

**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**

DISSERTAÇÃO

CORNÉLIO PROCÓPIO

2018

JEFFERSON DE FRANÇA FILHO

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

Dissertação 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

Sobrenome, Nome
S661a A análise sensorial como ferramenta para otimização do
processamento de alimentos. Nome e Sobrenome. – São
Paulo: [s.n.], 2009.
54f.: il.

Monografia apresentada à “Nome da Instituição de Ensino
Superior” como parte dos requisitos exigidos para a conclusão
Do Curso de ...

Orientador: Prof. Nome e Sobrenome

1. Processamento de alimentos. 2. Controle da produção.
I. Título. II. Sobrenome, Nome. (Orientador)

641.3 CDD



TERMO DE APROVAÇÃO

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

por

Jefferson de França Filho

Esta Dissertação 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.

Banca Examinadora:

Nome do coordenador, Grau
Coordenadora do Curso

Fabício Martins Lopes, Prof. Dr.
Orientador

Primeiro Membro da Banca, Título
Universidade

Segundo Membro da Banca, Título
Universidade

Terceiro Membro da Banca, Título
Universidade

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, Filho. **Retroalimentação Auditiva Atrasada: Aplicativo de Auxílio ao Tratamento de Pessoas com Gagueira**. 2018. 32 f. Dissertação – Bacharelado em Engenharia de Software, Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2018.

Texto do resumo (máximo de 500 palavras).

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

ABSTRACT

SOBRENOME, Nome. **Delayed Auditory Feedback: App of Aid the Treatment of People with Stuttering.** 2018. 32 f. Master Thesis – Electrical Engineering Graduate Program, Federal University of Technology - Paraná. Cornélio Procópio, 2018.

This is the english abstract. (maximum of 500 words).

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

LISTA DE ILUSTRAÇÕES

FIGURA 1 – Camadas do Software Android	21
QUADRO 1 – Requisitos Funcionais	22
QUADRO 2 – Requisitos Não-Funcionais	23
FIGURA 2 – Diagrama de Classes	24
FIGURA 3 – Diagrama de Casos de Uso	25
FIGURA 4 – Diagrama de Atividades	26
FIGURA 5 – Protótipo Tela Inicial	27
FIGURA 6 – Protótipo Tela Preferências	28
FIGURA 7 – Protótipo Tela Sobre	29

LISTA DE FIGURAS

FIGURA 1 – Camadas do Software Android	21
FIGURA 2 – Diagrama de Classes	24
FIGURA 3 – Diagrama de Casos de Uso	25
FIGURA 4 – Diagrama de Atividades	26
FIGURA 5 – Protótipo Tela Inicial	27
FIGURA 6 – Protótipo Tela Preferências	28
FIGURA 7 – Protótipo Tela Sobre	29

LISTA DE TABELAS

TABELA 1 – Cronograma de atividades	30
---	----

LISTA DE QUADROS

QUADRO 1 – Requisitos Funcionais 22

QUADRO 2 – Requisitos Não-Funcionais 23

LISTA DE ACRÔNIMOS

SUMÁRIO

1	INTRODUÇÃO	15
1.1	PROBLEMA	15
1.2	JUSTIFICATIVA	16
1.3	OBJETIVOS	16
1.3.1	Objetivo Geral	16
1.3.2	Objetivos Específicos	16
1.4	ORGANIZAÇÃO DO TEXTO	17
2	FUNDAMENTAÇÃO TEÓRICA	18
3	PROPOSTA	19
3.1	TECNOLOGIAS E FERRAMENTAS	19
3.2	MÉTODO	19
3.3	ANÁLISE E DESENVOLVIMENTO	20
3.3.1	Arquitetura	20
3.3.2	Requisitos	21
3.3.3	Diagrama de Classes	23
3.3.4	Diagrama de Casos de Uso	24
3.3.5	Diagrama de Atividades	25
3.4	PROTÓTIPOS DE TELA	26
3.5	CRONOGRAMA	29
	REFERÊNCIAS	31

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 em 31 pessoas com 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 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 Windows 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 restrições 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 mecanismo 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.
- Capítulo 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*, existem algumas ferramentas que trabalham somente com *software* e que exercem essa função juntamente com algum dispositivo de reprodução de áudio que contenha microfone.

Para computadores de mesa e notebooks com sistema operacional Windows, existe a ferramenta "Software Mais Fluência Win DAF/FAF Software", desenvolvida em 2009 pelo Henrique Confessor, é *freeware* podendo ser distribuída e utilizada livremente. Disponibilizada gratuitamente para *download* no site da "Abra Gagueira"([CONFESSOR, 2009](#)).

