UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
CÂMPUS CORNÉLIO PROCÓPIO
DIRETORIA DE GRADUAÇÃO E FORMAÇÃO PROFISSIONAL
DEPARTAMENTO DE COMPUTAÇÃO
BACHARELADO EM ENGENHARIA DE SOFTWARE

JEFFERSON DE FRANÇA FILHO

RETROALIMENTAÇÃO AUDITIVA ATRASADA: APLICATIVO DE AUXÍLIO AO TRATAMENTO DE PESSOAS COM GAGUEIRA

PROPOSTA DE TRABALHO DE CONCLUSÃO DO CURSO

CORNÉLIO PROCÓPIO
2018

JEFFERSON DE FRANÇA FILHO

RETROALIMENTAÇÃO AUDITIVA ATRASADA: APLICATIVO DE AUXÍLIO AO TRATAMENTO DE PESSOAS COM GAGUEIRA

Proposta de Trabalho de Conclusão do Curso apresentada ao Departamento de Computação da Universidade Tecnológica Federal do Paraná como requisito parcial para obtenção do título de “Bacharel em Engenharia de Software”.

Orientador: Prof. Dr. Fabrício Martins Lopes

CORNÉLIO PROCÓPIO
2018

FICHA CATALOGRÁFICA PREPARADA PELO AUTOR.

FRANÇA, Jefferson

F814 Retroalimentação Auditiva Atrasada: Aplicativo de Auxílio ao Tratamento de Pessoas com Gagueira/ Jefferson de França Filho - Cornélio Procópio: UTFPR, 2018.

33f. : il.

Inclui Bibliografia.

Proposta de Trabalho de Conclusão do Curso (Graduação em Engenharia de Software) - Universidade Tecnológica Federal do Paraná. Orientador: Prof. Dr. Fabrício Martins Lopes.

1. Retroalimentação Auditiva Atrasada. 2. Gagueira. 3. Aplicativo de Auxílio ao Tratamento de Pessoas com Gagueira. I. Título.

CDD-000

Minstério da Educação
Universidade Tecnológica Federal do Paraná
Câmpus Cornélio Procópio
Diretoria de Graduação e Formação Profissional
Departamento de Computação
Bacharelado em Engenharia de Software

TERMO DE APROVAÇÃO

Retroalimentação Auditiva Atrasada: Aplicativo de Auxílio ao Tratamento de Pessoas com Gagueira

por

Jefferson de França Filho

Esta Proposta de Trabalho de Conclusão do Curso foi julgada adequada para obtenção do Título de “Bacharel em Engenharia de Software” e aprovado em sua forma final pelo Departamento de Computação da Universidade Tecnológica Federal do Paraná.
Cornélio Procópio, 00/05/2018.

Érica Ferreira Souza, Doutora
Coordenadora do Curso

Banca Examinadora:

Fabrício Martins Lopes, Prof. Dr.
Orientador

Claiton de Oliveira, Prof. Dr.
UTFPR

Willian Watanabe, Prof. Dr.
UTFPR

“A Folha de Aprovação assinada encontra-se na Coordenação do Curso do Programa”

Dedico este trabalho à minha família, principalmente a minha mãe Jacira Aparecida Lopes por todo o suporte ao longo desta caminhada.

AGRADECIMENTOS

Agradeço ao meu orientador Prof. Dr. Fabrício Martins Lopes, pela sabedoria com que me guiou nesta trajetória.

As fonoaudiólogas Dr. Rosane Consalter e Dr. Cristiane M. C. de Oliveira, pelo suporte e dedicação com que me direcionaram neste caminho.

Gostaria de deixar registrado também, o meu reconhecimento à minha família, pois acredito que sem o apoio deles seria muito difícil vencer esse desafio.

Enfim, a todos os que por algum motivo contribuíram para a realização deste projeto.

Muitas palavras não indicam necessariamente muita sabedoria. (Tales de Mileto)

RESUMO

FRANÇA, Jefferson. **Retroalimentação Auditiva Atrasada: Aplicativo de Auxílio ao Tratamento de Pessoas com Gagueira**. 2018. 33 f. Proposta de Trabalho de Conclusão do Curso – Bacharelado em Engenharia de Software, Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2018.

A gagueira é uma doença caracterizada pela repetição de sons e sílabas, que não tem cura, mas que pode ser tratada por um fonoaudiólogo especializado na área. Existem tratamentos convencionais que trabalham com aspectos como a respiração, prolongamento das sílabas e articulação vocal, outros tratamentos, utilizam-se de aparelhos tecnológicos para melhorar a fluência da fala. A retroalimentação auditiva atrasada RAA é um método de tratamento da gagueira que simula o efeito coro, causado quando uma pessoa que gagueja ouve sua voz repetida, com um pequeno atraso e num tom diferente, causando melhorias significativas em relação às disfluências da fala. Além do SpeechEasy, um aparelho que combina hardware e software para simular o efeito coro, existem softwares que se utilizam de mecanismos de reprodução de áudio, como fones de ouvidos que possuam microfone, para proporcionar o RAA. O valor do SpeechEasy de aproximadamente dez mil reais limita sua aquisição para uma parcela pequena da população, para dispositivos móveis os softwares que trazem a mesma funcionalidade que o aparelho, em sua maioria não são gratuitos, ou possuem restrições em suas versões gratuitas. Este projeto tem como objetivo desenvolver um aplicativo para dispositivos móveis, que atenda a plataforma Android, tendo como funcionalidade principal proporcionar a simulação do efeito coro. Suprindo a necessidade da disponibilização de uma opção funcional e gratuita para o tratamento da gagueira utilizando a Retroalimentaąão Auditiva Atrasada.

Palavras-chave: Retroalimentação Auditiva Atrasada. Gagueira. Aplicativo de Auxílio ao Tratamento de Pessoas com Gagueira.

ABSTRACT

FRANÇA, Jefferson. **Delayed Auditory Feedback: App of Aid the Treatment of People with Stuttering.** 2018. 33 f. Dissertation – Bachelor Degree in Software Enginnering, Federal University of Technology - Paraná. Cornélio Procópio, 2018.

Stuttering is a disease characterized by the repetition of sounds and syllables, which have no cure but can be treated by a Speech Therapist specializing in the area. There are conventional treatments that work with aspects such as breathing, syllable prolongation and vocal articulation, other treatments, use of technological devices to improve speech fluency. Delayed auditory feedback DAF is a method of treating stuttering that simulates the chorus effect, caused when a person who stutters hears their voice repeatedly, with a short delay and in a different tone, causing significant improvements in relation to speech dysfluencies. In addition to SpeechEasy, a device that combines hardware and software to simulate the chorus effect, there are software that uses audio playback mechanisms, such as earphones that have a microphone, to provide the RAA. SpeechEasy's value of approximately ten thousand reais limits its acquisition to a small portion of the population, for mobile devices softwares that bring the same functionality as the device are mostly not free or restricted in their free versions. This project aims to develop an application for mobile devices, which meets the Android platform, with the main functionality to provide the simulation of the chorus effect. Addressing the need to provide a functional and free option for stuttering treatment using Delayed Auditory Feedback.

Keywords: Delayed Auditory Feedback. Stuttering. App of Aid the Treatment of People with Stuttering.

LISTA DE FIGURAS

LISTA DE TABELAS

SUMÁRIO

1 INTRODUÇÃO

Afetando cerca de 1% da população mundial e codificada na Classificação Internacional de Doenças (CID-10) com os caracteres F98.5, a gagueira é cientificamente considerada como distúrbio ou transtorno de fluência da fala (MERLO, 2013). Ou seja, é um distúrbio neurológico e involuntário, caracterizado por interrupções ou prolongamentos, audíveis ou não de sons e sílabas (BUCHEL; SOMMER, 2004).

