

UNIVERSIDADE REGIONAL DE BLUMENAU
CENTRO DE CIÊNCIAS EXATAS E NATURAIS
CURSO DE CIÊNCIA DA COMPUTAÇÃO – BACHARELADO

VOCAL: ASSISTENTE PARA O USO DE SMARTPHONES
OPERADO POR VOZ

ELI TONNY DE SOUZA

BLUMENAU
2016

ELI TONNY DE SOUZA

VOCAL: ASSISTENTE PARA O USO DE SMARTPHONES

OPERADO POR VOZ

Trabalho de Conclusão de Curso apresentado ao curso de graduação em Ciência da Computação do Centro de Ciências Exatas e Naturais da Universidade Regional de Blumenau como requisito parcial para a obtenção do grau de Bacharel em Ciência da Computação.

Prof. Aurélio Faustino Hoppe, Mestre - Orientador

BLUMENAU
2016

VOCAL: ASSISTENTE PARA O USO DE SMARTPHONES
OPERADO POR VOZ

Por

ELI TONNY DE SOUZA

Trabalho de Conclusão de Curso aprovado para obtenção dos créditos na disciplina de Trabalho de Conclusão de Curso II pela banca examinadora formada por:

Presidente: _____
Prof. Aurélio Faustino Hoppe, M.Sc – Orientador, FURB

Membro: _____
Prof. Dalton Solano dos Reis, M.Sc – FURB

Membro: _____
Prof. Joyce Martins, M.Sc – FURB

Blumenau, 06 de julho de 2016

Dedico este trabalho aos meus pais e aos amigos que sempre serviram como um porto seguro em minha vida.

AGRADECIMENTOS

A Deus por me conceder algumas linhas neste grande livro da vida.

Aos meus pais, Adilton Antonio de Souza e Irene Strieder, por me apoiarem e incentivarem nesta jornada.

Aos meus amigos, em especial aos amigos Anderson Gielow e Deyvid Berndt pelas contribuições, ideias e ajuda nos testes.

Aos meus colegas de faculdade André V. Bampi, Gustavo Sabel, Maicon M. Gerardi, Reinoldo Krause Junior, Vivian de Lima Panzenhagen e William Leander Seefeld por todos os trabalhos realizados e por sempre poder contar sua ajuda.

Ao meu orientador Aurélio Faustino Hoppe por todo o apoio, amizade e por ser um referencial de competência.

Não se pode aprender nada de uma lição que não venha acompanhada da dor. Já que não se pode conseguir nada sem um sacrifício. Mas quando se aguenta essa dor e a supera, as pessoas conseguem um coração forte que não perde para nada. Sim, um coração de aço.

Edward Elric

RESUMO

Este trabalho descreve o desenvolvimento de um aplicativo Android com o objetivo de auxiliar pessoas com deficiência visual na execução de ações básicas em *smartphones*. O aplicativo realiza o reconhecimento da voz do usuário e transforma um comando recebido em uma ação de interação com o dispositivo. Essas ações podem ser, por exemplo, de leitura de tela, toque ou escrita. Para a realização do reconhecimento de voz, sintetização de voz, captura de eventos do sistema operacional Android e aplicação de ações sob os componentes de tela foram utilizadas as bibliotecas `accessibility`, `accessibilityservice`, `speech` e `android.speech.tts`. Os resultados obtidos a partir de experimentos e testes de usabilidade demonstram que o aplicativo desenvolvido, apesar das limitações, é capaz de auxiliar na interação com *smartphones* de forma satisfatória.

Palavras-chave: Acessibilidade. *Smartphones*. Linguagem natural. Android.

ABSTRACT

This work presents the development of an Android application with the objective to help people with visual impairment in the execution of basic activities in smartphones. The application realizes the recognition of user voice and transforms a received command in an interaction action with the device. These actions can be, for example, read screen, touch or write. For the voice recognition, voice synthesis, Android events capture and action application on the screen components were used the libraries `accessibility`, `accessibilityservice`, `speech` and `android.speech.tts`. The results obtained through the experiments and usability tests show that the developed application, although the limitations, is able to help in the interaction with smartphones in a satisfactory way.