Para dispositivos móveis com sistema operacional Android ou IOS existe o *DAF 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](#)).

Com o intuito de fornecer o *Feedback Auditivo Atrasado* (FAA), 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](#)).

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.

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:

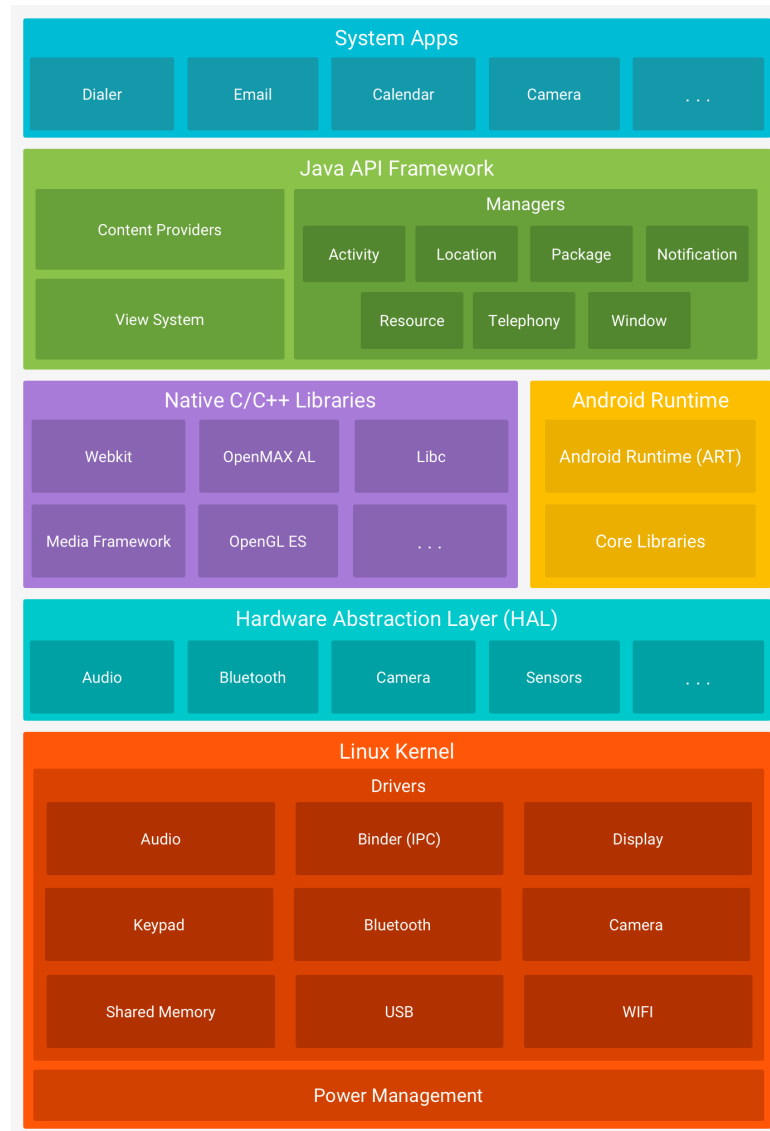
- *Sprint*: são iterações, ciclos de desenvolvimento que começam numa reunião de planejamento (*Sprint Planning*) e terminam com a revisão (*Sprint Review*) e a retrospectiva (*Sprint Retrospective*).
- *Product Owner*: é o responsável por definir prioridades a serem desenvolvidas em cada *sprint* e 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 a seguir apresenta a maioria dos componentes da plataforma Android.



Fonte: (DEVELOPERS, 2018)

Figura 1 – 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 o que 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.

Quadro 1 – Requisitos Funcionais

Id	Descrição	Prioridade	Requisitos Relacionados
RF01	O aplicativo deve permitir ao usuário editar as preferências de utilização do aplicativo, como tipo de fone utilizado, altura do volume, frequência e delay.	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 manter salva as preferências de utilização.	Importante	RF01 - RF04
RF04	O aplicativo deve fornecer a opção de um tutorial em seu primeiro uso.	Importante	N/A
RF05	O aplicativo deve permitir a utilização de fone <i>bluetooth</i> .	Importante	RF01 - RF002
RF06	O aplicativo deve permitir ao usuário uma opção de não receber ligações, enquanto o aplicativo estiver executando a simulação do efeito coro.	Desejável	RF01
RF07	O aplicativo deve conter uma tela de ajuda, onde existirá informações sobre a utilização do aplicativo.	Desejável	RF03
RF08	O aplicativo deve permitir visualizar o registro de vezes em que o usuário utilizou o aplicativo.	Desejável	RF02
RF09	O aplicativo deve permitir modos personalizados, como a criação de: modo casa, modo apresentação, modo tutorial, entre outros. Onde cada modo possui preferências pré-definidas.	Desejável	RF01-RF03

Fonte: O Autor.