A retroalimentação auditiva atrasada (RAA) é um método de tratamento da gagueira, que utiliza-se de duas grandezas, a frequência e o atraso (*delay*), para proporcionar o efeito coro, causado quando uma pessoa que gagueja, fala ou lê ao mesmo tempo que outra pessoa, ou seja, faz com que a pessoa que gagueja ouça suas próprias palavras com um certo atraso tendo a sensação de que está falando junto com outros (UDEMO, 2008).

Um aparelho tecnológico que oferece o RAA como funcionalidade é o *SpeechEasy* da Microsom, que se assemelha muito em sua aparência, com um aparelho para deficientes auditivos. Segundo a Microsom, o *SpeechEasy* tem eficiência em 75% das pessoas que o utilizam e cerca de 80% dos clientes que adquiriram o produto, estão satisfeitos com o resultado (MICROSON, 2015). Uma pesquisa realizada com 31 participantes que possuem gagueira, registrou resultados parecidos, apresentando melhorias de cerca de 79% na leitura e 61% na fala auto-expressiva dos participantes com a utilização do aparelho (ANDRADE et al., 2008).

Existem aplicativos que exercem a funcionalidade de simular o efeito coro, tanto para dispositivos móveis como para computadores de mesa e notebooks. Para realizar a simulação do efeito coro de maneira adequada é recomendado utilizar um aparelho de reprodução que contenha microfone, podendo ser um fone de ouvido convencional ou um *headset* de sua preferência.

1.1 PROBLEMA

Custando aproximadamente 10 mil reais e podendo ser adquirido somente sob consulta com uma fonoaudióloga especializada, o *SpeechEasy* acaba se tornando uma opção restrita para pessoas com poucas condições financeiras. Segundo o Instituto Brasileiro de Fluência (IBF) a Microson está em contato com o Ministério de Saúde para que o aparelho seja disponibilizado pelo SUS, porém enquanto isso não ocorre, sua disponibilidade é limitada para quem tem condições de investir cerca de 10 salários mínimos neste produto.

Ao contrário da plataforma *Window*s onde existe a ferramenta "Mais Fluência" que oferece a funcionalidade de simular o efeito coro, gratuitamente e sem limitações. Para plataforma Android encontra-se diversos aplicativos que sequer conseguem atender essa funcionalidade e

quando atendem existem limitações em suas versões gratuitas.

Existe uma grande dificuldade em encontrar uma ferramenta para dispositivos móveis que realmente atenda a funcionalidade de simular o efeito coro, que seja fornecida gratuitamente sem restricões de utilização.

1.2 JUSTIFICATIVA

O problema no Brasil é que o SpeechEasy não pode ser adquirido pela maioria das pessoas que necessitam, devido ao seu valor. Existem outras soluções como aplicativos mobile que tentam fazer o mesmo papel porém utilizando um fone de ouvido *bluetooth* ou qualquer outro tipo de dispositivo de reprodução.

É muito difícil encontrar aplicativos que disponibilizam essa funcionalidade de maneira eficiente e gratuita, outros aplicativos que trazem a função de simular o efeito coro de forma simples e bem superficial não são gratuitos, o que torna difícil obter resultados satisfatórios.

Realizar apresentações, seminários ou quaisquer atividades que necessitam de atividade vocal na universidade, para pessoas com gagueira é uma tarefa bem difícil, pois além da dificuldade de demonstrar conhecimento sobre o assunto exposto, existe também a dificuldade para se expressar de maneira fluente.

O problema na universidade é que não existem mecanismos que auxiliam esses alunos a lidarem com essas situações, o que acaba muitas vezes fazendo com que a pessoa que tem gagueira desista de realizar determinadas atividades pela dificuldade de comunicação.

1.3 OBJETIVOS

1.3.1 Objetivo Geral

Desenvolver um aplicativo para dispositivos móveis com sistema operacional Android, gratuito e funcional, que atenda o requisito principal de simular o efeito coro.

1.3.2 Objetivos Específicos

- Inclusão social de pessoas com gagueira em atividades que exijam comunicação oral, com a utilização do aplicativo.
- Acessibilidade para pessoas com poucas condições financeiras para adquirir o *SpeechEasy*, tornando o aplicativo uma alternativa gratuita que exerce o mesmo papel.
- Disponibilizar uma opção gratuita para fonoaudiólogos e profissionais da área de auxílio ao tratamento de pessoas com gagueira, utilizando o RAA.

1.4 ORGANIZAÇÃO DO TEXTO

O documento está organizado em capítulos, divididos em:

- Capítulo 2: apresenta a fundamentação teórica dando ênfase nos trabalhos relacionados, mostrando aplicativos que tenham similaridades com a ferramenta desenvolvida no presente trabalho.
- Capitulo 3: apresenta a proposta, citando as tecnologias e ferramentas utilizadas, especificando qual o método seguido para o desenvolvimento, a análise e desenvolvimento, onde apresenta-se os requisitos do sistema, os diagramas e protótipos de tela, e por fim o cronograma a ser seguido.
- Referências: apresenta as referências utilizadas.

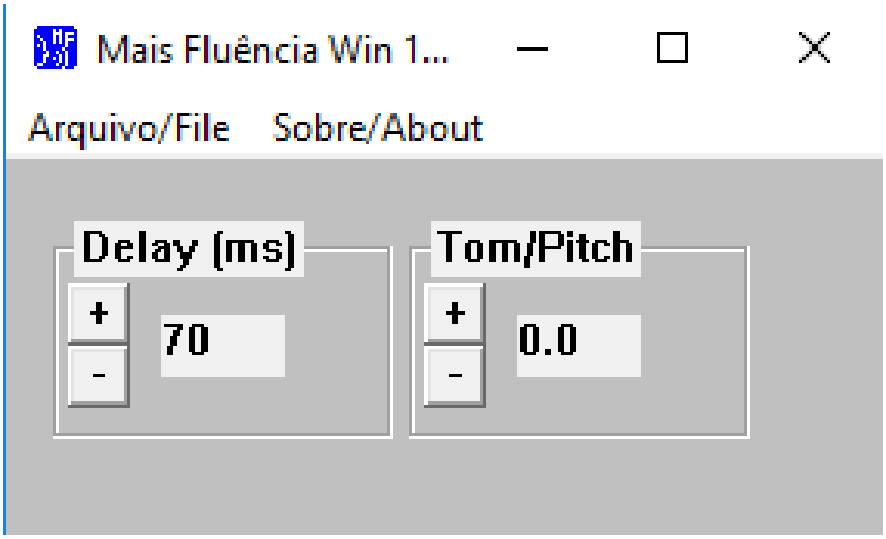
2 FUNDAMENTAÇÃO TEÓRICA

Com o objetivo de simular o efeito coro, além do *SpeechEasy* que integra *hardware* e *software* num dispositivo eletrônico personalizado, oferecendo opções de tamanho, e diferenciais como adaptação instantânea e menor efeito de oclusão (MICROSON, 2015). Existem algumas ferramentas que trabalham somente com *software* e que exercem essa função juntamente com algum dispositivo de reprodução de aúdio que contenha microfone. A figura 1 apresenta alguns dos modelos de tamanho do *SpeechEasy*.

Fonte: (MICROSON, 2015)

Figura 1 – Opções de tamanho do SpeechEasy.

Para computadores de mesa e notebooks com sistema operacional Windows, existe o "*Software* Mais Fluência Win DAF/FAF *Software*", desenvolvido em 2009 pelo Henrique Confessor, é *freeware* podendo ser distribuído e utilizado livremente, disponibilizado gratuitamente para *download* no site da "Abra Gagueira"(CONFESSOR, 2009). Sua interface simples, permite somente duas configurações, atraso e frequência, apresenta apenas dois botões auxiliares que tem as funções de fechar e exibir uma tela com as informações sobre o *software*, como: versão, e-mail para contato e o link para o blog do autor. É o *Software* mais funcional relatado neste documento, atende somente a funcionalidade de simular o efeito coro, não traz nenhum diferencial. A figura 2 apresenta a interface do "*Software* Mais Fluência Win DAF/FAF *Software*".

