

Santa Maria, RS

2021

DEDICATÓRIA

Dedico este trabalho a meus pais que esperaram longos anos por esse momento para que finalmente pudessem viver a vida que sempre sonharam: a calmaria de uma praia isolada.

AGRADECIMENTOS

Agradeço a meus pais, Gilmar e Milene, por todo apoio e condição dada em minha vida acadêmica. Foram anos de estudos para que este momento pudesse ser concretizado.

Agradeço a meu irmão Guilherme pela parceria indescritível neste longo período. Com a tua ajuda este período se tornou mais alegre.

Gratifico aos funcionários da UFSM, desde o cozinheiro do Restaurante Universitário pelos 4 anos de refeições até os professores que compartilharam seu tempo e conhecimento para a construção de ótimos profissionais para a nação.

Por fim, mas não menos importante, as amizades de coração consolidadas nestes 4 anos de graduação que certamente perdurarão o resto de nossas vidas. Se não fosse vocês, o período em Santa Maria não seria nada colorido e alegre.

"Não tenha pena dos mortos. Tenha pena dos vivos, e acima de tudo aqueles que vivem sem amor."

(ALVO DUMBLEDORE)

RESUMO

SISTEMA DE IDENTIFICAÇÃO DIGITAL E REFATORAÇÃO APLICADAS EM UM APLICATIVO PARA AVALIAÇÕES FONOLÓGICAS

AUTOR: FREDERICO HANSEL DOS SANTOS GASSEN
ORIENTADOR: JOÃO CARLOS DAMASCENO LIMA
CO-ORIENTADOR: CELIO TROIS

Os distúrbios de fala são problemas caracterizados por processos fonológicos que afetam crianças até sua fase de pré-adolescência. Nestes quadros, os profissionais da área fonoaudiológica atuam realizando avaliações e, se necessário, futuras terapias visando corrigi-los. Este contexto motivou a criação de um aplicativo para triagem fonológica infantil, o eFono, onde percebeu-se alguns problemas remanescentes como a impossibilidade de identificação e agrupamento das avaliações realizadas por um mesmo paciente, a defasagem tecnológica do aplicativo e a restrição de seu uso à um grupo específico de usuários. Para a solução destes problemas, este trabalho refatorou o código-fonte do aplicativo e pesquisou um sistema de identificação digital por voz para integração com o mesmo, escolha motivada pela retrocompatibilidade com as vozes já coletadas no trabalho anterior e também pela combinação de baixo custo computacional, facilidade de uso e alta assertividade em comparação com os demais sistemas de identificação biométricos já existentes. Na metodologia, este trabalho compara resultados obtidos de duas bibliotecas, Recognito e Alizé, responsáveis por realizar a identificação do locutor utilizando um *dataset* com 1045 avaliações fonológicas. A partir dos resultados dos testes, concluiu-se que a biblioteca Recognito não se mostrou eficiente para integração com o aplicativo, enquanto que a Alizé obteve uma taxa de acerto de 91,20%. Para trabalhos futuros, recomenda-se a exportação do Alizé para alguma linguagem de programação que possa ser mantida em um servidor para realizar a efetiva integração com o aplicativo supracitado.

Palavras-chave: Identificação digital, voz, fonoaudiologia.

ABSTRACT

DIGITAL IDENTIFICATION SYSTEM AND REFACTORING APPLIED TO A PHONOLOGICAL EVALUATION APP

AUTHOR: FREDERICO HANSEL DOS SANTOS GASSEN
ADVISOR: JOÃO CARLOS DAMASCENO LIMA
COADVISOR: CELIO TROIS

Speech disorders are problems characterized by phonological processes that affect kids until their pre-adolescence. In this cases, the speech terapists perform evaluations and, if necessary, future therapies aiming to correct them. This scenario motivated the development of a mobile app, called eFono, to perform childish phonological screening; the app, however, still has some remaining problems like the impossibility of identification and grouping of evaluations performed by the same patient, the depreciated techonologies of the app and the restricted use of a selected group of users. To solve these problems, this work refactored the app source code and investigated a voice digital identification system to integrate with the app, choice motivated by the backwards compatibility with the voices already collected in the previous work and a combination of factors such as low computational cost, ease of use and high assertivity when compared to other known biometric identification systems. In the methodology, this work compares the results obtained from two libraries, Recognito and Alizé, which both perform the speaker identification, using a dataset that contains 1045 phonological evaluations. With the tests results, it was concluded that Recognito tool was not efficient enough to integrate with the app, whereas Alizé got an assertivity rate of 91.20%. To future projects, it is recommended the exportation of Alizé to a programming language that can be maintained in a server to realize the effective integration with the app.

Keywords: Digital identification, voice, speech therapy.

LISTA DE FIGURAS

LISTA DE ABREVIATURAS E SIGLAS

AFI	Alfabeto Fonético Internacional
API	Application Programming Interface
CPU	Central Processing Unit
GMM	Gaussian Mixture Modeling
HTTP	Hypertext Transfer Protocol
PET	Phonological Evaluation Tools
SDK	Software Development Kit
SO	Sistema Operacional
SVM	Support Vector Machine
UX	User Experience

SUMÁRIO

1 INTRODUÇÃO

Historicamente, viver em sociedade sempre esteve acompanhado da comunicação, seja ela através dos grunhidos nos primórdios de nossa espécie ou até mesmo da internet que conhecemos hoje. Segundo Perles (2007), a comunicação sonora existia antes mesmo da invenção da escrita, em meados do século IV a.C., estopim para o início do que se entende por história. Em outras palavras, a comunicação sonora já era utilizada antes mesmo de um dos maiores marcos históricos da comunicação interpessoal, já mostrando sua importância em um período muito diferente ao que vivemos hoje.

A partir desta evolutiva, nota-se a importância da área fonoaudiológica, iniciada no Brasil na década de 1960 a partir da necessidade de reabilitar indivíduos portadores de distúrbios da comunicação, como afirma (BERBERIAN, 1995). Ela é responsável por estudar os fonemas, menor unidade sonora de uma língua, e suas funções na mesma. São sons que, articulados e combinados, geram as sílabas, que em seu conjunto geram as palavras, assim por diante até a concretização da comunicação.

Os distúrbios de fala, motivo da criação da área fonológica, são caracterizados por uma desorganização linguística do inventário de fonemas, sendo identificado por omissões e substituições destes na fala, especialmente consoantes e encontros consonantais (GRUNWELL, 1981; FERRANTE; BORSEL; PEREIRA, 2009). Em outras palavras, uma pessoa com desvio fonológico acaba por omitir ou substituir incorretamente fonemas em seu processo de fala. Esses conceitos são aprofundados na Seção 2.1, visando a melhor integração do leitor com o tema.

O fonoaudiólogo, dentre outras atividades, é responsável por identificar esses distúrbios através da realização de avaliações fonológicas. Nessas avaliações, segundo (CERON, 2015), normalmente apresenta-se uma sequência de imagens de objetos ou ações com seus respectivos nomes para que o paciente elicite suas pronúncias. Posteriormente, o profissional transcreve a fala para o alfabeto fonético e compara com os fonemas pré mapeados considerados corretos, levando em consideração a idade do paciente e região onde vive, dado que existem diferentes dialetos e sotaques específicos de uma localidade. Por exemplo, a transcrição da palavra “carro” em sua pronúncia comum para o alfabeto fonético resulta em “[’karu]”.

Visando a ampliação do acesso à essas avaliações, no trabalho de (MORO, 2018) foi desenvolvido um aplicativo Android batizado de *eFono* para realizar a triagem[1] fonológica infantil,

[1] Processo no qual se define a prioridade do tratamento com base na gravidade do seu estado.

o qual é aprofundado na Seção 3.2. Nele, a organização de execução é semelhante à organização da avaliação presencial, diferenciando-se, porém, nas adaptações tecnológicas. Primeiramente, apresenta-se ao paciente uma sequência de objetos e ações. Logo após, o aplicativo realiza a gravação da voz do paciente elicitando as palavras e envia as gravações para um servidor. Por fim, o sistema aplica o serviço de classificação desenvolvido por (ALMEIDA, 2018) nos fonemas identificados, retornando ao aplicativo o resultado da triagem inicial do paciente acerca de seu nível de distúrbio fonético.

O aplicativo cumpre corretamente a sua proposta de auxiliar no diagnóstico de avaliações fonológicas. Porém, possui algumas deficiências em sua concepção: (i) não utiliza perfil com identificador único para os avaliados, (ii) está defasado tecnologicamente e (iii) apresenta restrição de um único avaliador em múltiplas avaliações de um mesmo paciente. Os parágrafos a seguir apresentam melhor estes problemas e possíveis soluções relacionadas.

No que se refere ao quesito (i), no *eFono*, as avaliações são realizadas individualmente, dissociadas do conceito de um perfil pessoal. A maior possibilidade de estabelecer uma relação entre duas avaliações diferentes de uma mesma pessoa é através de seu nome e data de nascimento, informações preenchidas pelo avaliador passíveis de erros de digitação. Casos onde o paciente realiza a primeira avaliação, descobre algum tipo de desvio de fala, passa por uma terapia e realiza uma segunda avaliação para verificar se o problema foi corrigido poderiam estar contidos no perfil do paciente dentro do sistema do *eFono*. Assim, seria possível realizar uma comparação automática dos resultados de múltiplas avaliações.

Com relação à defasagem tecnológica do eFono (quesito ii), esta relaciona-se ao SO Android no qual o app foi implementado, visto que nos anos que se passaram desde a sua implementação o SO passou por mudanças em suas diretrizes de desenvolvimento, explicadas na Seção 3.2.1. Estas mudanças possuem vantagens como reduzir a taxa de falhas e diminuir a quantidade de código necessária para implementação de funcionalidades, justificando a refatoração do aplicativo em seus pontos.

Outro ponto levantado (iii) diz respeito à restrição de uso do aplicativo por um conjunto restrito de usuários. Sem a estrutura de perfis supracitada, a reavaliação de um paciente por um profissional diferente do que o avaliou a primeira vez é um desafio, dado que ele não terá as informações completas da primeira avaliação realizada pelo paciente no mesmo aplicativo. Em uma analogia ao sistema de vacinação brasileiro, seria um problema semelhante à inexistência de um sistema para controle do histórico de vacinação: o agente de saúde, em uma posterior

aplicação da vacina, não saberia se o cidadão já a recebeu e quais ainda falta receber.

Considerando os problemas acima mencionados, este trabalho propõe-se a realizar uma refatoração do aplicativo eFono, adequando-o às recomendações atuais do desenvolvimento Android e, especialmente, desenvolvendo e integrando um sistema de identificação digital para criação de perfis e agrupamento das avaliações realizadas por um mesmo paciente. Assim, o sistema poderia ser utilizado em larga escala ou integrado à alguma rede de saúde pública para auxílio fonoaudiológico.

Para a criação de perfis pessoais no aplicativo, existem diversas características no corpo humano que podem ser usadas para identificação digital. A Seção 2.2 apresenta algumas delas e as justificativas levantadas para a escolha da voz como fator biométrico no sistema. Como base deste sistema, foram investigadas uma série de bibliotecas para identificação digital por voz. Estas bibliotecas, apresentadas na Seção 2.3, abstraem muitas funcionalidades que podem ser utilizadas neste trabalho. Porém, algumas delas não são válidas para utilização neste contexto de avaliações fonológicas.

Para a escolha da solução mais adequada à este caso de uso, foi estruturado um ambiente de testes descrito na Seção 3.3.2 que visa comparar as soluções possíveis de serem utilizadas no âmbito deste trabalho. Este ambiente busca reproduzir o fluxo de execução de um sistema de avaliações fonológicas, incrementando a quantidade de locutores cadastrados ao passo de que as avaliações são realizadas, mensurando a taxa de assertividade da solução com quantidades elevadas de vozes armazenadas.

A realização dos testes fez uso de 1045 avaliações fonológicas infantis coletadas no INFONO, trabalho de (CERON, 2015). Estas vozes foram extraídas de um banco de dados e anexadas aos projetos de testes. Para a obtenção de resultados mais próximos da realidade do aplicativo, vozes coletadas no mesmo contexto são importantes para reduzir possíveis discrepâncias nos resultados entre os testes e a realidade das avaliações fonológicas. Além disso, uma amostra grande de vozes aumenta a confiança dos resultados alcançados, mostrando melhor o comportamento das bibliotecas ao longo das avaliações.

Como resultado destes testes, detalhados na Seção 3.3.3, concluiu-se que uma das bibliotecas não obteve respostas satisfatórias em suas possíveis configurações, enquanto que a outra mostrou-se eficaz no caso de uso do *eFono*. Porém, a disponibilização no formato de uma biblioteca Android presente no repositório invalidou sua integração com o aplicativo já no âmbito deste trabalho, visto que é recomendado a centralização em um servidor deste sistema. A

conversão das estruturas base dessa biblioteca para uma linguagem de programação disponível para uso no servidor do projeto tornou-se um dos trabalhos futuros deste projeto, os quais são mencionados na Seção 4.1.

Através das informações apresentadas, pode-se perceber os pontos passíveis de evolução no aplicativo *eFono* e suas respectivas justificativas introduzidas. Em seguida, o texto continua com o Capítulo 2, onde serão apresentados alguns conceitos e pontos previamente pesquisados por outros autores para melhor embasar as decisões tomadas posteriormente na metodologia deste trabalho, no qual apresenta-se as refatorações realizadas no aplicativo e realiza-se testes com algumas bibliotecas de identificação digital.