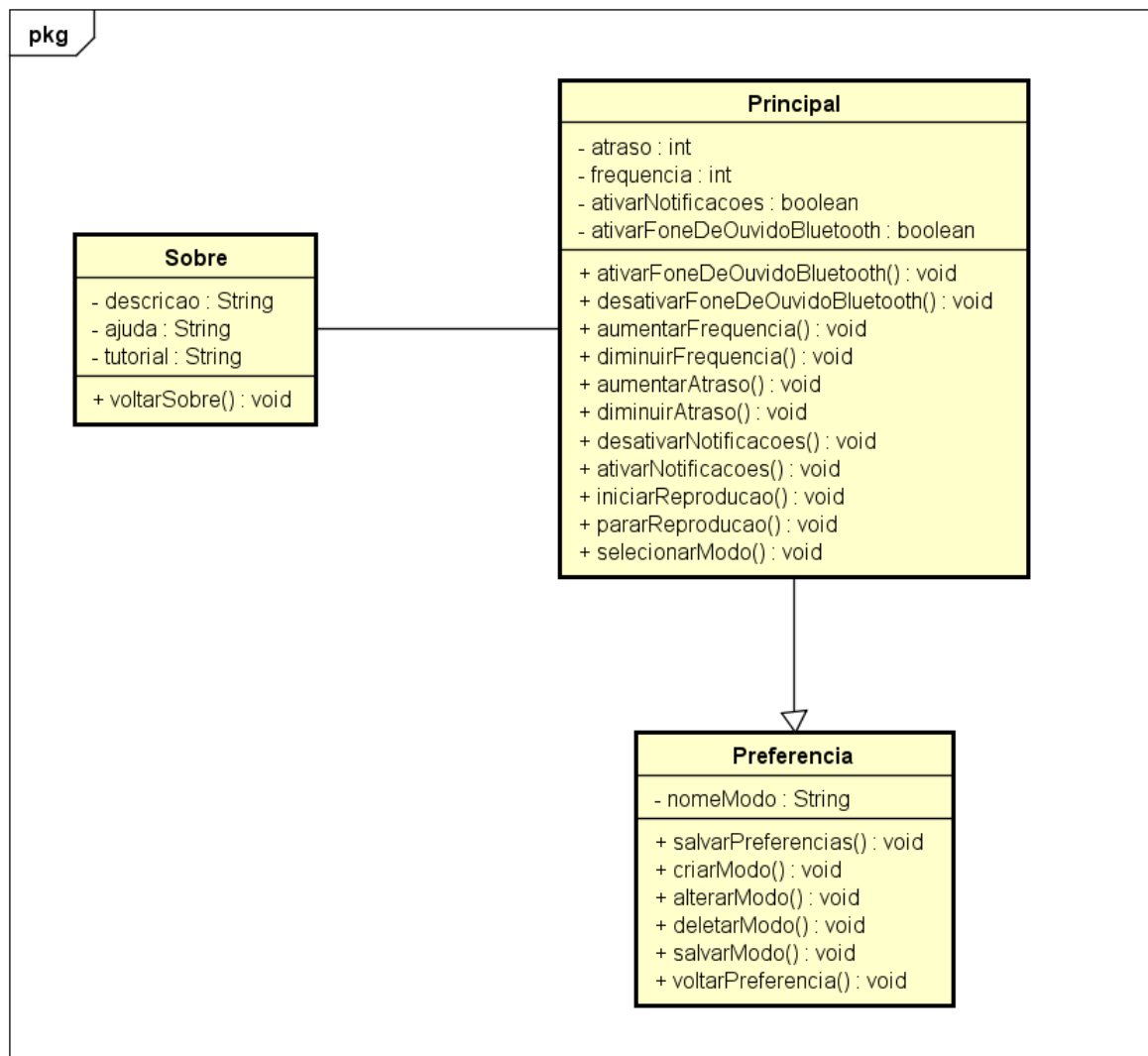
Quadro 2 – Requisitos Não-Funcionais

Id	Descrição	Categoria	Prioridade	Requisitos Relacionados
RNF01	O aplicativo deve ser desenvolvido para a plataforma Android.	Compatibilidade	Essencial	RNF03
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 <i>delay</i> , botão iniciar/-desligar, e opção de configurações).	Usabilidade	Desejável	RNF02

Fonte: O Autor.