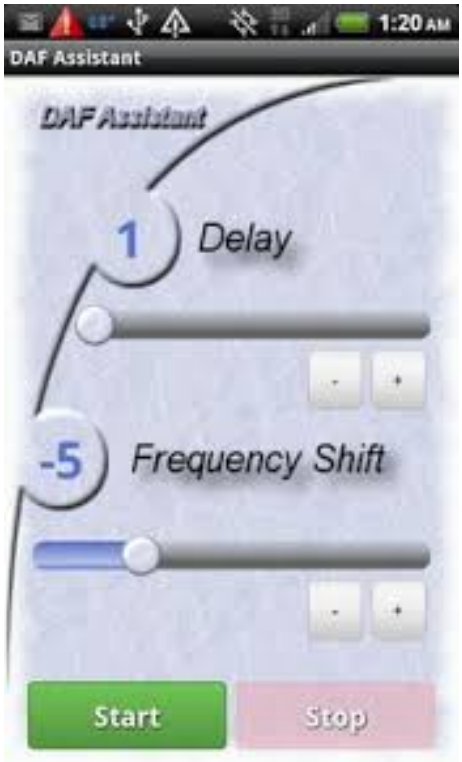


Fonte: O Autor.

Figura 2 – Interface do software Mais Fluência Win DAF/FAF.

Para dispositivos móveis com sistema operacional Android ou IOS existe o *DAF Assistant* (*Delayed Auditory Feedback Assistant*), que tem uma versão gratuita, porém com limite de tempo para sua utilização, já sua versão paga que não possui essa restrição, custa aproximadamente 13 reais na *Play Store* e 33 reais no *Itunes*, variando de acordo com preço do dollar (LCC, 2012).

O *DAF Assistant* possui uma interface intuitiva, contendo as opções de configurar atraso e frequência, traz apenas dois botões reponsáveis por iniciar ou parar a reprodução do efeito coro. Existem opções que não estão visíveis na tela, ficando acessíveis somente quando pressionado o botão de configuração do celular (dependendo de cada dispositivo), exibindo a opção de fechar o aplicativo, ou ir para uma tela de preferencias, onde é possível ativar a utilização de um *headset bluetooth*, além do *auto mute*, *mute after* e configurar a sensibilidade da fala em baixa, normal ou alta. A figura 3 apresenta a interface do aplicativo *DAF Assistant*.

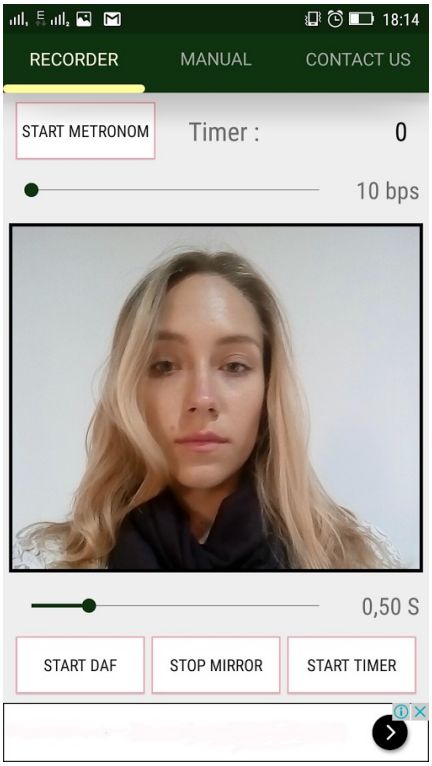


Fonte: (LCC, 2012)

Figura 3 – Interface do aplicativo DAF Assistant.

Com o intuito de fornecer o *feedback* auditivo atrasado (FAA), também para a plataforma Android existe o aplicativo "Terapia para a gagueira - FAA", que é gratuito e traz informações interessantes sobre o tratamento da gagueira, como dicas de como utilizar o aplicativo e informações adicionais sobre tratamentos que melhoram a fluência da fala. Lembrando que diferente da retroalimentação auditiva atrasada (RAA), o FAA trabalha apenas com o atraso na reprodução da voz, não alterando a frequência com que a voz é reproduzida (AGE, 2017).

A interface do aplicativo Terapia para a gagueira - FAA contém muita informação, possuindo muitos botões, isso se da pelo fato de oferecer diversas funcionalidades além do FAA, como: oferece um espelho utilizando a câmera frontal do dispositivo, opção de gravar o áudio enquanto faz a utilização do aplicativo, disponibiliza um metrônomo para controle do rítimo da fala e um vídeo contendo informações sobre o tratamento da gagueira. Apesar de gratuito comtém muitas propagandas, oque pode ocasionar incômodo em alguns usuários. A figura 4 apresenta a interface do aplicativo Terapia para Gagueira - FAA.



Fonte: (AGE, 2017)

Figura 4 – Interface do aplicativo Terapia para a gagueira - FAA.

3 PROPOSTA

A proposta desse projeto é desenvolver um aplicativo para dispositivos móveis com sistema operacional Android, tendo como função principal o retroalimentação auditiva atrasada (RAA), ou seja, um aplicativo que consiga reproduzir a voz do usuário simultaneamente com um pequeno atraso configurável, num tom diferente também configurável.

O atraso é medido em milissegundos, sendo possível utilizar o intervalo entre 250 e 3000 milissegundos (de 0,25 a 3 segundos). A frequência é medida em Hertz, possuindo as opções de intervalos entre 500 e 2.000 Hertz. Esses dados foram obtidos através de uma pesquisa, onde participaram 20 indivíduos com gagueira de 7 a 17 anos, resultando em uma diminuição estatisticamente significante na ocorrência de bloqueios e repetições de palavras em indivíduos com gagueira sem alteração do processamento auditivo central, ou seja, sem alteração na capacidade que o sistema nervoso tem para traduzir as informações enviadas pela audição (PICOLOTO et al., 2017).

A finalidade dessas configurações que devem ser adaptadas para cada indivíduo é simular o efeito coro, que nada mais é do que um efeito causado quando uma pessoa que possui gagueira, fala ou lê ao mesmo tempo que outra, trazendo melhorias significativas na fala (UDEMO, 2008).

3.1 TECNOLOGIAS E FERRAMENTAS

- Java: utiliza-se como linguagem de programação.
- Android Studio: utiliza-se como ambiente de desenvolvimento (Ambiente de Desenvolvimento Integrado (IDE)).
- Github: utiliza-se como repositório de armazenamento e controle de versões.
- Google Drive: utiliza-se como gerenciador de arquivos de texto, e planilhas.
- UML: utiliza-se como linguagem-padrão para a elaboração da estrutura de projetos de *software*.
- Astah: utiliza-se como ferramenta de modelagem Linguagem Unificada de Modelagem (UML).

3.2 MÉTODO

Uma alternativa para atender clientes e projetos de forma dinâmica, flexível e com produtividade elevada é a metodologia *Agile*, ou ágil em português, que tem se consolidado

ao longo dos últimos anos com a utilização de uma abordagem de planejamento iterativa. O *Scrum* é um *framework* muito utilizado entre as metodologias ágeis, especialmente pelo formato dinâmico como as etapas dos projetos são desenvolvidas (UDACITY, 2017).