Key-words: Accessibility. Smartphones. Natural language. Android.

LISTA DE FIGURAS

Figura 1 – Arquitetura Simplificada dos Componentes de um agente de conversação.....	20
Figura 2 – Arquitetura geral de um agente de conversação.	21
Figura 3 – Assistente pessoal Siri.....	22
Figura 4 – Assistente pessoal S Voice	23
Figura 5 – Assistente pessoal Cortana	24
Figura 6 – Diagrama de casos de uso.....	27
Figura 7 – Diagrama de Classes	28
Figura 8 – Fluxo da Aplicação	30
Figura 9 – Estrutura da frase	34
Figura 10 – Configuração de acessibilidade.....	44
Figura 11 – Tela inicial	44
Figura 12 – Tela de reconhecimento de voz.....	45
Figura 13 – Interação com aplicações.....	45

LISTA DE QUADROS

Quadro 1 – Comparação entre as características dos trabalhos relacionados	24
Quadro 2 – Parâmetros de configuração	31
Quadro 3 – Tratamento de Evento	32
Quadro 4 – Reconhecimento de voz	33
Quadro 5 – Resultado do reconhecimento de voz	33
Quadro 6 – Encontrando ação e sujeito.....	34
Quadro 7 – Ação leitura	35
Quadro 8 – Ação fala	35
Quadro 9 – Ação toque	35
Quadro 10 – Ação botão.....	36
Quadro 11 – Ação buscar	36
Quadro 12 – Ação desliga	37
Quadro 13 – Ação <i>home</i>	37
Quadro 14 – Ação escreva.....	38
Quadro 15 – Ação discar	39
Quadro 16 – Ação rolar.....	40
Quadro 17 – Ação letra	40
Quadro 18 – Ação limpar	41
Quadro 19 – Ação colar	41
Quadro 20 – Método <code>executaAcao</code>	42
Quadro 21 – Método <code>encontraNoTela</code>	43
Quadro 22 – Perfil dos voluntários	47
Quadro 23 – Lista de tarefas.....	47
Quadro 24 – Avaliação de usabilidade	48
Quadro 25 – Comparação com os trabalhos correlatos.....	49
Quadro 26 – UC01 Ação ler	56
Quadro 27 – UC02 Ação toque.....	56
Quadro 28 – UC03 Ação desligar.....	56
Quadro 29 – UC04 <i>home</i>	56
Quadro 30 – UC05 Ação botão.....	56

Quadro 31 – UC06 Ação escrever.....	57
Quadro 32 – UC07 Ação buscar	57
Quadro 33 – UC08 Ação discar	57
Quadro 34 – UC09 Ação rolar.....	57
Quadro 35 – UC10 Ação letra.....	58
Quadro 36 – UC11 Ação colar.....	58
Quadro 37 – UC12 Ação limpar	58
Quadro 38 – UC13 Ação falar.....	58
Quadro 39 – Verbos e ações.....	59
Quadro 40 – Questionário de perfil de usuário.....	60
Quadro 41 – Lista de tarefas.....	61
Quadro 42 – Avaliação de usabilidade	62

LISTA DE TABELAS

Tabela 1 – Censo Demográfico 2010 – Tipos de Deficiência.....	18
Tabela 2 – Censo Demográfico 2010 - Deficientes Visuais Graves no Brasil.....	18