2 REFERENCIAL TEÓRICO

Este capítulo apresenta uma visão bibliográfica sobre os conceitos existentes neste trabalho. Inicialmente, será conceituado avaliações fonológicas, ponto de atuação do *eFono*, juntamente com os desvios fonológicos e as tecnologias existentes ao seu auxílio, focando em trabalhos que permitem a identificação do usuário. Após, o capítulo mostra uma visão bibliográfica sobre os sistemas de identificação digital, mais precisamente sobre seu histórico e sobre sua visão algorítmica. Por fim, apresenta as bibliotecas existentes para a identificação do locutor citadas em artigos.

2.1 AVALIAÇÕES FONOLÓGICAS E SUAS TECNOLOGIAS

Este trabalho possui como objetivo a pesquisa de um sistema de identificação digital para integração com o *eFono*, aplicativo voltado para as avaliações fonológicas. As avaliações fonológicas demandam identificar padrões de erros em fonemas. Uma definição possível de fonema é a de (BECHARA, 2012), quando afirma que "chamam-se fonemas os sons elementares e distintivos que o homem produz quando, pela voz, exprime seus pensamentos e emoções". Por exemplo, “peixe” e “feixe” são palavras da língua portuguesa distinguidos por apenas um fonema, /p/ e /f/ respectivamente. Ou seja, substituindo-se apenas este fonema na elicitação dessas palavras troca-se totalmente o sentido da mesma: uma representa um animal e outra representa um conceito da física. Isso justifica a definição de fonema de (FERREIRA; REGO; SANTOS GOMES, 2015), quando diz que “fonema é a menor unidade sonora capaz de estabelecer distinção entre palavras”.

Algumas pessoas acabam por omitir ou substituir fonemas no processo de aquisição de fala, especialmente consoantes e encontros consonantais (GRUNWELL, 1981; FERRANTE; BORSEL; PEREIRA, 2009). Estas falhas são conhecidas por desvios fonológicos, problemas de saúde no qual este trabalho se insere, e gera problemas na comunicação interpessoal. Estes problemas ocorrem no processo de amadurecimento da criança, fase onde ela está no processo de aprendizado da língua materna:

> A formação do sistema fonológico da criança se dá de maneira gradativa e não-linear, entre o nascimento e, aproximadamente, a idade de cinco anos. Nesse período, ocorre o amadurecimento do componente fonológico da linguagem, resultando no estabelecimento do sistema fonológico semelhante ao alvo-adulto. (GHISLENI; KESKE-SOARES; MEZZOMO, 2010)

Em suma, os desvios fonológicos caracterizam-se pela omissão ou substituição incorreta de fonemas, a menor unidade sonora de uma palavra, em crianças de até aproximadamente cinco anos de idade.

As avaliações fonológicas são procedimentos de saúde realizados para detectar algum tipo de desvio fonológico pelo paciente. São elas que realizam o diagnóstico do problema e iniciam uma terapia de fala, se necessária. Estas avaliações normalmente seguem o mesmo procedimento (CERON, 2015). Primeiro apresenta-se ao paciente uma sequencia de imagens de objetos ou ações contendo diferentes fonemas a serem testados para que o paciente elicite as mesmas. Por exemplo, se o fonoaudiólogo deseja avaliar a pronúncia do fonema /p/[2] do paciente, poderá apresentar uma ilustração de um peixe ao mesmo e ouvir a pronúncia da palavra, já que esta contém o fonema /p/ a ser avaliado. Após este ato, o profissional realiza a transcrição da fala para o alfabeto fonético[3] e compara com os fonemas considerados corretos conforme sua idade e região onde vive. Isso é importante pois a avaliação possui uma função social atrelada: considerar como correto os sotaques e gírias comuns em seu meio social. Se o paciente pronunciar corretamente esta palavra, o terapeuta considera que o mesmo não possui um desvio fonológico neste fonema.

Dado a importância destas avaliações, diversos autores estudaram maneiras de oferecer ferramentas de apoio às mesmas. Essas ferramentas, conhecidas por PET (*Phonological Evaluation Tools*), auxiliam os profissionais ao longo dos fluxos avaliativos. Como constatado por (MORO, 2018), o uso da tecnologia neste processo facilita seu acesso, minimizando a necessidade de deslocamento e suprindo a escassez de profissionais em regiões afastadas das grandes capitais, além da manutenção destes procedimentos em períodos de isolamento social como o presenciado no ano de 2020 devido à pandemia do novo coronavírus.

Estas ferramentas, muitas vezes, restringem suas implementações ao fluxo de uma avaliação fonológica: apresentar imagens e realizar as gravações dos pacientes. Porém, funções auxiliares como a unificação de avaliações em perfis de pacientes ou comparações entre diferentes avaliações acabam não sendo encontradas. Pontos como estes são importantes do ponto de vista fonológico pois apresentam uma visão mais completa sobre um paciente específico.

De acordo com (SAVOLDI; CERON; KESKE-SOARES, 2013), é importante que essas avaliações utilizem instrumentos adequados, validados e padronizados. Porém, no Brasil,

[2] Notação textual para o fonema da letra P

[3] Alfabeto fonético é um sistema de notação fonética baseado no alfabeto latino, criado pela Associação Fonética Internacional como uma forma de representação padronizada dos sons do idioma falado.

existe uma escassez de instrumentos formais e objetivos disponíveis para estas avaliações. A autora (CHUCHUCA-MÉNDEZ et al., 2016) também relata que "no domínio de terapia da fala, percebe-se que existem poucas propostas que utilizam modelagem de conhecimento para melhorar tarefas como diagnóstico, planejamento de terapia e intervenção terapêutica".

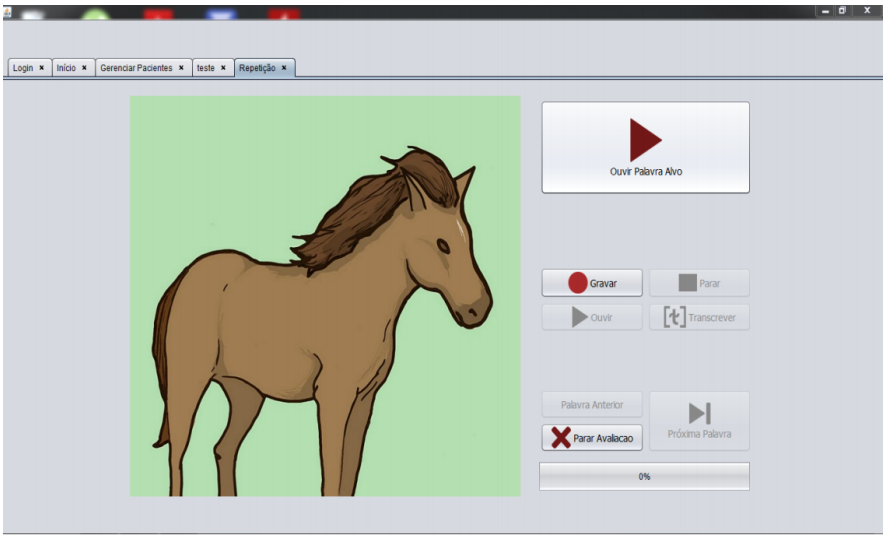
Estudos recentes mostram que o uso computacional nas terapias de fala têm ganhado força, passando a sugerir terapias virtuais para tratamentos de distúrbios de fala (JESUS; SANTOS; MARTINEZ, 2019; LEE, 2019). Um exemplo de ferramenta para terapia virtual é a *The Table to Tablet (T2T)* cuja elaboração foi "dividida em quatro fases: desenvolvimento de atividades, pré-testagem etnográfica com um exemplo da população alvo, implementação do sistema e *beta-testing*"(JESUS; SANTOS; MARTINEZ, 2019). Em sua conclusão, os autores salientam a importância deste produto e sua coexistência com metologias estruturadas fisicamente, maximizando as oportunidades de acesso à estes exames.

Outra ferramenta detalhada em (CERON, 2015) é o Instrumento de Avaliação Fonológica - INFONO (Figura 1), um *software* para avaliação do desvio fonológico cujo objetivo é disponibilizar ao fonoaudiólogo um instrumento de aplicação simples e rápida, além de aumentar a atratividade às crianças. Ele foi desenvolvido devido à percepção da carência de instrumentos com critérios de validação, fidedignidade e normatização, "que podem influenciar na definição de condutas terapêuticas e até mesmo nas reavaliações para o seguimento ou não da terapia". Em sua conclusão, a autora reitera a importância da formalização dos processos de construção de um sistema voltado às avaliações fonológicas, tendo ao final um produto testado e validado que atende à todos os fonemas da Português Brasileiro.

Pela Figura 1, percebe-se a estrutura da avaliação relatada na Seção 2.1. O sistema apresenta a figura cuja palavra deve ser elicitada pelo paciente, juntamente com alguns botões de ação para ouvir a palavra alvo referente à figura e para gravar a voz do paciente. Na parte inferior da imagem, nota-se uma porcentagem que indica o progresso da avaliação, inferindo que a ilustração do cavalo representa apenas uma de muitas palavras que o paciente deve pronunciar. O aplicativo *eFono*, alvo deste trabalho e posteriormente apresentado, possui uma estrutura semelhante à esta.

Esta seção conceituou os desvios fonológicos, as avaliações para seu diagnóstico e algumas abordagens computacionais criadas para auxiliá-las. Todas elas fizeram uso da fonoaudiologia na construção dos mesmos, porém, sem nenhuma citação a algum tipo de agrupamento de avaliações por perfis ou algum comparativo realizado entre diferentes avaliações por um mesmo

Figura 1 – Tela da palavra-alvo “Cavalo”, no INFONO



Fonte: (CERON, 2015)

paciente. Assim, evidencia-se a relevância deste trabalho, cujo objetivo de construção e integração de um sistema de identificação digital pode ser considerado um diferencial no contexto das avaliações fonológicas. Estes sistemas serão abordados na próxima seção.

2.2 SISTEMAS DE IDENTIFICAÇÃO DIGITAL

Nos tempos atuais, segundo (MORO, 2018), "a área computação móvel em geral vem ganhando um notável destaque. A cada dia surgem diferentes tipos de dispositivos móveis aumentando sua presença na vida cotidiana, desde *smartphones* à dispositivos que podem ser incorporados ao corpo para monitorar dados vitais, entre outros". Devido à esta evolução, e acelerado pela pandemia do ano de 2020, tornou-se corriqueira a migração de diversos processos dos meios físicos para os meios digitais, como compras online, transações financeiras e reuniões de negócios, como afirmado em (KIM, 2020). Em novembro de 2013, segundo (SUMNER, 2015), a gigante rede de supermercados americana Tesco anunciou a instalação de câmeras para escanear a face de seus clientes no pagamento de seus produtos, mostrando que processos como estes requerem a identificação do usuário, seja ela por senhas ou através das nossas características biométricas. Neste contexto, diversos estudos e implementações no ramo de identificação digital por biometria evoluíram para permitir essas operações com maior segurança e eficácia.

Sistemas de identificação digital são desenvolvidos há décadas; (LUIS-GARCÍA et al.,

2003) apresentam uma visão geral dos sistemas de identificação, percorrendo as tecnologias biométricas em uso na época, enquanto que (RIBARIC; FRATRIC, 2005; FRISCHHOLZ; DIECKMANN, 2000) apresentam soluções mais específicas, como o uso da palma da mão e lábios para identificação.

A definição do método de identificação digital utilizada neste trabalho passou por uma análise dos requisitos e possibilidades, levando em consideração a união de dois principais fatores: (i) possibilidade de identificação das avaliações realizadas em trabalhos anteriores, aprofundados na Seção 2.1 e (ii) sistema de valor aceitável a partir de critérios definidos em (SAINI; RANA, 2014) como acurácia, custo de memória e custo de processamento.

Para o primeiro fator, notou-se que as únicas informações já existentes nos bancos de dados do *eFono*, passíveis de uso em uma identificação, são as vozes dos pacientes nas avaliações realizadas, implicando na possibilidade única do uso da voz neste processo. Já sobre o segundo critério, o comparativo adaptado de (WAYMAN, 2001) e apresentado na Tabela 1 auxiliou na tomada de decisão para este trabalho:

Tabela 1 – Comparação entre os vários métodos de assinatura digital biométrica

Característica	Impressão digital	Palma da mão	Retina	Íris	Face	Veia	Voz
Facilidade de uso	Alta	Alta	Baixa	Média	Média	Média	Alta
Assertividade	Alta	Alta	Alta	Alta	Alta	Alta	Alta
Custo	Alto	Muito alto	Muito alto	Muito alto	Alto	Muito alto	Baixo
Aceitação pelo usuário	Média	Média	Média	Média	Média	Média	Alta
Autenticação remota	Disponível	Disponível	Disponível	Disponível	Disponível	Disponível	Possui
Compatibilidade	Parcial	Sim	Compatível	Compatível	Sim	Compatível	Sim

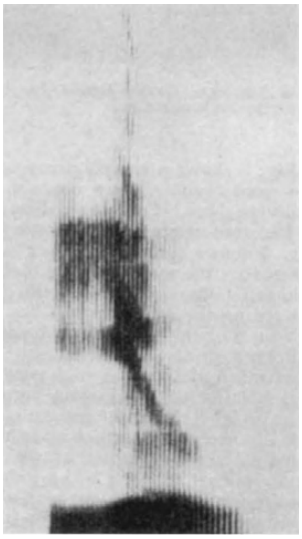
Fonte: (WAYMAN, 2001), adaptado.