Para o desenvolvimento do aplicativo descrito neste documento, utiliza-se uma metodologia incremental adaptada e baseada no *Scrum*, seguindo alguns de seus conceitos mais importantes, como:

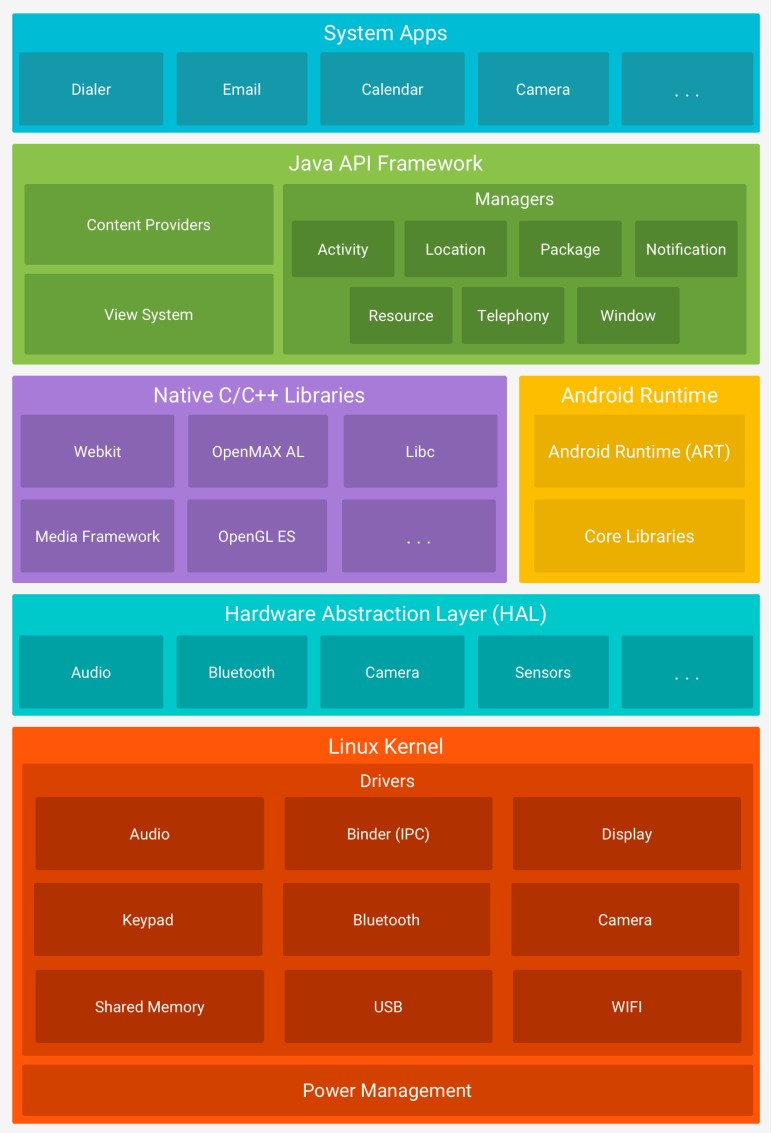
- *Sprint*: são iterações, ciclos de desenvolvimento que começam numa reunião de planejamento (*Sprint Planning*) e terminam com a revisão (*Sprint Review*) e a retrospectiva (*Sprint Retrospective*).
- *Product Owner*: é o responsável por definir prioridades a serem desenvolvidas em cada *sprint* e fazer a intermediação entre equipe de negócios e equipe de *scrum*.
- *Scrum Master*: responsável por resolver impedimentos que possam prejudicar a equipe *scrum*, e assegurar que todos sigam a metodologia proposta.
- *Sprint Planning*: reunião para planejar quais itens do *backlog* do produto serão priorizados em determinada *sprint*, que abrange determinado período(1 até 4 semanas).
- *Sprint Meeting Review*: Reunião de revisão da *sprint*, discutindo tudo que foi desenvolvido naquele ciclo.
- *Sprint Retrospective*: realizada após a reunião de revisão e antes da reunião de planejamento, visa estabelecer possíveis melhorias.

3.3 ANÁLISE E DESENVOLVIMENTO

3.3.1 Arquitetura

Utiliza-se como arquitetura de desenvolvimento, a arquitetura de camadas do software Android, que é executado sobre um *kernel* Linux. Os aplicativos Android são gravados na linguagem de programação Java e são executados em uma máquina virtual (VM) (ABLESON, 2009).

A figura 5 situada logo abaixo apresenta a maioria dos componentes da plataforma Android.



Fonte: (DEVELOPERS, 2018)

Figura 5 – Camadas do Software Android.

3.3.2 Requisitos

Nesta seção apresenta-se os requisitos do sistema, divididos em:

- Requisitos Funcionais (RF): apresentam as funcionalidades do sistema, ou seja, define oque o sistema fará.
- Requisitos Não-Funcionais (RNF): apresentam os atributos de qualidade para o sistema, ou seja, como o sistema fará determinada atividade, podendo ser categorizados em: usabilidade, desempenho, padrão, etc (VENTURA, 2016b).

A prioridade dos requisitos pode ser classificada em:

- Essencial: deve ser implementado para que o sistema funcione.
- Importante: sem este requisito o sistema pode funcionar, mas não da maneira esperada.
- Desejável: este tipo de requisito não compromete o funcionamento do sistema.

A Tabela 1 apresenta os requisitos funcionais do sistema, contendo sua descricão, prioridade e os requisitos relacionados.

Id	**Descrição**	**Prioridade**	**Requisitos Relacionados**
RF01	O aplicativo deve permitir ao usuário editar as preferências de frequência e atraso.	Essencial	RF02 - RF04
RF02	O aplicativo deve permitir ao usuário iniciar e interromper a simulação do efeito coro.	Essencial	N/A
RF03	O aplicativo deve permitir a utilização de fone de ouvido *bluetooth*.	Importante	RF01 - RF002
RF04	O aplicativo deve conter uma tela "Sobre", contendo informações sobre a utilização do aplicativo.	Desejável	N/A
RF05	O aplicativo deve permitir criar e selecionar modos personalizados, como a criação de: modo casa, modo apresentação, modo tutorial, entre outros. Onde cada modo possui preferências pré-definidas.	Desejável	RF01-RF03
RF06	O aplicativo deve conter uma tela "Modos", onde é possível visualizar todos os modos cadastrados.	Desejável	RF05
RF07	O aplicativo deve permitir excluir um modo.	Desejável	RF05-RF06
RF08	O aplicativo deve dar suporte para os idiomas inglês e português. Traduzindo todos os elementos de interface de acordo com a linguagem do dispositivo.	Desejável	N/A

Fonte: O Autor.

Tabela 1 – **Requisitos Funcionais**

A tabela 2 apresenta os requisitos n/ ao funcionais do sistema, de acordo com sua prioridade.