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).



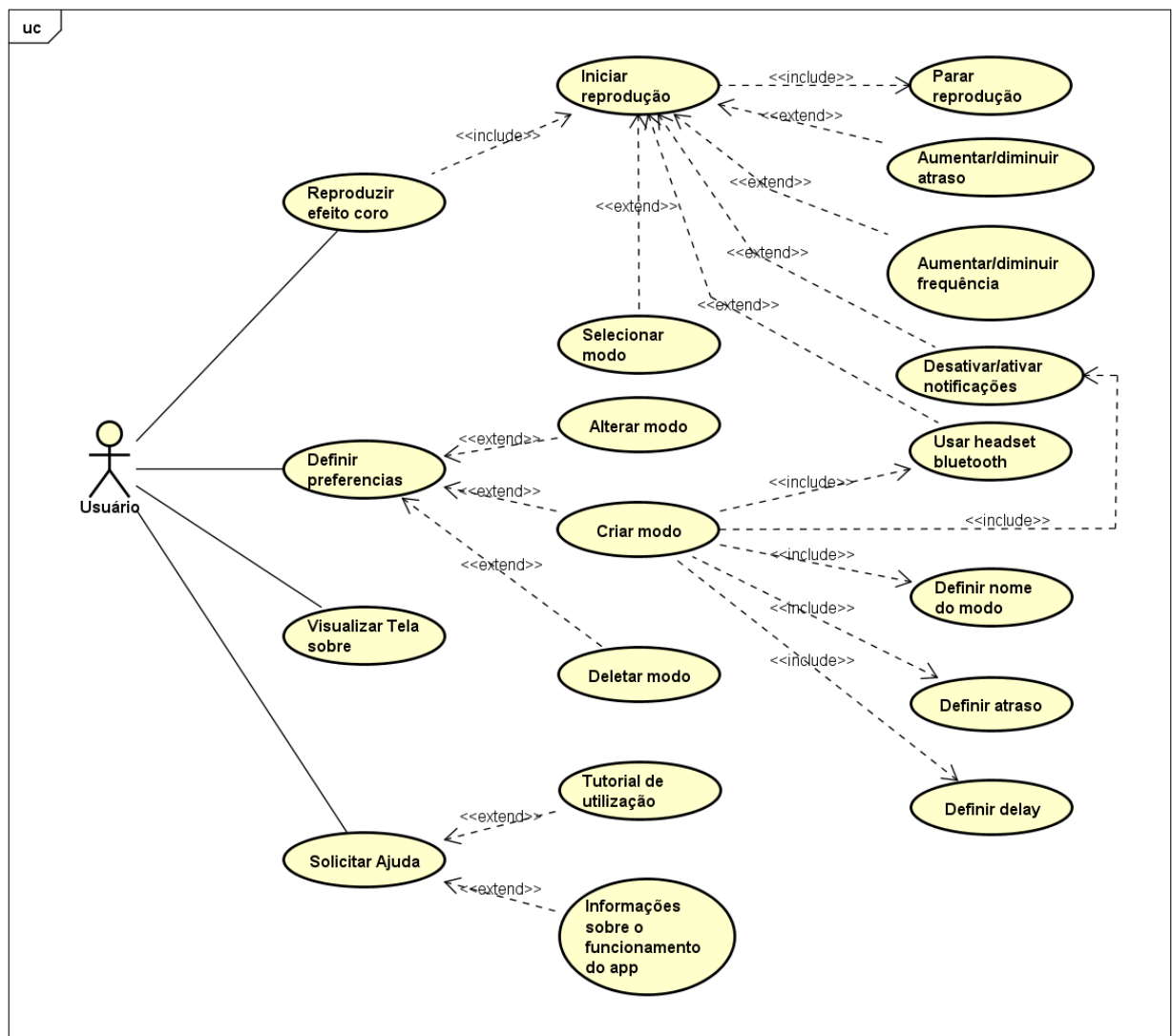
powered by Astah

Fonte: O Autor.