A partir da análise da Tabela 1, podemos perceber um comparativo entre as diversas características de nosso corpo humano que são utilizadas para identificação digital, apresentando suas vantagens e desvantagens em seis diferentes pontos levantados pelo autor. O primeiro ponto levantado diz sobre a facilidade de uso da tecnologia, isto é, a facilidade de implementação de seu sistema, considerando a voz juntamente com a impressão digital e palma da mão como mais vantajosos. Um segundo ponto significativo dessa tabela diz respeito à sua relação entre assertividade, que indica uma maior precisão na sua identificação, e custo computacional, que se refere ao poder computacional necessário para executar um sistema que utilize a característica corporal correspondente. Analisando as sete hipóteses, percebe-se que a voz, apesar de empatar com os outros itens no quesito assertividade, acaba sendo a única com baixo custo

computacional. A partir destes critérios, a característica utilizada para identificação digital mais adequada para este caso de uso é a voz.

Em um enfoque maior sobre o uso da voz neste processo, percebe-se que pesquisas relacionadas na área são realizadas há muitas décadas, como pode ser concluído através do estudo de (KERSTA, 1962), no qual apresentava uma análise manual ao espectrograma da voz humana, conforme a Figura 2.

Figura 2 – Voz masculina entoando a palavra você, em inglês



Fonte: (KERSTA, 1962)

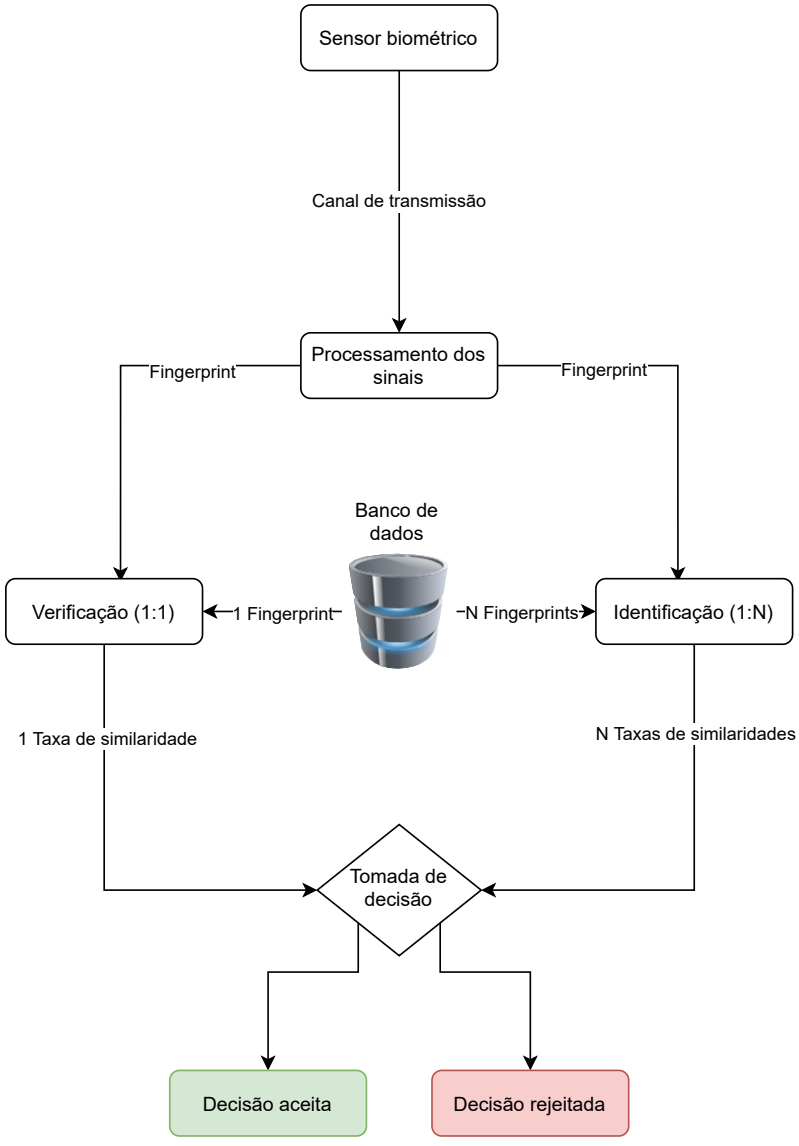
Algumas décadas depois (WANG; WANG; TAN, 2004) apresentaram um estudo onde afirmam que SVMs (*Support Vector Machine*) e o método de Dempster-Shafer[4] na construção de um sistema de identificação combinando voz e impressão digital apresentaram bons resultados, introduzindo a entrada da inteligência artificial nestes sistemas, cuja permanência se mantém até os dias atuais. Já a pesquisa de (BOLES; RAD, 2017) faz uso de *Deep Learning* para a construção de sistemas de identificação e autenticação digital. Nela, é extraído o coeficiente de *Mel-Frequency Cepstral* e uma SVM é treinada e testada em dois conjuntos de dados diferentes, um contendo áudios da *LibriSpeech*, um repositório online contendo 1000 horas de falas em inglês, e outro com vozes gravadas pelos próprios pesquisadores.

Aprofundando para uma visão arquitetural de um sistema de identificação digital por biometria, (JAIN; HONG; PANKANTI, 2000), na Figura 3, apresentam um esquemático do processo. Inicialmente, o dado necessário para a identificação é coletado através de um sensor

[4] Teoria matemática da evidência que permite combinar evidência de diferentes fontes e chegar a um grau de credibilidade (representada por uma função de credibilidade) que leva em conta toda a evidência disponível

biométrico, como uma câmera caso deseje-se utilizar a face ou um microfone no caso da voz. Como este dado deve ser coletado do usuário, normalmente este sensor encontra-se na aplicação final, enquanto que o sistema de identificação normalmente encontra-se centralizado em um servidor. Por isto, os autores apresentam a comunicação entre estes dois pontos como sendo realizada através de um canal de transmissão.

Figura 3 – Um sistema genérico de biometria



Fonte: (JAIN; HONG; PANKANTI, 2000), adaptado

Em seguida, o dado chega no processamento dos sinais, local onde encontra-se toda a lógica algorítmica e matemática responsável pela geração do identificador único do usuário (*fingerprint*). Em uma referência aos estudos supracitados, é neste processamento que estão as SVMs, os algoritmos de Deep Learning, o coeficiente de *Mel-Frequency Cepstral* e o método de

Dempster-Shafer. Este é o ponto mais importante do sistema, pois é nele que o arquivo de áudio será transformado em um *fingerprint*. Por fim, o identificador gerado no passo anterior pode ser levado à dois processos que antecedem a tomada de decisão: verificação ou identificação do usuário.

O primeiro pode ser entendido como um algoritmo que recebe dois *fingerprints*, um recém coletado e outro armazenado no banco de dados do sistema, e afirma se ambos pertencem à mesma pessoa ou não. Por exemplo, esta verificação pode ser encontrada no processo de desbloqueio de um *smartphone* através da impressão digital: o usuário cadastrou previamente sua digital e toda vez que tentar realizar o desbloqueio, o dispositivo coletará sua impressão digital através do sensor, processará os sinais e gerará um identificador único que será comparado com um identificador cadastrado previamente. Ou seja, o banco de dados fornecerá um único identificador e o sistema decidirá se pertencem à mesma pessoa ou não, desbloqueando ou não o dispositivo. Este exemplo é válido apenas para *smartphones* que permitem o pré cadastro de uma única impressão digital; atualmente, já é possível cadastrar múltiplas impressões digitais para a possibilidade de desbloqueio com mais um dedo.

Na identificação do usuário, o algoritmo recebe o *fingerprint* coletado do usuário e todos os *fingerprints* previamente cadastrados no sistema. Assim, o custo computacional para este processamento torna-se mais complexo: deve-se realizar N comparações, uma para cada identificador oriundo do banco de dados, tendo como resultado final uma lista de usuários considerados possíveis "donos"do dado coletado, como a voz, no caso deste trabalho. Este processo ocorre nas perícias criminais quando encontra-se uma impressão digital de um suspeito: compara-se com uma base de impressões digitais pré-coletadas buscando identificar um conjunto de pessoas suspeitas de possuírem a impressão digital presente no local do crime.

A tomada de decisão, última etapa do fluxo, normalmente funciona da seguinte maneira: recebe uma, no caso da verificação, ou N, no caso de uma identificação, taxa(s) de similaridade(s). A partir desta(s) taxa(s), com o auxílio de um limiar de aceitação pré-configurado no sistema, afirma se nenhuma, uma ou mais de uma pessoa é considerada aceita como autora do dado coletado inicialmente. Por exemplo, foi coletado uma voz e foi constatado que as três pessoas previamente cadastradas possuem taxas de similaridade de 30%, 50% e 90% com esta voz. Considerando um limiar de aceitação de 85%, somente a última pessoa seria identificada como autora da voz coletada. Este limiar normalmente é consolidado através de estudos iniciais, onde cadastra-se múltiplas vozes conhecidas e testa-se as mesmas no sistema. A partir das taxas de

similaridades informadas, verifica-se o maior valor de similaridade entre as vozes de duas pessoas distintas e o menor valor entre duas vozes de uma mesma pessoa. Assim, considera-se o limiar algum ponto entre estes dois valores: toda taxa acima deste limiar é considerada aceita e toda taxa abaixo é considerada não-aceita.

2.3 BIBLIOTECAS PARA IDENTIFICAÇÃO DIGITAL POR VOZ

O desenvolvimento de *software*, tanto acadêmico quanto do mercado de trabalho, possui muitas áreas de finalidade. É possível desenvolver sistemas para usuários finais, como sites e aplicativos em geral, como também é possível desenvolver sistemas para auxiliar outros desenvolvedores, onde encontra-se os SDKs[5] e as APIs[6], muito conhecidos na comunidade de desenvolvedores. Estas tecnologias abstraem um conjunto de funcionalidades computacionais e são disponibilizadas para integração via requisições web, no caso da API, ou integração local, no caso dos SDKs. Neste contexto, pode-se encontrar tecnologias complexas e indispensáveis, como o Android SDK, responsável por permitir o desenvolvimento para o desenvolvimento para dispositivos móveis, até as mais simples, como a simples biblioteca *strings.h*, comum na linguagem C e utilizada para manipulação de Strings, e a Geolocation API[7], cuja função é informar a localização do usuário a partir dos dados celulares do mesmo.

Os SDKs e APIs mostram-se extremamente úteis por abstrair funcionalidades e agilizar o desenvolvimento, minimizando os recursos alocados para desenvolver uma funcionalidade já existente. Assim, algumas bibliotecas nas mais diversas tecnologias podem auxiliar a elaboração de um módulo de identificação digital por voz. Algumas destas bibliotecas serão abordadas nas subseções a seguir.

2.3.1 Reconhecimento do Locutor

A primeira opção encontrada foi o Reconhecimento do Locutor, um dentre os diversos recursos disponíveis na plataforma de serviços de computação Microsoft Azure. Este recurso é disponibilizado através de uma API integrada via requisições HTTP, tecnologia já existente no aplicativo, e também possui a funcionalidade desejada neste projeto, identificação do locutor.

[5] *Software Development Kit*, ou kit de desenvolvimento de *software*, é um conjunto de ferramentas e bibliotecas para uso de outros programadores

[6] *Application Programming Interface*, é um conjunto de rotinas e padrões estabelecidos por um software para a utilização das suas funcionalidades por aplicativos que não pretendem envolver-se em detalhes da implementação do software, mas apenas usar seus serviços.

[7] Disponível em https://developers.google.com/maps/documentation/geolocation/overview

Porém, seu elevado custo financeiro inviabilizou sua utilização, o que levou ao descarte desta possibilidade.

2.3.2 VoiceIt

Como alternativa à Azure está a solução da empresa estadunidense VoiceIt, também no formato de API, que atua em um nicho especializado em identificação digital por biometria, com as opções de face e voz. Porém, em um estudo da plataforma, foi identificado alguns requisitos como a duração exata de 5 segundos nos arquivos de áudio e a dependência de frases pré-cadastradas. Estes pontos não contemplam o caso de uso do *eFono*, dado que as gravações coletadas nas avaliações fonológicas normalmente possuem uma duração próxima de 1 segundo e também a dinamicidade das palavras presentes nas avaliações, impossibilitando a utilização de frases fixas pré-cadastradas. Segundo (VOICEIT, 2021), estas imposições estão ligadas com a necessidade de abrangência de fonemas suficientes do Alfabeto Fonético Internacional, AFI (Alfabeto Fonético Internacional), para gerar um identificador único preciso na biblioteca, normalmente alcançado com frases de no mínimo 8 sílabas sem a repetição de fonemas. Assim, sua utilização também foi desconsiderada.