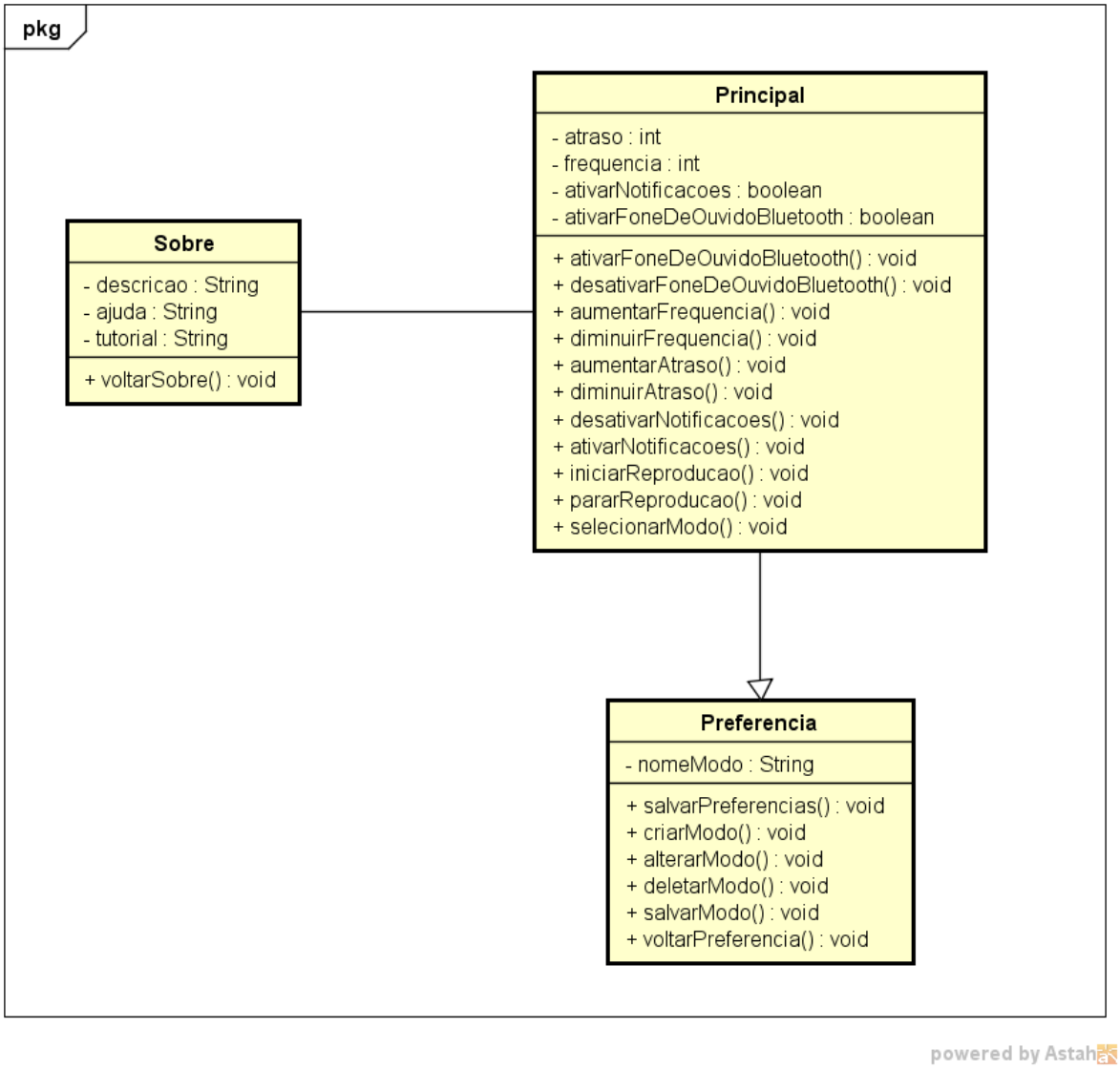
Id	Descrição	Categoria	Prioridade	Requisitos Relacionados
RNF01	O aplicativo deve ser desenvolvido para a plataforma Android.	Compatibilidade	Essencial	RFN03
RNF02	O usuário do aplicativo deve ser capaz de usufruir das suas funcionalidades com no máximo 1 minuto de utilização.	Usabilidade	Importante	RNF04
RNF03	O aplicativo deve ser implementado na linguagem de programação JAVA.	Implementação	Importante	RNF01
RNF04	A interface do aplicativo deve ser simples, com no máximo 5 botões, ou controladores (Aumentar ou diminuir a frequência e o *delay*, botão iniciar/-desligar, e opção de configurações).	Usabilidade	Desejável	RNF02

Fonte: O Autor.

Tabela 2 – **Requisitos Não-Funcionais**

3.3.3 Diagrama de Classes

Apresenta-se o diagrama de classes, uma representação da estrutura e relações das classes que servem de modelo para objetos (TYBEL, 2017). Visualizar figura 6.

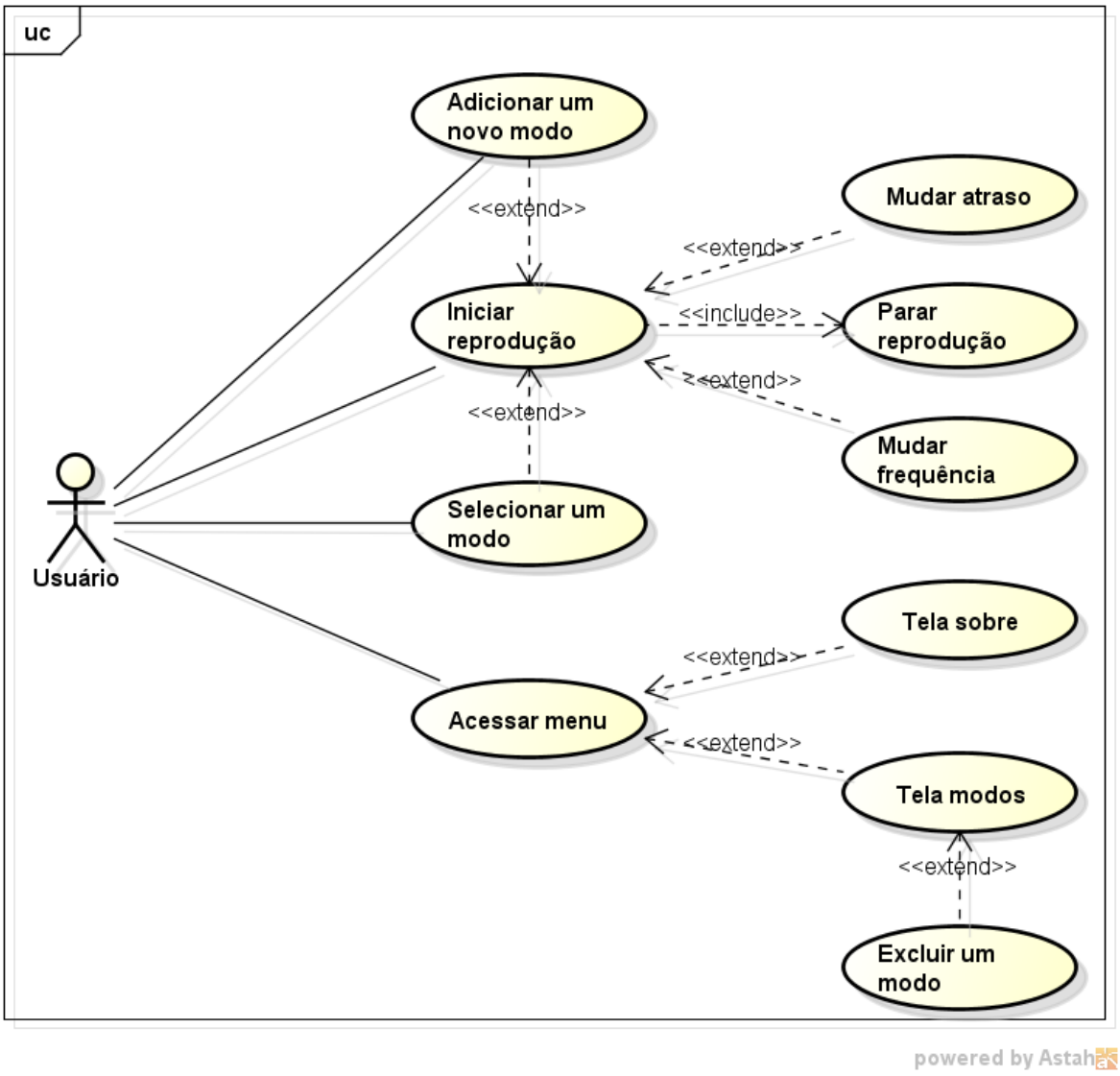


Fonte: O Autor.

Figura 6 – Diagrama de Classes.

3.3.4 Diagrama de Casos de Uso

Apresenta-se o diagrama de casos de uso, documentando o que o sistema faz do ponto de vista do usuário, ou seja, descreve as principais funcionalidades do sistema e a interação dessas funcionalidades com os usuários do mesmo sistema (RIBEIRO, 2012). Visualizar figura 7.