Figura 2 – Diagrama de Classes.

3.3.4 Diagrama de Casos de Uso

Apresenta-se o diagrama de casos de uso, documenta 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).



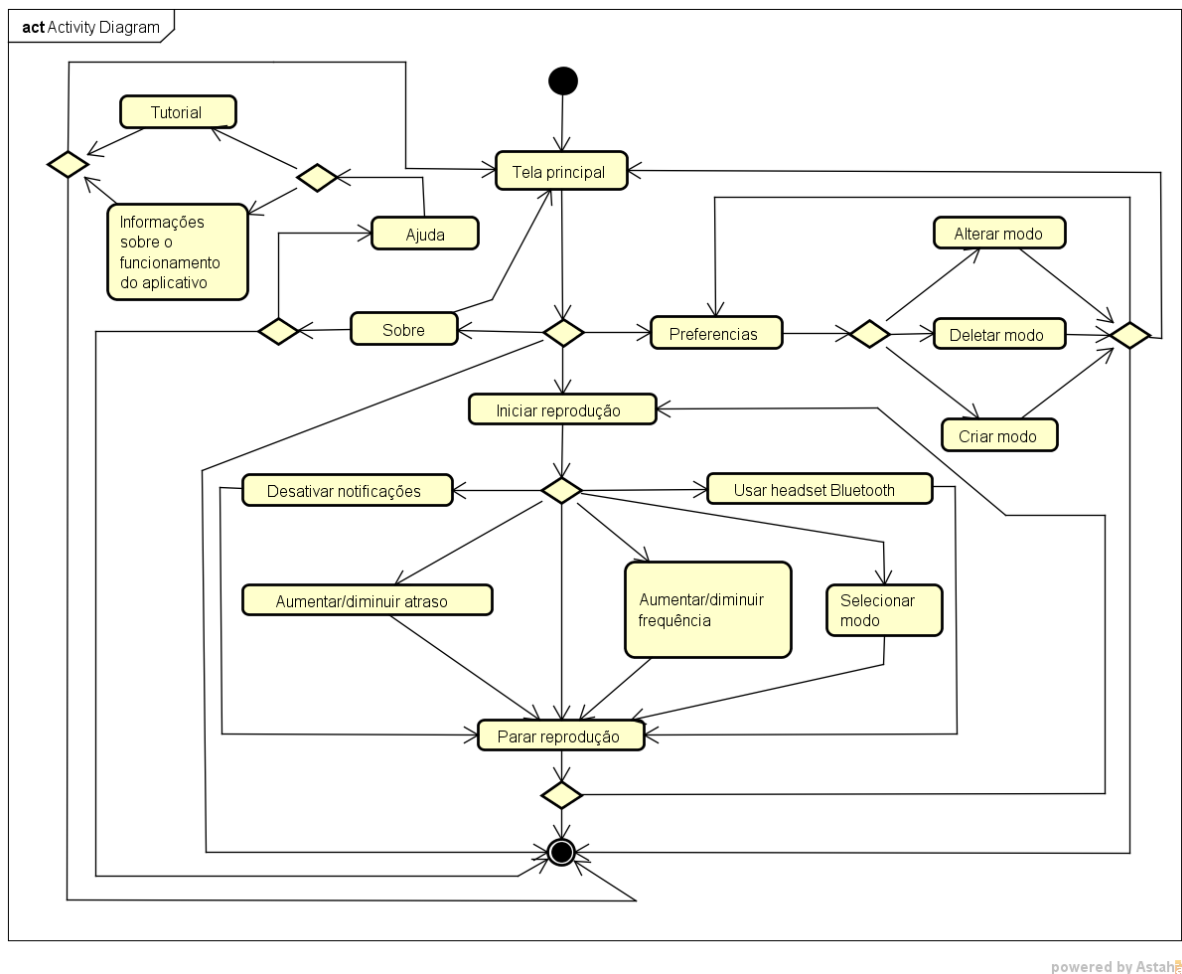
powered by Astah

Fonte: O Autor.

Figura 3 – 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).