2.3.3 Recognito

No nicho das soluções para integração local encontra-se a biblioteca Recognito (CRICKX, 2014), um SDK disponível para a linguagem Java que realiza a identificação digital por voz de maneira independente do texto. Segundo o autor, a mesma foi capaz de identificar com sucesso as vozes de mais de 500 locutores de TEDs[8]. Porém, o mesmo ressalta que o contexto de uma TED possui qualidades que talvez não se encontre em contextos práticos: ótima qualidade de gravação, locutores profissionais com clareza e um bom volume e estabilidade nos ruídos de fundo dado que os áudios utilizados para treinamento e identificação foram coletados no mesmo ambiente. A Figura 4 mostra um exemplo de utilização da biblioteca para o cadastro e para identificação de um locutor.

A biblioteca, de acordo com (CHATZARAS; SAVVIDIS, 2015) e (SOUZA, 2018), ao receber uma amostra de voz inicia o processamento removendo trechos de silêncio presentes no áudio e normalizando o mesmo, objetivando maximizar o percentual de dados biométricos

[8] Tecnologia, Entretenimento e Design, é uma organização sem fins lucrativos que espalha ideias através de conversas de cerca de 18 minutos em mais de 100 diferentes idiomas.

Figura 4 – Exemplo de utilização do Recognito

```
// Cria uma nova instância da biblioteca e define o sample rate dos áudios de entrada
Recognito<String> recognito = new Recognito<>(16000.0f);

// Cadastre a amostra de áudio com a chave 'Elvis'
VoicePrint print = recognito.createVoicePrint("Elvis", new File("OldInterview.wav"));

// Identifique o locutor do áudio parametrizado
List<MatchResult<String>> matches = recognito.identify(new File("SomeFatGuy.wav"));

// Colete o resultado mais provável
MatchResult<String> match = matches.get(0);

// Verifique se a probabilidade do melhor resultado é considerada aceita
if(match.getLikelihoodRatio() >= THRESHOLD) {
    // Imprima o locutor identificado pelo sistema
    System.out.println("A voz pertence à " + match.getKey() +
            " com probabilidade de " + match.getLikelihoodRatio() + "%");
}
```

Fonte: https://github.com/amaurycrickx/recognito, adaptado

e minimizar a influência do volume da voz do locutor, respectivamente. Neste momento, o *framework* armazena os dados em um vetor de 20 dados do tipo *double*, responsável por armazenar números reais, totalizando 160 *bytes* para cada voz extraída.

Na etapa de cadastro da voz de um novo usuário, os *bytes* calculados na etapa anterior são mesclados com as características das vozes já cadastradas, armazenando o novo conjunto de dados após a mesclagem. Para a realização da identificação do locutor, é gerado o *voiceprint* utilizando a amostra de voz enviada. Depois, a biblioteca compara este dado com cada um dos *voiceprints* previamente registrados utilizando a distância euclidiana. Além da comparação com as vozes individualmente, também é calculado a distância com o Modelo Universal do sistema, uma média de todas as vozes armazenadas. Assim, a tomada de decisão se baseia na proximidade da amostra à ser identificada com as amostras presentes no sistema proporcional à proximidade com o Modelo Universal: quanto mais próximo do Modelo Universal, menor a possibilidade do locutor já possuir sua voz cadastrada; analogamente, quanto mais próximo de uma voz específica, maior a probabilidade de compartilharem o mesmo locutor. A biblioteca retorna como resultado uma lista com tamanho igual à quantidade de amostras na biblioteca contendo o identificador da voz e sua probabilidade de compartilhar o mesmo locutor, ordenado por este valor.

Em um sistema onde somente existe uma voz armazenada somado ao fato do Modelo

Universal ser a média de todas as vozes (neste caso, somente uma), este Modelo será exatamente igual à voz. Por este motivo, ao realizar uma identificação neste contexto, a mesma retornará a probabilidade de 50% de pertencer à voz já cadastrada, visto que a distância euclidiana entre a amostra enviada e a voz conhecida é a mesma na comparação com o Modelo Universal. Assim, não existindo uma proximidade maior com o Modelo ou com a voz conhecida, o sistema infere que a nova voz possui exatamente a probabilidade citada. Isto já mostra uma falha em sua posterior utilização, pois se for configurado um limiar de aceitação menor que 50%, toda voz posterior à primeira será considerada aceita pelo sistema, não realizando a sua inserção (já que pertencem à mesma pessoa, não é necessário inserir uma nova); porém, se for configurado um valor acima de 50%, não importa qual seja à segunda gravação, o sistema sempre irá considerar como não aceito, mesmo se a amostra for exatamente a mesma da anterior. Isso justifica a utilização de um limiar maior que este valor nos testes posteriores e diferentes locutores nas identificações iniciais.

2.3.4 Alizé

Como uma segunda solução para implementação local está o Alizé, um conjunto de tecnologias biométricas *open source* desenvolvido para uso em pesquisas em centros de estudos. Ela é composta de funções implementadas em C++ disponibilizadas através de uma biblioteca Android[9] para integração, o que impossibilita o seu uso no *eFono* dado que não teria a capacidade de ser executado de forma centralizada em algum servidor do sistema, critério para possuir somente uma instância com as vozes previamente cadastradas. Contudo, mostra-se possível a exportação das funções principais desta biblioteca para a utilização em um servidor. Sua utilização foi considerada para análise de seus resultados que, devido à análise inicial na sua estrutura interna, à ser dissertada no próximo parágrafo, induz à uma maior assertividade, dado que será constatado nos testes da Seção 3.3.2, tornando esta tecnologia válida para fins de pesquisa e comparação.

Conforme apresentado em (ALIZÉ, 2014), a biblioteca foi construída em uma arquitetura multicamadas para maximizar o alcance de suas diferentes necessidades de uso. Na camada de baixo nível estão as funções comuns de qualquer sistema de autenticação biométrica como treinamento do modelo e pré-processamento dos dados, funções auxiliares que atuam sobre a

[9] As bibliotecas Android são um conjunto de funções Java, arquivos com extensão .jar, mesclados com um conjunto de recursos Android em um arquivo .aar

entrada/saída dos dados e manipulação de dados, além dos modelos estatísticos principais do sistema. Acima desta camada encontra-se o módulo *LIA_SpkTools* cujo objetivo é interligar a camada de baixo nível com a de alto nível. Esta, por sua vez, foi desenvolvida em Java e disponibilizada para integração externa através de uma biblioteca Android. Para possibilitar a comunicação entre as funções de alto nível em Java com as funções de baixo nível em C++, usou-se o Android NDK (*Native Development Kit*, ou Kit de Desenvolvimento Nativo), que conforme (GOOGLE, 2020a) é "um conjunto de ferramentas que permite implementar partes do seu app em código nativo, usando linguagens como C e C++".

O funcionamento da ferramenta foi apresentado por (LARCHER et al., 2013) em seu trabalho, onde também foram realizadas e aprovadas algumas avaliações de performance mediados pelo NIST[10], mostrando o potencial desta ferramenta. Para a adição de um novo locutor, o Alizé gera um modelo estatístico para um conjunto de características do áudio de entrada. Esta geração está relacionada com o GMM, ou Modelo de Mistura Gaussiana, que quando aplicado sobre a voz gera um vetor de dados que identifica a amostra de entrada, possuindo três possibilidades de algoritmo intermediário para esta geração, todos disponibilizados pela ferramenta. Para a geração das relações entre os padrões, utilizada tanto na verificação quanto na identificação do locutor, três diferentes possibilidades de complexos algoritmos envolvendo SVMs e comparações de *i-Vector* informam um modelo com uma pontuação que reflete na probabilidade da amostra geradora deste modelo ter sido elicitada pelo mesmo locutor da amostra de reconhecimento/identificação.

As ferramentas apresentas anteriormente, em uma primeira vista, possuem potencial para se encaixar como auxiliar ao sistema de identificação digital. Os mesmos valorizam a utilização de bibliotecas, seja por integração local ou API mantidas em servidores de terceiros, alocando os recursos de pesquisa e desenvolvimento na intercomunicação de todo o sistema. Porém, os impedimentos existentes nas duas primeiras soluções impossibilitam seus encaixes neste trabalho. Para a concretização do desenvolvimento, as duas soluções viáveis, Recognito e Alizé, devem ser validadas através de testes que comprovem seus benefícios no *eFono* realizados na Seção 3.3.2.

Os estudos e análises realizadas neste capítulo apresentaram uma visão acadêmica acerca dos objetivos deste trabalho. Com eles, pôde-se entender a avaliação fonológica, trabalhos sobre tecnologias de auxílio à essas avaliações e a estruturação de um sistema de identificação

[10] *National Institute of Standards and Technology*, ou Instituto Nacional de Padrões e Tecnologia, é uma agência governamental não regulatória da administração de tecnologia do Departamento de Comércio dos Estados Unidos

digital por biometria, no qual insere-se a identificação digital por voz. No próximo capítulo será apresentada a metodologia utilizada para a construção do sistema objetivo deste trabalho e sua integração com o *eFono*.

3 METODOLOGIA

Este capítulo visa apresentar os métodos utilizados para o comparativo entre as bibliotecas de identificação digital explicadas na Seção 2.3. Inicialmente, será contextualizado o projeto no qual este trabalho está inserido, apresentando a sua arquitetura com seus módulos já existentes e o módulo-alvo. Após, apresenta-se o aplicativo que coletará os áudios à serem identificados pelo sistema, juntamente com as adaptações realizadas no mesmo. Por fim, será aprofundado o *dataset* utilizado para a realização dos testes, o ambiente para esta validação e sua execução em si, visando determinar a biblioteca mais adequada para nosso caso de uso.

3.1 ARQUITETURA DO PROJETO

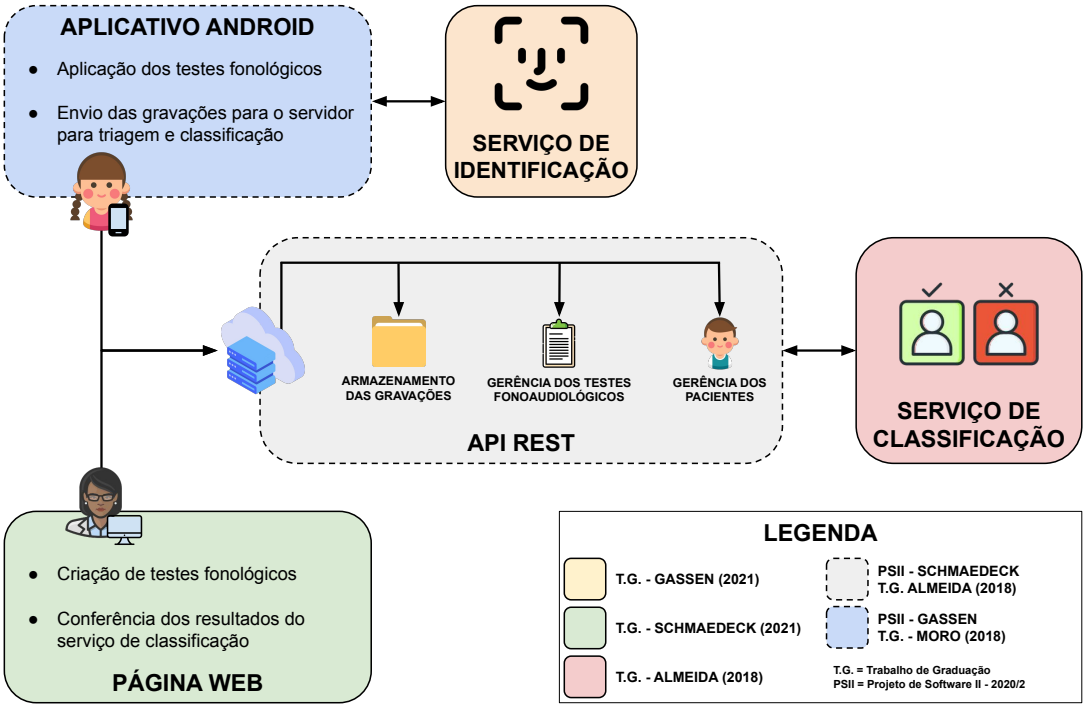
Como mencionado no Capítulo 1, este trabalho será integrado em um sistema maior dividido em cinco grandes módulos, desenvolvidos ao longo de quatro trabalhos, que serão aprofundados posteriormente. A estratégia do desenvolvimento em módulos se mostra muito eficaz em casos como este, onde um grupo de desenvolvedores atua em paralelo e é necessária a intercomunicação de diversas frentes de processamento e apresentação do sistema. Assim, os desenvolvimentos se mantém completamente isolados até a integração final. Além disso, usou-se a estratégia de cliente-servidor, apropriada para casos onde há um servidor central contendo os dados e múltiplos clientes com a necessidade de utilizá-los, adicioná-los, editá-los ou removê-los. (COULOURIS et al., 2013) define esta estratégia como "um programa em execução (um processo) em um computador interligado em rede, que aceita pedidos de programas em execução em outros computadores para efetuar um serviço e responder apropriadamente. Os processos que realizam os pedidos são referidos como clientes e a estratégia geral é conhecida como computação cliente-servidor".