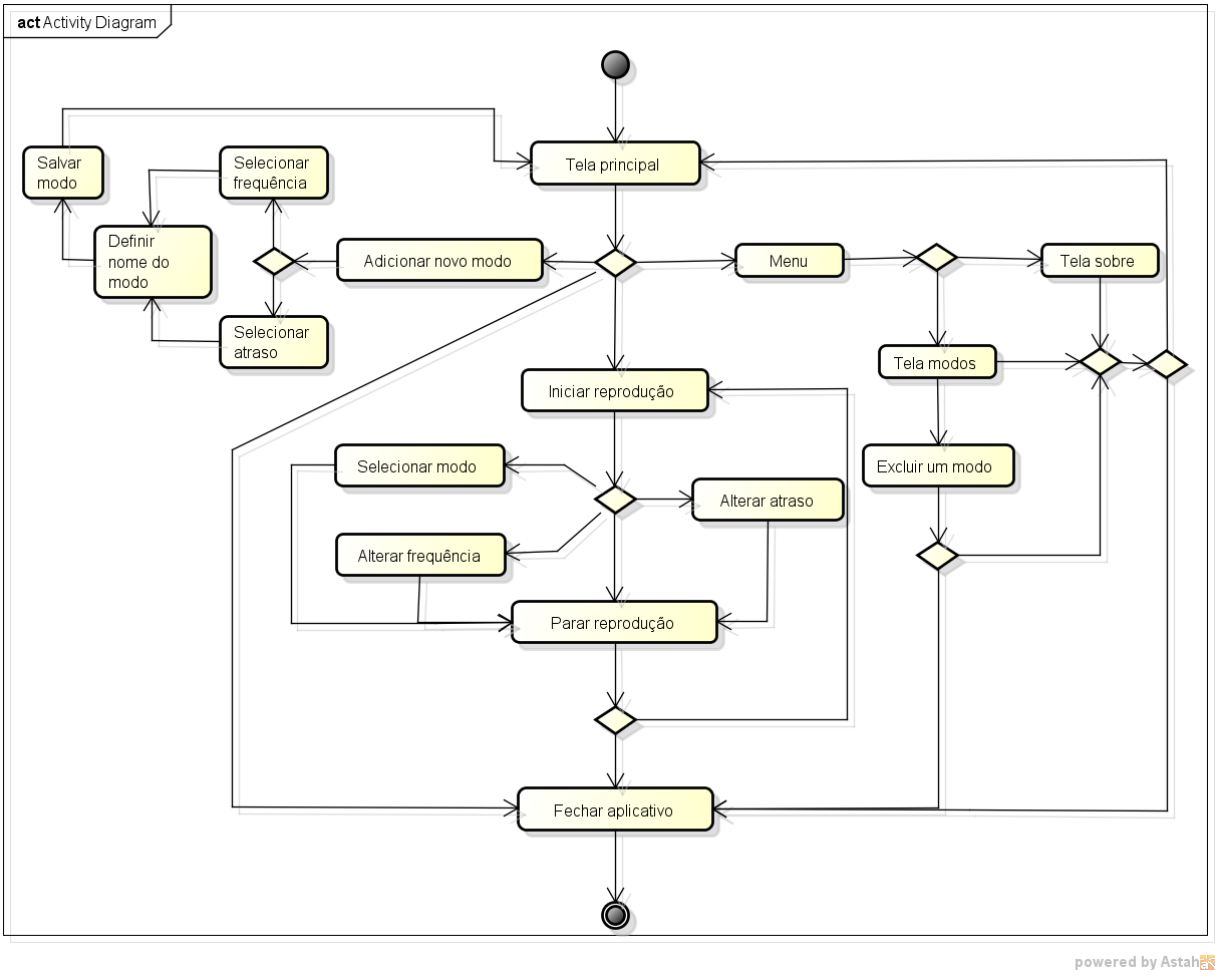


Fonte: O Autor.

Figura 7 – Diagrama de Casos de Uso.

3.3.5 Diagrama de Atividades

Apresenta-se o diagrama de atividades, com o objetivo de mostrar o fluxo de atividades em um único processo, especificando o comportamento do software do ponto de vista funcional (VENTURA, 2016a). Visualizar figura 8.



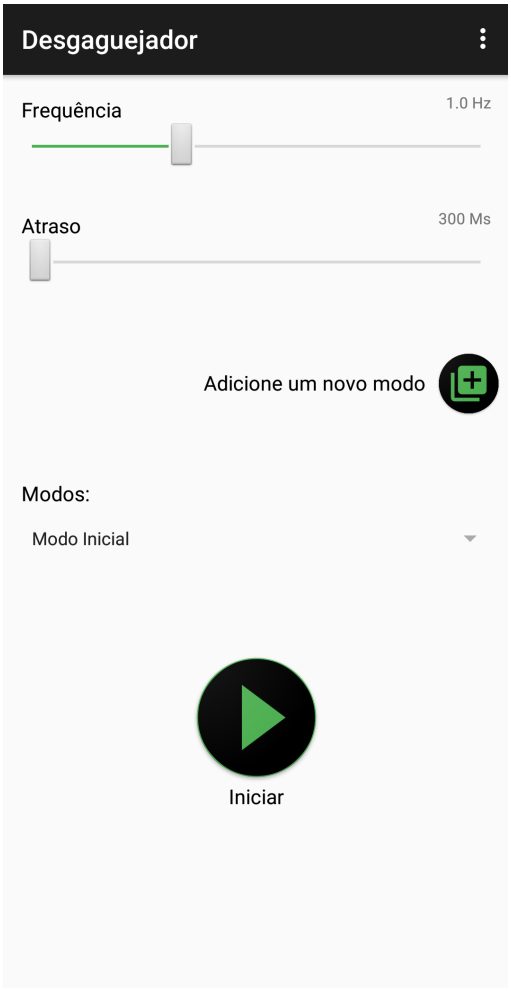
Fonte: O Autor.

Figura 8 – Diagrama de Atividades.

3.4 DESENVOLVIMENTO DE TELAS

Apresenta-se as telas do sistema.

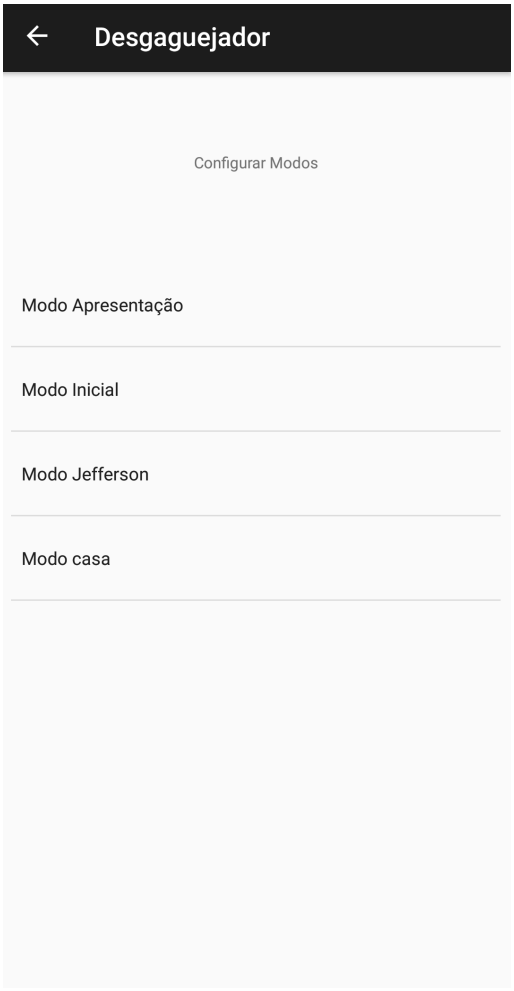
- Tela Inicial: Nesta tela o usuário tem acesso a todas as funcionalidades do sistema, além de iniciar a simulação do efeito coro, ele pode ajustar o atraso e a frequência de acordo com suas preferências, também tem as opções de selecionar e adicionar um novo modo. Desta tela também existe a opção de navegar entre as telas "Sobre"e "Modos", selecionando a opção referente a cada tela no menu localizados no canto superior direito. Visualizar figura 9.