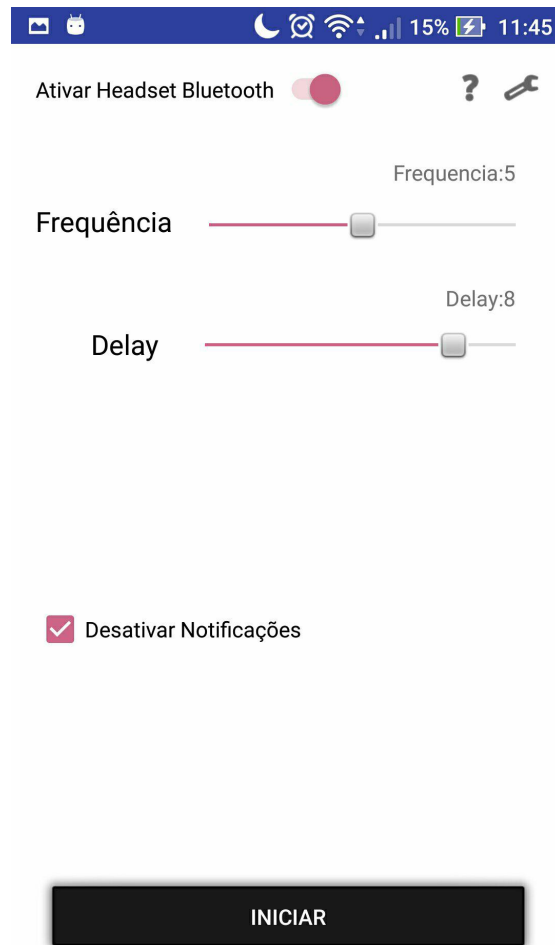
Fonte: O Autor.

Figura 4 – Diagrama de Atividades.

3.4 PROTÓTIPOS DE TELA

Apresenta-se os protótipos de tela 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 a opção de ativar e desativar a função de utilizar *headset bluetooth*, localizado no canto superior esquerdo da tela, juntamente com desativar as notificações marcando o *combobox* "Desativar notificações". Desta tela também existe a opção de navegar entre as telas "Sobre" e "Preferências", selecionando o ícone referente a cada tela localizados no canto superior direito.



Fonte: O Autor.

Figura 5 – Protótipo Tela Inicial.

- Tela Preferências: Nesta tela o usuário pode criar, alterar, ou excluir um modo, adicionando o atraso e frequência desejados, além de selecionar as opções de ativar/desativar *headset bluetooth* e notificações;

DAF

Preferências

Delay

Frequência

☐ Ativar headset bluetooth

☒ Desativar notificações

CRIAR MODO

EXCLUIR MODO

ALTERAR MODO

Fonte: O Autor.

Figura 6 – Protótipo Tela Preferência.

- Tela Sobre: Nesta tela o usuário encontra informações sobre o aplicativo, assim como informações sobre o funcionamento do aplicativo e um tutorial de ajuda.



Fonte: O Autor.

Figura 7 – Protótipo Tela Sobre.

3.5 CRONOGRAMA

Apresenta-se o cronograma, determinando todas as atividades do processo de desenvolvimento do software, com o período definido de 10 meses, definido de acordo com o programa "Protagonismo Estudantil", projeto ao qual o aplicativo apresentado neste documento foi contemplado com bolsa remunerada.

Fase	Out	Nov	Mar	Abr	Mai	Jun	Jul	Ago	Set	Out
Levantamento de requisitos	•									
Análise de requisitos	•									
Projeto		•								
Implementação			•	•	•	•	•			
Testes				•	•	•	•	•		
Implantação									•	
Divulgação										•

Fonte: O Autor.

Tabela 1 – Cronograma de atividades

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>>. 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>>. Citado na página 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>. Citado na página 15.
- BUCHER, 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>>. Citado na página 15.
- CONFESSOR, Henrique. Software mais fluência win daf/faf software. **Abra Gagueira**, 2009. Disponível em: <http://www.abragagueira.org.br/mais_fluencia.asp>. Citado na página 18.
- DEVELOPERS, Android. Arquitetura da plataforma android. **Android Developers**, 2018. Disponível em: <<https://developer.android.com/guide/platform/?hl=pt-br>>. 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>>. Citado na página 18.
- MERLO, Sandra. Caracterização da gagueira. **Instituto Brasileiro de Fluência - IBF**, 2013. Disponível em: <http://www.gagueira.org.br/conteudo.asp?id_conteudo=29>. Citado na página 15.
- 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/>>. Citado na página 15.
- 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>>. 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>>. 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>>. Citado na página 19.
- UDEMO. Efeito coro. **Folha de São Paulo**, 2008. Disponível em: <http://www.udemo.org.br/Leituras/Leituras_161.htm>. Citado 2 vezes nas páginas 15 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/>>. 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/>>. Citado na página 21.