A Figura 5 apresenta a arquitetura do projeto em questão, mostrando o ano em que foi desenvolvido, seu responsável e suas intercomunicações. O primeiro módulo a ser apresentado é o aplicativo Android[11], desenvolvido em (MORO, 2018) e adaptado para este trabalho, chamado de *eFono*. Ele é um dos dois clientes *front-end*[12], focado para os pacientes e possui funções como cadastrar um novo paciente, iniciar uma avaliação fonológica para um paciente e realizar a avaliação em si, enviando para o servidor as gravações das vozes do paciente para triagem e

[11] Sistema operacional de dispositivos móveis mantido pela Google
[12] Termo utilizado em referência à sistemas ou parte deles que são visíveis à usuários

classificação. Este módulo será aprofundado na Seção 3.2.

Figura 5 – Arquitetura do projeto



Fonte: acervo pessoal

Na esquerda inferior da Figura está o outro módulo de *front-end*: a página web. Seu desenvolvimento ocorreu em (SCHMAEDECK, 2021) e ele é voltado para a gerência do sistema por parte dos profissionais da área da fonoaudiologia, com funções como a criação, edição e remoção de testes fonológicos, além da conferência dos resultados das avaliações.

Na parte central, encontra-se a API Rest[13], o cerne de comunicação do sistema, implementado no trabalho de (ALMEIDA, 2018) e adaptado no trabalho de (SCHMAEDECK, 2021). Todas as informações geradas e requisitadas tanto pelo aplicativo Android quanto pela página web passam por esta central, localizada em um servidor em domínio do projeto com um banco de dados onde encontram-se dados de pacientes já avaliados, os testes fonológicos cadastrados pelos fonoaudiológicos e as gravações das avaliações.

Para permitir a integração entre os dois clientes *front-end* e a API Rest, assim como em qualquer aplicação multi-módulos, se faz necessário o uso de algum protocolo de comunicação. Para a escolha deste, observou-se uma tendência atual do uso do protocolo HTTP (*Hypertext*

[13] Estilo de arquitetura onde os sistemas solicitantes acessam e manipulam representações textuais de recursos da Web usando um conjunto uniforme e predefinido de operações sem estado

Transfer Protocol), reconhecido como o protocolo mais comum na Internet, segundo (CHEN; CHENG, 2016). Ele consiste em mensagens de requisição, enviadas por um cliente e processadas por um servidor, e mensagens de resposta, devolvidas pelo servidor ao cliente, ideal para o projeto em questão já que este não possui como requisitos o estabelecimento de túnel de comunicação em tempo real ou uma latência de comunicação extremamente baixa.

Para a realização de uma avaliação fonológica é imprescindível algum meio para classificar as palavras elicitadas por um paciente como corretas ou incorretas do ponto de vista fonético. Nos meios tradicionais, esse papel é do profissional da fonoaudiologia; porém, em um sistema digital, é necessário algum serviço para isso. Este foi o contexto da criação de um quarto módulo do projeto, o serviço de classificação de (ALMEIDA, 2018). Em uma apresentação computacional, o mesmo encontra-se no servidor da API Rest como um conjunto de funções que recebe como entrada a gravação de uma palavra e como saída devolve um valor de aceitação indicando a assertividade fonética da palavra pronunciada pelo paciente.

O último módulo é o foco deste trabalho: o serviço de identificação. Normalmente, os serviços de identificação digital por biometria se dividem em métodos para cadastros dos dados, como as vozes neste trabalho, e os métodos de identificação ou verificação. Ademais, como a relação entre cadastros realizados para um mesmo paciente e assertividade do sistema é diretamente proporcional, é interessante cadastrar o máximo possível de vozes de um mesmo paciente ao serviço. Por isso, ele é integrado diretamente ao *eFono* para reduzir o tempo de comunicação e processamento da API e para receber as vozes dos pacientes ao passo de que ocorre a avaliação. Na próxima seção será apresentado uma visão técnica do *eFono* para entendimento dos caminhos e limitações para a construção do sistema de identificação.

3.2 *EFONO*, O APLICATIVO ANDROID

Atualmente, existem apenas dois SOs para dispositivos móveis considerados relevantes, responsáveis por abranger praticamente todos os aparelhos presentes na sociedade: Android e iOS. O primeiro, mantido pela empresa estadunidense Google, está presente em todos os dispositivos com exceção do iPhone, como são chamados os dispositivos fabricados pela também estadunidense Apple, mantenedora do sistema operacional iOS. Ou seja, de um lado temos um SO abrangente e não limitado aos dispositivos da própria Google, enquanto que do outro lado temos um propositalmente limitado aos dispositivos construídos na própria empresa. Esta filosofia se propaga para o desenvolvimento voltado a estes SOs, onde uma possui uma acessibi-

lidade de desenvolvimento enorme e a outra restringe a construção de aplicativos ao seu IDE[14] XCode, somente disponível para seu sistema operacional encontrado apenas em computadores Apple.

O contexto supracitado, somado a outros fatores mencionados em (MORO, 2018), justificaram o desenvolvimento e a manutenção do aplicativo *eFono* somente para os dispositivos Android, ambos realizados através do IDE oficial da linguagem, o Android Studio[15]. Ele tem como funcionalidades a possibilidade de execução do aplicativo em dispositivos físicos ou emuladores, depuração de código e até mesmo analisadores de performance, ideias para estudos e testes nos diversos quesitos relevantes em *smartphones*: uso de CPU, uso de memória, uso de rede e consumo de bateria. Pontos como este dão robustez ao desenvolvimento Android, agilizando e qualificando o surgimento de aplicativos para seus dispositivos e suas posteriores manutenções, necessárias para este trabalho.

Após a definição das ferramentas de desenvolvimento ocorreu a primeira implementação do *eFono*, relatada com detalhes no trabalho citado anteriormente, cuja estrutura base não foi alterada para este trabalho. O mesmo inicia na tela de *login* do sistema, onde os usuários com permissões para realizar as avaliações poderiam se cadastrar para posteriormente realizar o *login* no mesmo. Inicialmente, optou-se pela liberação de três categorias de usuários: fonoaudiólogo, agente de saúde ou responsável, usuários com permissões para realizar as avaliações.

Depois de efetuado o *login* na plataforma, o aplicativo apresenta a lista de pacientes já cadastrados no sistema pelo respectivo usuário, bem como a possibilidade de adicionar um paciente à lista. É importante ressaltar que se outro usuário realizar o cadastro deste mesmo paciente, não há alguma relação ou link entre eles, afinal, este é um dos objetivos deste trabalho. A Figura 6 mostra como se da a visualização dos pacientes no eFono e suas informações, como nome, data de nascimento, idade e data de sua última avaliação no aplicativo.

Ao selecionar um paciente, o usuário tem à sua disposição a lista de possíveis avaliações pré cadastradas na API para escolher qual deseja realizar com o paciente selecionado. Cada avaliação contém uma sequência diferente de palavras e é construída com foco em algum desvio de fala. Por exemplo, caso deseja-se verificar a pronúncia do fonema /p/, escolhe-se uma avaliação que contenha palavras com o fonema alvo. Assim, ao realizá-la, o avaliado terá pronunciado uma ou mais palavras com este fonema e será possível verificar se há algum desvio no mesmo.

[14] *Integrated Development Environment* ou Ambiente de Desenvolvimento Integrado é um programa de computador que reúne ferramentas de apoio ao desenvolvimento de *software*

[15] Disponível em https://developer.android.com/studio

Figura 6 – Visualização da lista de pacientes do *eFono*

Fonte: acervo pessoal

Escolhida uma avaliação para o paciente, a mesma será dada como iniciada e será exibido a sequência de telas do processo avaliativo, onde cada uma almeja apresentar e gravar a elicitação de uma palavra.

A Figura 7 mostra a tela de coleta da palavra "bola"em uma avaliação teste de três palavras. Esta tela possui algumas funções interessantes além da gravação da pronúncia do paciente, como a possibilidade de reprodução da gravação no botão central inferior, a opção de mostrar ou esconder a palavra-alvo em texto e uma visualização na forma de onda da voz gravada em cada palavra. É nesta tela, mais especificamente no botão inferior direito da Figura 7, que ocorre a gravação das vozes de entrada do sistema de identificação. Assim, ao final de seu fluxo, o sistema já terá disponível as gravações para a aplicação do sistema objetivo deste trabalho.

Esta seção dá noção do fluxo para a realização de uma avaliação pelo aplicativo. Porém, para esta realização, são utilizadas tecnologias que estão em constante evolução. Pontos novos apareceram e outros tornaram-se defasados no contexto do desenvolvimento para Android

Figura 7 – Avaliação da palavra “bola”

Fonte: acervo pessoal

desde a primeira implementação do aplicativo, no trabalho de (MORO, 2018), até a elaboração deste trabalho. Assim, optou-se neste projeto por algumas modificações visando utilizar as ferramentas recomendadas para facilitar a possível integração com o sistema de identificação e minimizar possíveis erros. A subseção a seguir apresenta estas principais alterações e suas justificativas.

3.2.1 As adaptações realizadas

O desenvolvimento Android sempre se deu através da linguagem de programação Java, que utiliza a técnica de programação orientada a objetos, que segundo (POO; KIONG; ASHOK, 2007), é uma abordagem mais próxima do mundo real, onde existem classes de objetos com capacidade de realizar ações específicas e interações entre si. Assim, torna-se mais didática a construção de um sistema computacional, visto que sua estrutura será próxima da realidade.

Em 2017 esta supremacia do Java começou a mudar. Na Google I/O[16] daquele ano, foi anunciado o suporte oficial do Android Studio à linguagem Kotlin, o que possibilitaria o desenvolvimento de aplicativos sem o uso de Java. Dois anos mais tarde, no mesmo evento, anunciou-se a substituição de Java para Kotlin como linguagem de programação oficial para o desenvolvimento Android. A partir deste momento, as bibliotecas e práticas de programação que começaram a surgir eram disponibilizadas inicialmente para Kotlin, muitas vezes sem a posterior disponibilidade para Java. Assim, desenvolver em Java começou a ganhar tons de dificuldade e riscos, visto a falta de suporte e documentação à mesma.

Além das dificuldades comentadas no parágrafo anterior, a linguagem Kotlin por si só possui algumas vantagens em relação a Java. Segundo (GOOGLE, 2020b), um aplicativo em Kotlin requer menos código para sua implementação, resultando em aumento na produtividade por 67% dos desenvolvedores profissionais, taxa 20% menor de falhas, interoperabilidade com as bibliotecas pré-codificadas em Java e uma nova estrutura de corrotinas, o que facilita a programação assíncrona, trechos de código que executam em segundo plano como cálculos massivos ou envio de requisições à servidores.

A primeira versão do *eFono*, realizada em 2018, ocorreu em um momento novo no contexto do Kotlin, o que implica em pouco suporte da comunidade à linguagem, justificando seu desenvolvimento em Java, uma linguagem muito mais consolidada cujos problemas e dúvidas seriam mais facilmente solucionados. Porém, na realização deste trabalho, o momento do desenvolvimento em Kotlin já está em uma fase madura, onde suas documentações e bibliotecas já estão em um estágio mais avançado. Assim, optou-se pela reimplementação do aplicativo em Kotlin, utilizando as novas sintaxes e estruturas que a tecnologia oferece.

Em uma amostra do código, a Figura 8 apresenta a mesma classe do *eFono* referente às informações de uma avaliação fonológica por parte de um paciente na comparação entre Kotlin e Java. Nela, estão presentes informações como o seu identificador único, suas referências do paciente-alvo e avaliação-alvo, suas datas de início e término e a lista de passos completados pelo paciente, onde estão localizados as informações dos áudios gravados. O primeiro ponto que se percebe à partir da imagem é o tamanho da codificação das classes, sendo em Kotlin a amostra mais sucinta, fato que se deve principalmente ao acoplamento do método construtor da classe à sua definição, o que reduz a duplicidade na declaração das variáveis existentes no código Java e necessidade de atribuição das mesmas. Essa redução de código facilita a

[16] Conferência de programadores que é organizada anualmente pela Google nos Estados Unidos para apresentar técnicas de programação recomendadas pela empresa

manutenção dos mesmos em possíveis adaptações futuras.

Um segundo ponto à ser levantado é a presença do caractere '?' em algumas tipificações no código Kotlin: ele indica explicitamente que a variável em questão pode receber o valor *null*, ou seja, uma referência vazia, explicitude não existente na linguagem Java. Este fato, além de impossibilitar a criação de objetos sem este caractere com valores nulos, obriga ao programador o tratamento para o caso nulo, o que na prática reduz a quantidade de exceções de ponteiro nulo, quando tenta-se acessar um valor interno de uma variável que não foi inicializada. Por fim, percebe-se a diferença sintática na declaração da variável *KEY*, utilizada como chave para transportar objetos dessa classe entre as *activities*[17] do aplicativo. Em Java, ela se dá a partir da sintaxe *public static final String KEY*, o que configura uma *String* constante pública da classe. Em Kotlin, as variáveis, constantes e métodos das classes estão localizadas dentro do trecho *companion object*, centralizando suas localizações. Para mais, as constantes possuem a palavra-chave *val*, diferente de *var* utilizada para as variáveis, facilitando o entendimento do código pelo programador.