Fonte: O Autor.

Figura 9 – Tela Inicial.

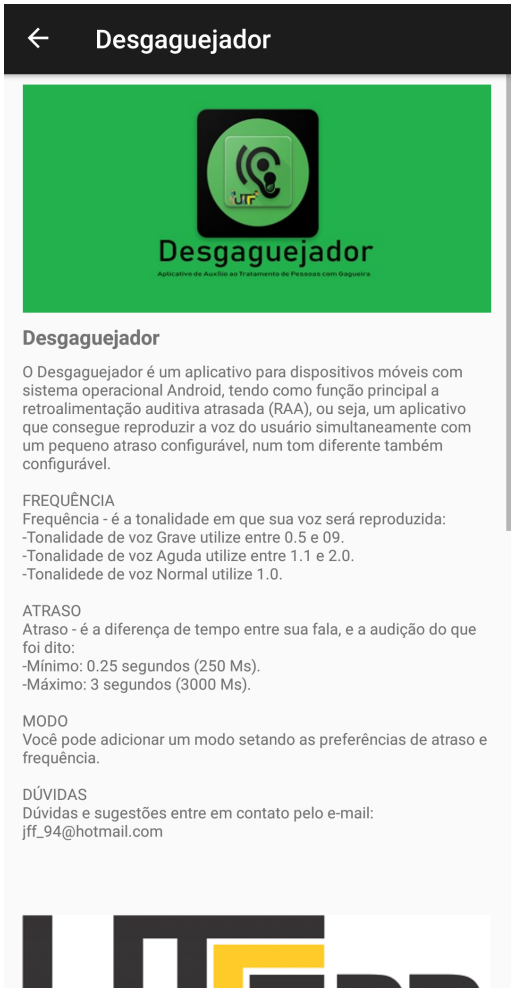
- Tela Modos: Nesta tela o usuário pode visualizar todos os modos cadastrados, além da possibilidade de excluir um modo, para isso basta clicar e segurar por alguns segundos em um modo, assim a opção "Excluir"será exibida. Visualizar figura 10.



Fonte: O Autor.

Figura 10 – Tela Modos.

- Tela Sobre: Nesta tela o usuário encontra informações sobre o desenvolvimento do aplicativo, assim como informações sobre o funcionamento do aplicativo. Visualizar figura 11.



Fonte: O Autor.

Figura 11 – Tela Sobre.

3.5 CÓDIGO FONTE

O código fonte do aplicativo esá versionado e disponível no repositório do github: <https://github.com/JaTemJeff/daf-app-utfpr-cp-2018>, encontra-se em sua versão de build 1.3, ultima atualização em 04/11/2018.

O principal método responsável pela simulação do efeito coro, se encontra na classe "MainActvity"denominado *"start()"*. Composto por outros métodos e variáveis, que foram implementados utilizando a biblioteca *"RxAndroidAudio"* disponibilizada pelo fornecedor *"Piasy"*, disponível em: <https://github.com/Piasy/RxAndroidAudio>. Abaixo são descritos as responsabilidades dos principais métodos que compõe o método *"start()"*:

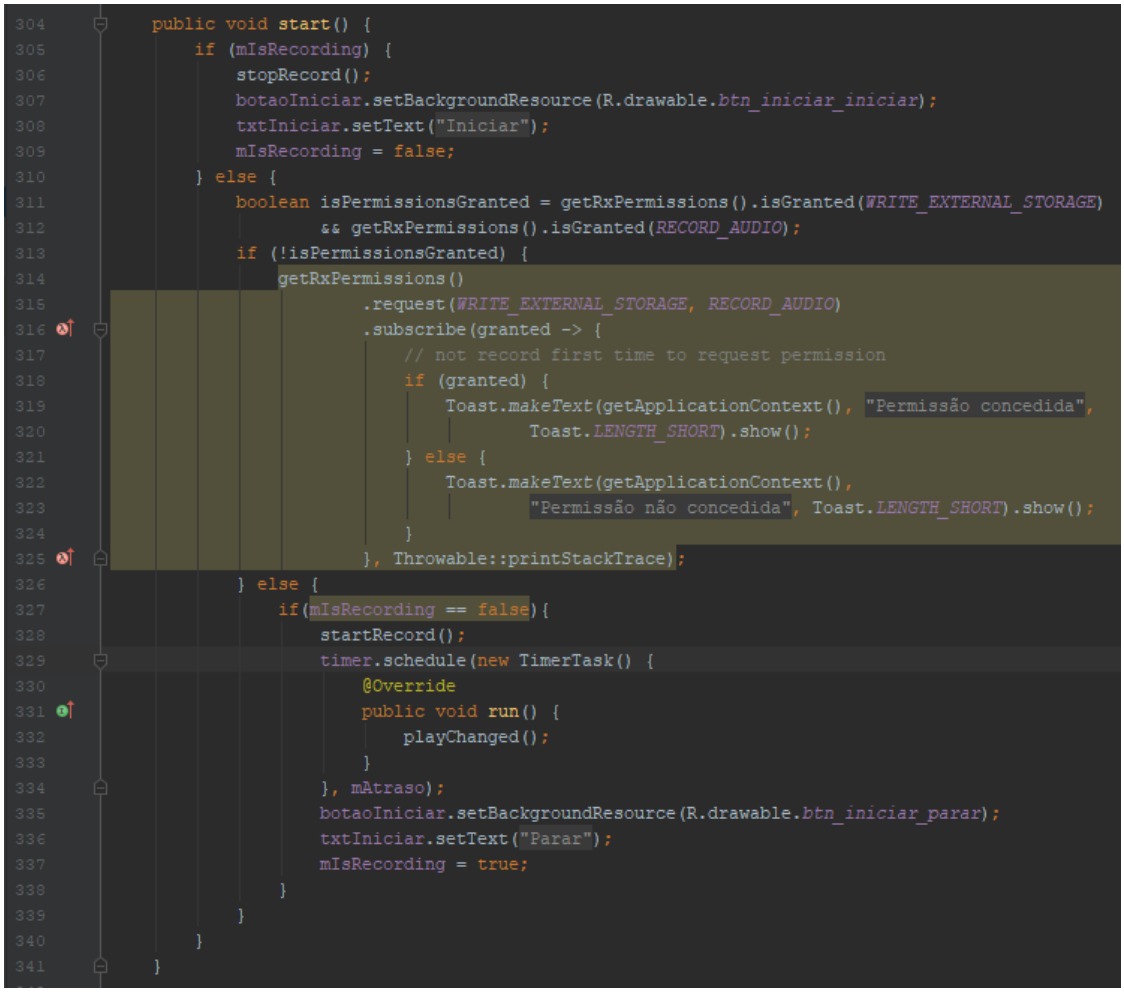
- Método *"stopRecord()"* - responável por intenrromper a gravação de áudio.
- Método *"startRecord()"* - responsável por iniciar a gravação de áudio.

- Método *"playChanged()"* - responsável por aplicar a frequência na faixa de áudio reproduzida.

A Figura 12 apresenta o método *"start()"*. Onde as linhas 305 até 309 são reponsáveis por verificar se o áudio está sendo gravado, caso esteja, é chamado o metodo *"stopRecord()"*, interrompendo a gravação, alterando a imagem de *background* do botão iniciar e o texto abaixo dele, além de setar como *"false"* a variável *"mIsRecording"* utilizada para a verificação se o áudio está sendo gravado ou não.