Figura 8 – Comparação da classe *PatientEvaluation* em Kotlin, à esquerda, e Java, à direita

```kotlin
import br.ufsm.inf.efono.model.evaluation.Evaluation
import java.io.Serializable
import java.util.Date
import kotlin.collections.ArrayList

class PatientEvaluation(
    var id: Int? = null,
    var patient: Patient,
    var evaluation: Evaluation,
    var startDate: Date,
    var finishDate: Date? = null,
    var completedSteps: ArrayList<PatientEvaluationStep>? = ArrayList()
) : Serializable {

    companion object {
        const val KEY = "PatientEvaluation"
    }

}
```

```java
import br.ufsm.inf.efono.model.evaluation.Evaluation;
import java.io.Serializable;
import java.util.ArrayList;
import java.util.Date;

public class PatientEvaluation implements Serializable {

    public static final String KEY = "PatientEvaluation";

    private Integer id;
    private Patient patient;
    private Evaluation evaluation;
    private Date startDate;
    private Date finishDate;
    private ArrayList<PatientEvaluationStep> completedSteps;

    public PatientEvaluation(
            Integer id,
            Patient patient,
            Evaluation evaluation,
            Date startDate,
            Date finishDate,
            ArrayList<PatientEvaluationStep> completedSteps) {
        this.id = id;
        this.patient = patient;
        this.evaluation = evaluation;
        this.startDate = startDate;
        this.finishDate = finishDate;
        this.completedSteps = completedSteps;
    }

    // getters e setters

}
```

Fonte: acervo pessoal

Além da conversão da linguagem, percebeu-se outro ponto passível de melhoria no apli-

[17] Classe que serve como ponto de entrada para a interação de um *app* com o usuário através de uma interface

cativo. (MORO, 2018) optou pela utilização da biblioteca *OkHttp*, desenvolvida pela Square[18] para as requisições com a *API Rest* do sistema; porém, existe a biblioteca *Retrofit*, também desenvolvida pela Square, que possui algumas melhorias em relação à primeira, com destaque para a possibilidade de criação de métodos que abstraem toda a comunicação HTTP, facilitando a comunicação com o código nativo. A Figura 9 apresenta um exemplo desta biblioteca, onde há um método em Kotlin chamado *listRepos* com a anotação *@GET("users/{user}/repos")*. Esta sintaxe indica que quando for executado esta função, na verdade será realizada uma função GET para o endereço base parametrizado na biblioteca adicionando-se ao final o endereço relativo, localizado dentro da anotação do método, resultando no endereço *base_url/users/{user}/repos*, com a parte do endereço entre chaves será substituída pelo parâmetro *@Path* do método. Como retorno, o método devolverá um objeto do tipo *Call<List<Repo»*, que retornará a lista de repositórios devolvida na requisição. Deste modo, minimiza-se o trabalho necessário para realizar as integrações com o servidor e serialização/desserialização dos objetos de envio/resposta, respectivamente.

Figura 9 – Exemplo de interface da Retrofit

```
interface GitHubService {
  @GET("users/{user}/repos")
  fun listRepos(@Path("user") user: String): Call<List<Repo>>
}
```

Fonte: https://square.github.io/retrofit, adaptado

Ao final das adaptações realizadas, a estrutura do aplicativo como um todo está implementada da maneira recomendada segundo as diretrizes do desenvolvimento Android. Assim, temos uma base sólida com riscos minimizados para realização de testes com as bibliotecas apresentadas em 2.3, à ser desenvolvido na próxima seção.

3.3 OS TESTES COM AS BIBLIOTECAS DE *VOICEPRINT*

Nesta seção se dará início a construção do sistema de identificação digital por voz, iniciando com a pesquisa de bibliotecas de auxílio que possam encapsular funções como registro e identificação da voz, passando pela escolha dos *datasets* para a realização dos testes e finali-

[18] Companhia responsável por desenvolver bibliotecas open source, disponível em https://square.github.io/

zando com a avaliação dos resultados buscando entender a melhor maneira de configuração do sistema.

3.3.1 O *dataset* utilizado

O sistema do *eFono* é desenvolvido na Universidade Federal de Santa Maria, onde também está sediado o trabalho de (CERON, 2015), o INFONO. Devido à seu estágio mais avançado, e sua maior proximidade com pesquisadores da área de fonoaudiologia, o INFONO já foi utilizado em mais de 1000 avaliações, cada uma com 84 palavras elicitadas por diferentes crianças. Uma base de dados desta dimensão com dados obtidos em um contexto similar ao aplicativo reduz possíveis distorções nos resultados de testes motivado por poucas amostras ou *datasets* não equivalentes. Além do mais, soma-se à isto a presença de 91 pacientes que possuam mais de uma avaliação, sendo o objetivo principal deste trabalho a descoberta de pacientes comuns entre diferentes avaliações.

O *dataset* foi gerado a partir da extração das informações do banco de dados, resultando em um arquivo .csv[19] e um diretório contendo todas as gravações. A Tabela 2 mostra as primeiras amostras de dados presentes no arquivo de texto, onde estão localizadas todas as informações coletadas das avaliações.

Tabela 2 – Amostras de dados do *dataset*

ID Avaliação	Nome	Data	ID Gravação	Sexo	Cidade	Estado	Data de Nascimento	Palavra
1	Nome do Paciente	2014-09-11	1	NULL	Santa Maria	Rio Grande do Sul	2006-09-10	Anel
1	Nome do Paciente	2014-09-11	2	NULL	Santa Maria	Rio Grande do Sul	2006-09-10	Barriga
1	Nome do Paciente	2014-09-11	3	NULL	Santa Maria	Rio Grande do Sul	2006-09-10	Batom
1	Nome do Paciente	2014-09-11	4	NULL	Santa Maria	Rio Grande do Sul	2006-09-10	Bebê

Fonte: acervo pessoal.

Em uma análise inicial, percebe-se que a tabela é orientada à gravação, onde cada linha representa uma gravação já que o identificador que é incrementado é o da gravação; porém, existem mais informações relevantes além da gravação para este trabalho. Na primeira coluna, encontra-se o identificador único da avaliação cuja cidade e estado de aplicação também estão presentes na tabela. Como complemento das informações relacionadas à avaliação, encontra-se o nome do paciente avaliado, seu sexo e sua respectiva data de nascimento. Em um possível agrupamento por cada passo da avaliação, a tabela também informa o respectivo ID e a palavra

[19] Arquivos de texto comuns no armazenamento de informações tabulares que possuem dados separados por vírgula onde todas as linhas possuem a mesma organização

que foi apresentada ao paciente, cuja tentativa de pronúncia encontra-se gravada em arquivos .wav[20] no diretório anexo.

Para possibilitar a relação entre cada informação de gravação e seu respectivo arquivo, seu nome possui a palavra-alvo seguida por um caractere '_', finalizando com o ID da gravação, além da extensão supracitada. Por exemplo, a gravação do áudio da primeira linha da tabela é um arquivo 'Anel_1.wav', já que seu ID é 1 e sua palavra-alvo é Anel. Assim, viabiliza-se a relação entre os arquivos de áudio e as informações das avaliações, possibilitando a utilização deste *dataset* para a realização de testes visando entender o comportamento das bibliotecas escolhidas neste conjunto de dados.

3.3.2 Ambiente de testes

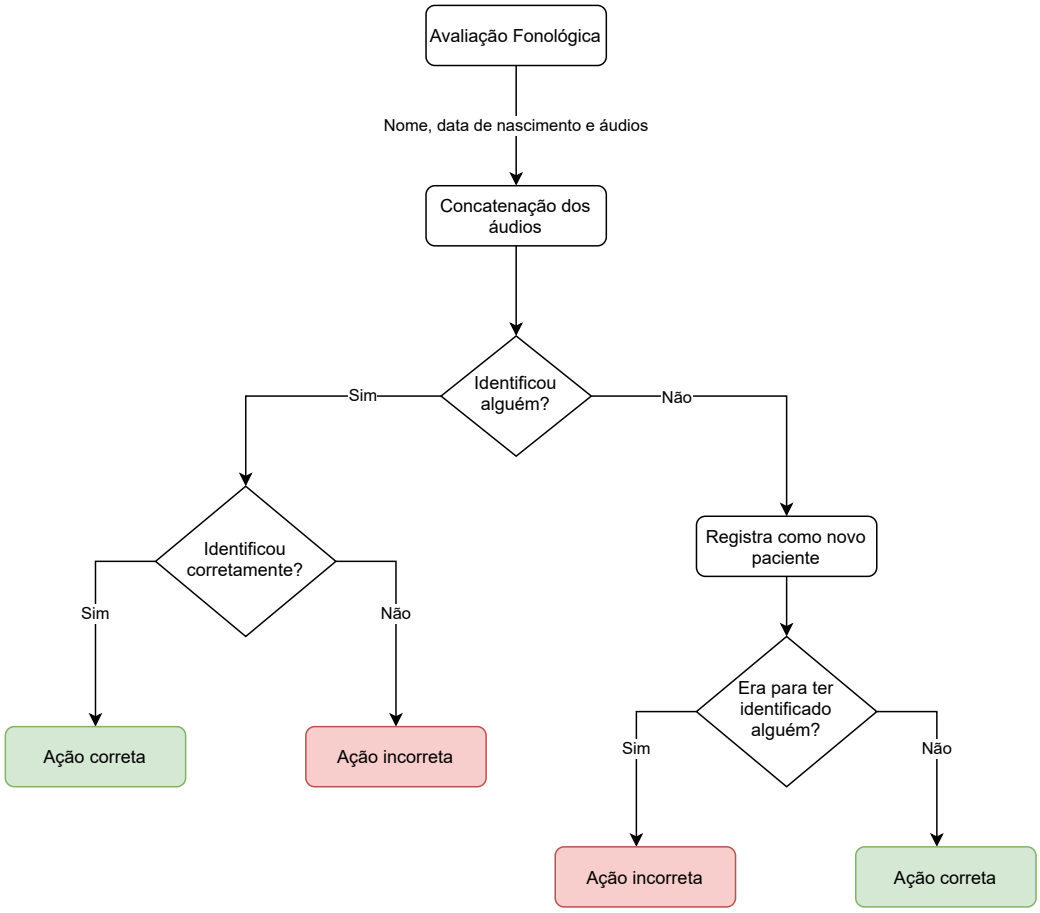
Em posse dos dados de mais de 1000 avaliações, precisamente 1045, e com o conhecimento de sua organização, o próximo passo é criar um ambiente de testes que possa ler estas informações, aplicar os *frameworks* escolhidos anteriormente para identificação e cadastro dos pacientes e verificar os seus resultados. Como os testes serão realizados em duas bibliotecas, uma disponível para integração via projeto desktop Java e outra via projeto Android, se faz necessária a criação de dois ambientes de testes com o mesmo fluxograma. Este, por sua vez, foi elaborado à partir do entendimento do processo avaliativo presente no aplicativo e aprofundado na Seção 3.2, e está descrito na Figura 10.

O fluxo inicia com a coleta dos dados das avaliações fonológicas e suas devidas correções, especialmente no nome e data de nascimento, devido à falhas de digitação no cadastro do paciente, como exemplificado em uma data de nascimento cujo ano era '0009', induzindo uma correção do primeiro caractere de '0' para '2'. Em seguida, os áudios coletados na avaliação passam por um processo de concatenação resultando em um único áudio com todas as palavras elicitadas. Este processo é interessante para maximizar a quantidade de fonemas na amostra e torná-la mais única no quesito biométrico.

Neste momento, têm-se o nome, data de nascimento e um único áudio do paciente que passou pela avaliação fonológica. Para a primeira tomada de decisão, onde o sistema irá identificar alguém, é enviado o áudio através dos métodos fornecidos pelos SDKs: *SimpleSpkDetSystem.addAudio(File)* seguido de *SimpleSpkDetSystem.identifySpeaker()* no Alizé[21] e *Re-*

[20] Arquivos de áudio desenvolvidos pela Microsoft e IBM para armazenamento em PCs.
[21] Disponível em https://github.com/ALIZE-Speaker-Recognition/android-alize

Figura 10 – Fluxograma do ambiente de testes



Fonte: acervo pessoal

cognito.identify(File)[22] no caso do Recognito. Estes métodos retornam o locutor já cadastrado com maior probabilidade de ser o dono deste áudio, além da sua probabilidade. Logicamente, se não houver nenhum locutor cadastrado, o sistema não retornará nenhum nome. A partir daí o fluxo é dividido em duas sub-árvores: o caso de haver algum locutor identificado com probabilidade considerada aceita e o caso contrário. O Recognito, em específico, não possui um limiar próprio ou recomendado de aceitação, o que obrigou à construção do sistema testes extras com diferentes limiares, cujos valores escolhidos foram de 55% a 80% e variação de cinco pontos percentuais.

Na sub-árvore da esquerda, onde o sistema inferiu que existe uma probabilidade aceitável de algum outro locutor possuir a voz do áudio em questão, o fluxo segue para uma tomada de decisão importante: "o sistema identificou corretamente o locutor?". Em termos práticos é impossível responder esta pergunta com 100% de certeza, entretanto, após análises acerca das

[22] Disponível em https://github.com/amaurycrickx/recognito

informações fornecidas pelo *dataset*, que são as mesmas que o *eFono* fornece, mostrou-se viável para responder à pergunta acima a utilização do conhecido algoritmo Distância de Levenshtein aplicado sobre o nome concatenado com a data de nascimento do paciente avaliado com o retornado pelo sistema. Este algoritmo, segundo (HEERINGA, 2004), é uma "medida sensitiva da distância entre duas sequências de caracteres, conhecido na computação como *strings* buscando o menor custo em um conjunto de operações como inserções, remoções ou substituições necessárias para transformar uma *string* em outra", e possui como retorno valores entre 0,0 e 1,0 na implementação escolhida. Ao aplicar este algoritmo com as palavras "Mateus"e "Mateus", o resultado é 1,0, enquanto que entre "Mateus"e "Marcos"o resultado é 0,5.