As linhas 310 até 325 são responsáveis por garantir as permissões de utilizaccão do aplicativo, que são as de microfone e acesso ao armazenamento. Assim, caso as permissões ainda não estejam concedidas, nas linhas 314 até 325 utiliza-se a biblioteca *"RxPermissions"* disponibilizada pelo fornecedor *"tbruyelle"*, disponível em: <https://github.com/tbruyelle/RxPermissions>, para conceder essas permissões, bastando o usuário aceitar essa ação.

As linhas 327 até 337, destinam-se a gravar e reproduzir o áudio setando as preferências de atraso e frequência do usuário. Caso a variável *"mIsRecording"* tenha o valor como *false*, utiliza-se o método *"startRecord()"* para iniciar a gravação. Para reproduzir o áudio de acordo com a frequência utiliza-se o método *"playChanged()"*, onde dentro deste método existe uma variável chamada de *"mRatio"*, responsável por armazenar o valor do *seekbar* "Frequência", ficando assim a definição de acordo com oque for setado pelo usuário. Para setar o atraso utiliza se o método *"schedule()"* da classe *timer*, passando como parâmetro a variável *mAtraso* responsável por armazenar o valor do *seekbar* "Atraso".



```
304     public void start() {
305         if (mIsRecording) {
306             stopRecord();
307             botaoIniciar.setBackgroundResource(R.drawable.btn_iniciar_iniciar);
308             txtIniciar.setText("Iniciar");
309             mIsRecording = false;
310         } else {
311             boolean isPermissionsGranted = getRxPermissions().isGranted(WRITE_EXTERNAL_STORAGE)
312                     && getRxPermissions().isGranted(RECORD_AUDIO);
313             if (!isPermissionsGranted) {
314                 getRxPermissions()
315                         .request(WRITE_EXTERNAL_STORAGE, RECORD_AUDIO)
316                         .subscribe(granted -> {
317                             // not record first time to request permission
318                             if (granted) {
319                                 Toast.makeText(getApplicationContext(), "Permissão concedida",
320                                         Toast.LENGTH_SHORT).show();
321                             } else {
322                                 Toast.makeText(getApplicationContext(),
323                                         "Permissão não concedida", Toast.LENGTH_SHORT).show();
324                             }
325                         }, Throwable::printStackTrace);
326             } else {
327                 if(mIsRecording == false){
328                     startRecord();
329                     timer.schedule(new TimerTask() {
330                         @Override
331                         public void run() {
332                             playChanged();
333                         }
334                     }, mAtraso);
335                     botaoIniciar.setBackgroundResource(R.drawable.btn_iniciar_parar);
336                     txtIniciar.setText("Parar");
337                     mIsRecording = true;
338                 }
339             }
340         }
341     }
```

Fonte: O Autor.

Figura 12 – Método *"start()"*.

REFERÊNCIAS

ABLESON, Frank. Introdução ao desenvolvimento do android. **IBM Developer Works**, 2009. Disponível em: <https://www.ibm.com/developerworks/br/library/os-android-devel/index.html>. Acesso em: 07 mai. 2018. Citado na página 20.

AGE, Information. Terapia para a gagueira - faa. **Google Play**, 2017. Disponível em: <https://play.google.com/store/apps/details?id=delayed.auditory.feedback.stuttering.therapy.daf>. Acesso em: 08 mai. 2018. Citado 2 vezes nas páginas 17 e 18.

ANDRADE, Claudia Regina Furquim de et al. The effect of speecheasy on stuttering frequency, speech rate and speech naturalness. **Revista da Sociedade Brasileira de Fonoaudiologia**, scielo, v. 13, p. 411 – 412, 00 2008. ISSN 1516-8034. Disponível em: <http://www.scielo.br/scielo.php?script=sci_arttext&pid=S1516-80342008000400018&nrm=iso>. Acesso em: 01 mai. 2018. Citado na página 12.

BUCHEL, Christian; SOMMER, Martin. What causes stuttering? **PLOS Biology**, Public Library of Science, v. 2, n. 2, 02 2004. Disponível em: <https://doi.org/10.1371/journal.pbio.0020046>. Acesso em: 01 mai. 2018. Citado na página 12.

CONFESSOR, Henrique. Software mais fluência win daf/faf software. **Abra Gagueira**, 2009. Disponível em: <http://www.abragagueira.org.br/mais_fluencia.asp>. Acesso em: 04 mai. 2018. Citado na página 15.

DEVELOPERS, Android. Arquitetura da plataforma android. **Android Developers**, 2018. Disponível em: <https://developer.android.com/guide/platform/?hl=pt-br>. Acesso em: 07 mai. 2018. Citado na página 21.

LCC, Artefact. Daf assistant. **Google Play**, 2012. Disponível em: <https://play.google.com/store/apps/details?id=com.artefactsoft.daf&hl=pt>. Acesso em: 04 mai. 2018. Citado 2 vezes nas páginas 16 e 17.

MERLO, Sandra. Caracterização da gagueira. **Instituto Brasileiro de Fluência - IBF**, 2013. Disponível em: <http://www.gagueira.org.br/conteudo.asp?id_conteudo=29>. Acesso em: 01 mai. 2018. Citado na página 12.

MICROSON. Saiba mais sobre a gagueira. **Como funciona o SpeechEasy?**, 2015. Disponível em: <http://www.microsom.com.br/saiba-mais-sobre-gagueira/saiba-mais-sobre-gagueira-como-funciona/>. Acesso em: 02 mai. 2018. Citado 2 vezes nas páginas 12 e 15.

PICOLOTO, Luana Altran et al. Efeito da retroalimentação auditiva atrasadana gagueira com e sem alteração doprocessamento auditivo central. **CoDAS**, scielo, v. 29, 00 2017. ISSN 2317-1782. Disponível em: <http://www.scielo.br/scielo.php?script=sci_arttext&pid=S2317-17822017000600307&nrm=iso>. Acesso em: 01 mai. 2018. Citado na página 19.

RIBEIRO, Leandro. O que é uml e diagramas de caso de uso: Introdução prática à uml. **Devmedia**, 2012. Disponível em: <https://www.devmedia.com.br/o-que-e-uml-e-diagramas-de-caso-de-uso-introducao-pratica-a-uml/23408>. Acesso em: 04 mai. 2018. Citado na página 24.

TYBEL, Douglas. Orientações básicas na elaboração de um diagrama de classes. **Devmedia**, 2017. Disponível em: <https://www.devmedia.com.br/orientacoes-basicas-na-elaboracao-de-um-diagrama-de-classes/37224>. Acesso em: 06 mai. 2018. Citado na página 23.

UDACITY. Metodologia scrum e agile. oque são e como aplicá-las? **Web Mobile Marketing Digital**, 2017. Disponível em: <https://br.udacity.com/blog/post/metodologia-scrum-agile>. Acesso em: 05 mai. 2018. Citado na página 20.

UDEMO. Efeito coro. **Folha de São Paulo**, 2008. Disponível em: <http://www.udemo.org.br/Leituras/Leituras_161.htm>. Acesso em: 02 mai. 2018. Citado 2 vezes nas páginas 12 e 19.

VENTURA, Pínio. Entendendo o diagrama de atividades da uml. **Até o Momento**, 2016. Disponível em: <http://www.ateomomento.com.br/uml-diagrama-de-atividades/>. Acesso em: 04 mai. 2018. Citado na página 25.

VENTURA, Plínio. O que é um requisito não-funcional. **Até o momento.**, 2016. Disponível em: <http://www.ateomomento.com.br/o-que-e-um-requisito-nao-funcional/>. Acesso em: 05 mai. 2018. Citado na página 21.