O limiar configurado para a Distância de Levenshtein foi 0,8, adequado para tolerar simples erros de digitação no nome ou data de nascimento. Porém, existe a possibilidade deste algoritmo aprovar a identificação caso ela aconteça com pacientes com nomes e data de nascimento muito próximas, como por exemplo "Júlio da Silva2010-10-10"e "Júlia da Silva2010-08-10", onde existe diferença de um caractere entre seus nomes e dois dígitos em seus meses de nascimento. Neste caso, o algoritmo teria 0,88 de resposta e afirmaria que a identificação foi correta, gerando um falso positivo e um erro na contagem dos resultados. Contudo, esta possibilidade em uma amostra real de 1000 pessoas é relativamente baixa, então os resultados finais não devem ser alterados.

Na sub-árvore da direita do fluxo de testes, temos o contexto "o sistema disse que o áudio da avaliação não pertence à ninguém da base". Neste ato, o sistema realiza o cadastro deste paciente por considerá-lo novo através dos métodos *Recognito.createVoicePrint(String key, File audio)* e *SimpleSpkDetSystem.addAudio(File)* mais *SimpleSpkDetSystem.createSpeakerModel(String key)*, onde o parâmetro *key* é o identificador desta pessoa. Como citado anteriormente, este identificador utiliza o nome concatenado com a data de nascimento da criança para possibilitar a validação do locutor identificado no fluxo anterior. Nesta chave, também faz-se necessário um valor único que é usado no caso de erro da identificação, pois se o locutor Luiz dos Santos nascido em 10 de fevereiro de 2010 já tiver sido registrado, passar novamente por uma avaliação e o sistema identificar como uma pessoa ausente na base de vozes cadastradas, não será possível cadastrá-lo pois a sua chave, "Luiz dos Santos2010-02-10"já está presente, não satisfazendo o critério de unicidade. O valor único utilizado neste caso é o *timestamp*[23], que será único em um sistema sequencial.

[23] Segundos que se passaram desde a *unix epoch*, no início do dia 1º de janeiro de 1970

Para descobrir se o sistema estava correto ao não identificar algum locutor anterior, faz-se uso do mesmo algoritmo de comparação de *strings*, desta vez entre a chave do paciente avaliado e todas as chaves de quem já foi cadastrado anteriormente, buscando o maior resultado e verificando: existe uma pessoa cadastrada anteriormente cujo nome + data de nascimento, ao ser parametrizado para a Distância de Levenshtein, possui um resultado maior que 0,8? Se sim, a identificação foi incorreta pois este locutor já foi cadastrado anteriormente. Se não, ação correta.

Assim, ao final da execução deste fluxograma, nas 1045 avaliações e nas duas opções de bibliotecas, será possível calcular a taxa de assertividade do sistema utilizando contadores para as ações corretas e incorretas. A utilização do mesmo fluxograma para os dois testes é extremamente importante para que os resultados não divirjam pelo ambiente em que foram coletados. Para mais, a única diferença prática entre os dois ambientes de testes foi o pré-cadastro de 30 avaliações no Recognito para a melhor distinção entre seu Modelo Universal e seus locutores individualizados, explanado em 2.3.3. Os resultados desta execução serão tema da próxima subseção, dedicada à esta discussão.

3.3.3 Resultados dos testes

Após a realização das sessões espelhadas de testes, o primeiro ponto de discussão refere-se ao tempo necessário em cada plataforma. O Recognito gerou seus resultados em poucos minutos, enquanto que o Alizé precisou de cerca de 12 horas para executar as 1045 avaliações e produzir suas taxas de assertividades, diferença ligada a dois principais fatores.

O Recognito, por ser uma biblioteca Java, passou por sua bateria de testes em uma implantação *desktop*, sendo executada em um computador robusto com alto poder de processamento, o que diminui seu tempo de execução. Além disso, a tecnologia interna ao SDK, aprofundada anteriormente, apresenta uma conversão do arquivo de áudio em um vetor de 20 posições e um cálculo de distância euclidiana entre este vetor e os outros previamente armazenados, o que sugere uma execução mais rápida por sua simplicidade em comparação com o Alizé.

Este, por sua vez, executou em um emulador no Android Studio com capacidade computacional inferior à máquina supracitada, afinal, esta é uma premissa básica do desenvolvimento *mobile*: possuir *hardware* inferior ao desenvolvimento *desktop*. Além disso, o Alizé utiliza uma série de transformações e normalizações tanto do arquivo de entrada quanto dos áudios previa-

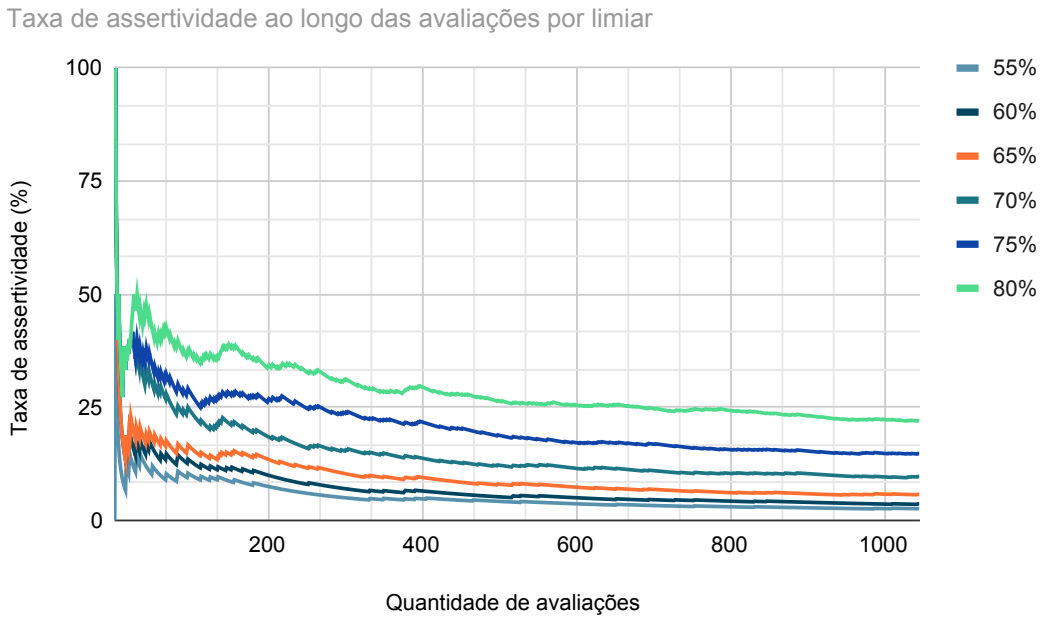
mente cadastrados para identificar o locutor que aumenta o tempo de processamento. O cálculo realizado em cada áudio previamente cadastrado, nesse caso, requer um alto grau de processamento dado que o tempo de execução de cada avaliação foi crescendo exponencialmente: a primeira avaliação durou menos de 1 segundo enquanto que a última avaliação durou cerca de 10 minutos.

Como os testes foram realizados em dois ambientes com bibliotecas diferentes, justifica-se suas discussões de forma individual, alimentado pela diferença percebida nos resultados.

3.3.3.1 Recognito

Com a utilização da biblioteca Recognito, os resultados visualizados na Figura 11 foram calculados após seis rodadas de execução do ambiente de teste, onde foi utilizado em cada rodada uma configuração diferente de limiar de 55% até 80% variando em cinco pontos percentuais. Esta configuração foi necessária, como explicado anteriormente, pois a biblioteca e o autor não estabelecem ou recomendam algum limiar, ficando a cargo da implantação esta decisão.

Figura 11 – Resultados utilizando a biblioteca Recognito



Fonte: acervo pessoal

O gráfico possui como eixo vertical a taxa de assertividade, em porcentagem, do sistema,

calculado em cada etapa do processo, apresentado no eixo horizontal com a quantidade de avaliações cadastradas até o momento. Percebe-se que as taxas iniciais sofrem muita variação pois cada novo acerto ou erro tem maior influência no cálculo final, estabilizando no final. Essa estabilidade alcançada ao longo das avaliações é de extrema importância, minimizando ruídos que poderiam ocorrer com baixa quantidade de amostras.

Em uma análise do gráfico, percebe-se que quanto maior o limiar configurado, melhores os resultados. Porém, não necessariamente isto indica que aumentando ainda mais o limiar irá resultar em um sistema com taxas aceitáveis para uma integração, pois se fosse configurado o limiar para 100%, somente áudios idênticos seriam considerados aprovados pela identificação. A utilização do limiar máximo resultaria em uma taxa de assertividade de aproximadamente 91%, pois este percentual de avaliações deve ser considerado não-identificado pelo *framework*, enquanto que os outros 9% das avaliações possui locutores já existentes na parcela avaliada anteriormente. Contudo, não identificar nenhum caso corretamente é o mesmo que não utilizar sistema algum, ou seja, esta configuração é inválida.

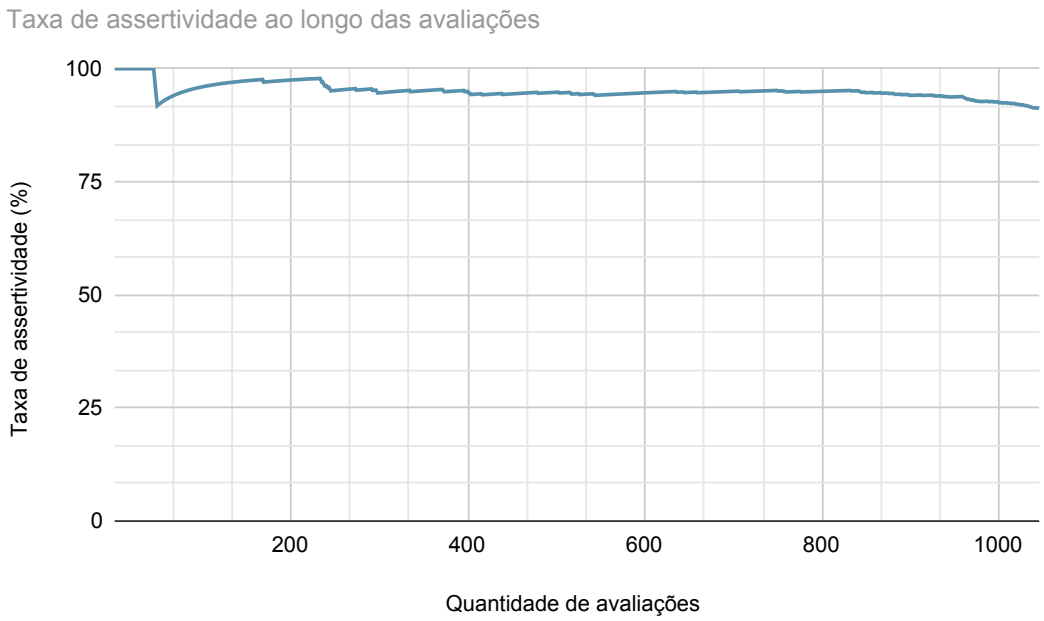
Os valores finais variaram de cerca de 2,5% até 22% com os limiares de 55% até 80%, respectivamente, aumento este justificado pelo ponto levantado anteriormente. Ademais, a falta de um crescimento em algum ponto da linha temporal de avaliações, como um possível ganho de assertividade independente das características do *dataset*, assinala para a ineficiência desta biblioteca no caso de uso do *eFono*.

O trabalho de (SOUZA, 2018), que utiliza a mesma biblioteca, mostra uma maior assertividade em sua utilização. A autora não citou a taxa de assertividade do sistema de identificação por voz em específico; porém, seu caso de uso mostrou que o fluxo de execução possuía apenas 10 diferentes locutores, onde em seu cadastro foi utilizado frases com a maximização da variedade de fonemas visando abranger todas as características de falas dos mesmos. Isto pode ter aproximado o caso de uso do caso citado pelo autor da biblioteca (CRICKX, 2014), onde o sistema acertou corretamente mais de 500 locutores de TEDs, relembrando a possibilidade de discrepância deste valor devido à qualidade sonora. Para mais, a utilização de microfones de *smartphones* ao invés de microfones de computadores, utilizados no *dataset* deste trabalho, gera uma melhor qualidade sonora que pode ter auxiliado na baixa assertividade destes testes.

3.3.3.2 Alizé

Na mesma linha de visualização dos resultados da alternativa anterior, com exceção da pluralidade de limiares do sistema motivado pelo fornecimento de um limiar interno, o Alizé apresentou melhores resultados como pode ser visualizados na Figura 12. Estes resultados foram obtidos através de um emulador de um *smartphone* Android, executado no Android Studio, com a versão 11 do SO.

Figura 12 – Resultados utilizando a biblioteca Alizé



Fonte: acervo pessoal

A taxa final de assertividade foi de 91,20%, com 953 acertos e 92 erros ao longo dos cadastros, resultando em um erro a mais do que a quantidade de avaliações que deveriam ser identificadas como já presentes na base. Contudo, ao contrário do falso resultado de sucesso encontrado no limiar máximo do Recognito, neste caso o resultado mostrou ser fidedigno à proposta da biblioteca, a partir do fato de que a mesma errou em algumas decisões onde deveria considerar o locutor como novo, mas também acertando outros casos onde realmente era o mesmo locutor. Ou seja, os resultados do Alizé realmente mostram como a biblioteca possui qualidade em comparação com algoritmos mais simples de identificação.

Outro ponto positivo desta solução é a sua estabilidade no intervalo de valores entre 100% e 90% ao longo de todas as verificações, não se fazendo necessário algum pré-cadastro

de locutores no sistema dado que o sistema alcançou uma sequência de quase 50 acertos consecutivos inicialmente.

4 CONCLUSÃO

Os desvios de fala são problemas existentes em crianças de 1 a 5 anos de idade e que devem ser diagnosticados por profissionais da área fonoaudiológica, que são capacitados com anos de estudo à problemas como este. Assim como em qualquer problema social ou de saúde, aumentar a gama de possíveis soluções, ainda mais aliado ao fato de utilizar a pesquisa brasileira para isso, é sempre válida.

Algumas tecnologias de apoio às avaliações que diagnosticam estes desvios, como mencionados na Seção 2.1, foram desenvolvidos em conjunto aos profissionais capacitados e já são utilizados como ferramentas de apoio às avaliações em sua forma mais comum: a elicitação das palavras-alvo em um encontro com o fonoaudiólogo para verificação de suas pronúncias. Contudo, as ferramentas existentes não apresentam alguma forma de identificação ou relação entre as avaliações realizadas por um mesmo paciente de uma maneira inerente às falhas humanas de digitação nos dados do paciente. Esta conclusão motivou a criação deste trabalho para realizar esta identificação através da voz que, se finalizada com sucesso, possibilitaria funções como comparação entre as avaliações de um mesmo paciente para verificação de evolução ou regressão nos seus desvios ao longo dos anos.

A identificação digital biométrica, por si só, já é um grande desafio para a computação, e não foi diferente para este trabalho. Diferentes casos de uso, a possibilidade de fraudes envolvidas como vozes gravadas de terceiros ou utilização de rostos em fotos, são o alicerce para a criação de muitas empresas computacionais dedicadas à isto, como a *VoiceIt*, cuja API foi cogitada neste trabalho e descartada devido justamente à sua especificidade de caso de uso, com frases de tamanho fixo e variedade de fonemas.

A presença de soluções livres para uso na internet, premissa de fóruns e convenções ao redor do mundo como o FISL[24], incentiva e possibilita a pesquisa nas universidades brasileiras, como foi o caso do *Alizé* e do *Recognito*, duas ferramentas propostas para a identificação do locutor a partir de arquivos de áudio, para este trabalho. Assim, pôde-se testar seus desempenhos com o *dataset* coletado por um projeto sediado nesta mesma Universidade.

Com o *dataset* disponível, este trabalho criou dois ambientes de testes equivalentes para realizar a comparação entre as bibliotecas *Recognito* e *Alizé*, um em Java e outro em Android, respectivamente. Estes ambientes buscaram avaliar a taxa de assertividade das bibliotecas em

[24] Fórum Internacional de Software Livre, realizado quase anualmente em Porto Alegre, no Rio Grande do Sul

identificar os locutores das avaliações fonológicas presentes no *dataset*.

Os resultados alcançados foram extremamente opostos entre as bibliotecas, ilustrando a diferença de implementação das mesmas: a solução mais robusta obteve uma melhor performance, maior até do que se previa. Esta diferença poderia ser menor com gravações de melhor qualidade, porém, as limitações no *dataset* utilizado obscureceram este estudo. Com as amostras utilizadas, o resultado de mais de 90% de acertos seria válido para justificar a integração com esta biblioteca. Contudo, a sua forma direta de disponibilização, através de uma biblioteca *.aar*, inviabiliza esta integração no período de realização deste trabalho. Computacionalmente falando, não é escalável realizar o *download* de todos os áudios já cadastrados para os *smartphones* no momento em que desejam que esta identificação seja realizada, isso sem considerar a duração para cadastrar todos os locutores e realizar a identificação do novo.

Este trabalho justifica-se como válido para o contexto da identificação digital por voz, pois apresentou algumas soluções para a concretização de seu objetivo explanando-as individualmente com seus prós e contras e justificando sua não-integração. Custo financeiro no caso do Reconhecimento do Locutor, quebra de requisitos no *VoiceIt*, baixa performance com o *dataset* utilizado no Recognito e inviabilidade de integração com a solução Android do Alizé.

Além disso, fez-se necessário, como relatado na Seção 3.2.1, realizar adaptações ao aplicativo visando adequá-lo às mudanças do desenvolvimento Android nos dois anos que se passaram de seu desenvolvimento, como a conversão de Java para Kotlin, tornando-o mais robusto para estudos futuros e a possível continuação do desenvolvimento do *eFono*.

4.1 TRABALHOS FUTUROS

Os resultados encontrados com a utilização do Recognito foram inferiores aos encontrados no trabalho de (SOUZA, 2018) e apontados pelo autor da biblioteca, onde a taxa de acertos de manteve em 100% para mais de 500 locutores de TEDs. Isso pode ter sido motivado pelo *dataset* utilizado, cujas gravações foram realizados em computadores de mesa sem a preocupação com sua qualidade como no caso das TEDs. Um primeiro passo que pode ser considerado é a testagem com áudios coletados por *smartphones*, como no trabalho da autora, cuja qualidade sonora é extremamente superior.

Um outro estudo importante em aberto e se concretizado, possibilitaria a integração, é a exportação do sistema interno ao Alizé, solução que obteve bons resultados quando executados localmente porém impossível de ser integrado ao aplicativo. Seus algoritmos são desenvolvidos

em C++ sem a utilização de recursos exclusivos do Android, o que aparentemente torna viável arquiteturalmente sua exportação para outros sistemas que podem ser executados no servidor do projeto. Se implementado sem alteração em sua performance, o objetivo geral deste trabalho seria alcançado com sucesso.

REFERÊNCIAS

ALIZÉ. **ALIZE Speaker Recognition Platform**. Disponível em: https://github.com/ALIZE-Speaker-Recognition. Acessado em: jan. 2021.

ALMEIDA, A. T. R. Desenvolvimento de uma API Rest para um sistema de auxílio na triagem de desordens da fala infantil. **Santa Maria: Universidade Federal de Santa Maria**, [S.l.], 2018.

BECHARA, E. **Moderna gramática portuguesa**. [S.l.]: Nova Fronteira, 2012.

BERBERIAN, A. P. **Fonoaudiologia e educação**. [S.l.]: Plexus Editora, 1995.

BOLES, A.; RAD, P. Voice biometrics: deep learning-based voiceprint authentication system. In: SYSTEM OF SYSTEMS ENGINEERING CONFERENCE (SOSE), 2017. **Anais...** [S.l.: s.n.], 2017. p.1–6.

CERON, M. I. Instrumento de Avaliação Fonológica (INFONO): desenvolvimento e estudos psicométricos. **Santa Maria: Universidade Federal de Santa Maria**, [S.l.], 2015.

CHATZARAS, A.; SAVVIDIS, G. **Seamless speaker recognition**. 2015.

CHEN, J.; CHENG, W. Analysis of web traffic based on HTTP protocol. In: INTERNATIONAL CONFERENCE ON SOFTWARE, TELECOMMUNICATIONS AND COMPUTER NETWORKS (SOFTCOM), 2016. **Anais...** [S.l.: s.n.], 2016. p.1–5.

CHUCHUCA-MÉNDEZ, F. et al. An educative environment based on ontologies and e-learning for training on design of speech-language therapy plans for children with disabilities and communication disorders. In: IEEE CACIDI 2016-IEEE CONFERENCE ON COMPUTER SCIENCES. **Anais...** [S.l.: s.n.], 2016. p.1–6.

COULOURIS, G. et al. **Sistemas Distribuídos-**: conceitos e projeto. [S.l.]: Bookman Editora, 2013.

CRICKX, A. **Recognito**: text independent speaker recognition in java. Disponível em: https://github.com/amaurycrickx/recognito. Acessado em: jan. 2021.

FERRANTE, C.; BORSEL, J. V.; PEREIRA, M. M. d. B. Análise dos processos fonológicos em crianças com desenvolvimento fonológico normal. **Revista da Sociedade Brasileira de Fonoaudiologia**, [S.l.], v.14, n.1, p.36–40, 2009.

FERREIRA, A. L. B.; REGO, E. G. S. S.; SANTOS GOMES, N. dos. ANÁLISES COMPARATIVAS DE CONCEITO DE FONEMAS EM LIVROS DIDÁTICOS. **Revista Philologus**, [S.l.], 2015.

FRISCHHOLZ, R. W.; DIECKMANN, U. BiolD: a multimodal biometric identification system. **Computer**, [S.l.], v.33, n.2, p.64–68, 2000.

GHISLENI, M. R. L.; KESKE-SOARES, M.; MEZZOMO, C. L. O uso das estratégias de reparo, considerando a gravidade do desvio fonológico evolutivo. **Revista CEFAC**, [S.l.], v.12, n.5, p.766–771, 2010.

GOOGLE. **Android NDK**. Disponível em: https://developer.android.com/ndk. Acessado em: jan. 2021.

GOOGLE. **Abordagem Kotlin do Android**. Disponível em: https://developer.android.com/kotlin/first. Acessado em: jan. 2021.

GRUNWELL, P. **The nature of phonological disability in children**. [S.l.]: Academic Press London, 1981.

HEERINGA, W. J. **Measuring dialect pronunciation differences using Levenshtein distance**. 2004. Tese (Doutorado em Ciência da Computação) — University Library Groningen][Host].

JAIN, A.; HONG, L.; PANKANTI, S. Biometric identification. **Communications of the ACM**, [S.l.], v.43, n.2, p.90–98, 2000.

JESUS, L. M.; SANTOS, J.; MARTINEZ, J. The Table to Tablet (T2T) Speech and Language Therapy Software Development Roadmap. **JMIR Research Protocols**, [S.l.], v.8, n.1, p.e11596, 2019.

KERSTA, L. G. Voiceprint identification. **Nature**, [S.l.], v.196, n.4861, p.1253–1257, 1962.

KIM, R. Y. The impact of COVID-19 on consumers: preparing for digital sales. **IEEE Engineering Management Review**, [S.l.], v.48, n.3, p.212–218, 2020.

LARCHER, A. et al. **ALIZE 3.0 - open source toolkit for state-of-the-art speaker recognition**. 2013.

LEE, S. A. S. Virtual speech-language therapy for individuals with communication disorders: current evidence, limitations, and benefits. **Current Developmental Disorders Reports**, [S.l.], v.6, n.3, p.119–125, 2019.

LUIS-GARCÍA, R. de et al. Biometric identification systems. **Signal processing**, [S.l.], v.83, n.12, p.2539–2557, 2003.

MORO, A. Aplicação mobile para triagem fonológica infantil. **Santa Maria: Universidade Federal de Santa Maria**, [S.l.], 2018.

PERLES, J. B. Comunicação: conceitos, fundamentos e história. **Biblioteca on-line de Ciências da Comunicação**, [S.l.], 2007.

POO, D.; KIONG, D.; ASHOK, S. **Object-oriented programming and Java**. [S.l.]: Springer Science & Business Media, 2007.

RIBARIC, S.; FRATRIC, I. A biometric identification system based on eigenpalm and eigenfinger features. **IEEE transactions on pattern analysis and machine intelligence**, [S.l.], v.27, n.11, p.1698–1709, 2005.

SAINI, R.; RANA, N. Comparison of various biometric methods. **International Journal of Advances in Science and Technology**, [S.l.], v.2, n.1, p.2, 2014.

SAVOLDI, A.; CERON, M. I.; KESKE-SOARES, M. Quais são as melhores palavras para compor um instrumento de avaliação fonológica? **Audiology-Communication Research**, [S.l.], v.18, n.3, p.194–202, 2013.

SCHMAEDECK, M. V. Sistema para construção de testes fonoaudiológicos voltados a triagem de desordens dos sons da fala em crianças. **Santa Maria: Universidade Federal de Santa Maria**, [S.l.], 2021.

SOUZA, I. V. e. Modelo para identificação de contexto social através da inferência de interações sociais. **Santa Maria: Universidade Federal de Santa Maria**, [S.l.], 2018.

SUMNER, S. **You**: for sale: protecting your personal data and privacy online. [S.l.]: Syngress, 2015.

VOICEIT. **Frequently Asked Questions**. Disponível em: https://voiceit.io/faq. Acessado em: jan. 2021.

WANG, Y.; WANG, Y.; TAN, T. Combining fingerprint and voiceprint biometrics for identity verification: an experimental comparison. In: INTERNATIONAL CONFERENCE ON BIOMETRIC AUTHENTICATION. **Anais...** [S.l.: s.n.], 2004. p.663–670.

WAYMAN, J. L. Fundamentals of biometric authentication technologies. **International Journal of Image and Graphics**, [S.l.], v.1, n.01, p.93–113, 2001.