LISTA DE ABREVIATURAS E SIGLAS

SMS – *Short Message Service*

UML – *Unified Modeling Language*

SUMÁRIO

1 INTRODUÇÃO	15
1.1 OBJETIVOS	16
1.2 ESTRUTURA	16
2 FUNDAMENTAÇÃO TEÓRICA	17
2.1 TECNOLOGIA ASSISTIVA	17
2.2 PROCESSAMENTO DE LINGUAGEM NATURAL.....	19
2.3 TRABALHOS CORRELATOS	21
2.3.1 Siri.....	22
2.3.2 S Voice	22
2.3.3 Cortana	23
2.3.4 Comparação entre as características dos trabalhos correlatos.....	24
3 DESENVOLVIMENTO	26
3.1 REQUISITOS	26
3.2 ESPECIFICAÇÃO	26
3.2.1 Casos de uso	26
3.2.2 Diagrama de Classes	27
3.3 IMPLEMENTAÇÃO	29
3.3.1 Técnicas e ferramentas utilizadas	29
3.3.2 Fluxo geral da aplicação.....	30
3.3.3 Configuração da aplicação	31
3.3.4 Tratativa de Eventos	32
3.3.5 Reconhecimento de voz	32
3.3.6 Tratamento da frase reconhecida	33
3.3.6.1 Ações de exteriorização de voz.....	34
3.3.6.2 Ações de toque	35
3.3.6.3 Ação de parada.....	36
3.3.6.4 Ação de retorno	37
3.3.6.5 Ações para digitação.....	37
3.3.6.6 Ação para rolagem.....	39
3.3.6.7 Ações de ditado	40
3.3.6.8 Métodos auxiliares.....	41

3.3.7 Operacionalidade da implementação	43
3.4 RESULTADOS E DISCUSSÕES	46
3.4.1 Experimento 01: Avaliação de usabilidade com pessoas sem deficiência visual	46
3.4.1.1 Metodologia	46
3.4.1.2 Aplicação do teste.....	46
3.4.1.3 Análise dos resultados obtidos	47
3.4.2 Experimento 02: Avaliação de usabilidade por pessoas com deficiência visual.....	48
3.4.3 Comparação com os trabalhos correlatos.....	49
4 CONCLUSÕES	51
4.1 EXTENSÕES	52
REFERÊNCIAS.....	53
APÊNDICE A – DESCRIÇÃO DOS CASOS DE USO.....	55
APÊNDICE B – RELAÇÃO DE VERBOS E AÇÕES.....	59
APÊNDICE C – LISTA DE TAREFAS E QUESTIONÁRIO DE AVALIAÇÃO DE USABILIDADE	60

1 INTRODUÇÃO

A tecnologia está cada vez mais presente nas vidas das pessoas, sendo utilizada para os mais diversos fins. É importante assegurar que pessoas portadoras de deficiência também possam desfrutar destas tecnologias (GARCÍA; GALVÃO FILHO, 2012, p. 8). De acordo com o Instituto Brasileiro de Geografia e Estatística (2010), 3,4% da população brasileira possui algum tipo de deficiência visual grave. Estes números mostram o tamanho da necessidade no que se diz respeito a tecnologias assistivas. Segundo Hazard, Galvão Filho e Rezende (2007, p. 29), as tecnologias assistivas são quaisquer ferramentas, processos ou recursos que visam auxiliar pessoas com algum tipo de dificuldade ou deficiência, podendo ser desde um objeto simples como uma colher adaptada até um *software* de computador.

Entre os *softwares* que visam auxiliar pessoas com deficiência visual no uso de tecnologias com forte apelo visual, como computadores e *smartphones*, estão os chamados leitores de tela ou sintetizadores de voz. São programas que enviam para os usuários as informações contidas em tela no formato de som, geralmente sendo acionados por atalhos de teclado, pelo toque de tela ou até por comandos de voz, no caso dos assistentes pessoais que realizam processamento de linguagem natural para identificação de comandos. Isto proporciona a essas pessoas facilidade de acesso à informação (HAZARD; GALVÃO FILHO; REZENDE, 2007, p. 56). O acesso a essas tecnologias por parte das pessoas portadoras de deficiência é também uma forma de inclusão social. Contudo, essa possibilidade representa mais do que isso, pois, traz para essas pessoas autonomia e a possibilidade uma vida independente, culminando assim, em uma melhor qualidade de vida (HAZARD; GALVÃO FILHO; REZENDE, 2007, p. 13).

O desenvolvimento de tecnologias assistivas é importante, pois, segundo Radabaugh (1993) para as pessoas sem deficiência, a tecnologia torna as coisas mais fáceis e para as pessoas com deficiência, a tecnologia torna as coisas possíveis. As ferramentas existentes no mercado atualmente, como os leitores de tela e assistentes pessoais, ainda possuem algumas limitações, o que pode gerar dificuldades no uso de aparelhos como *smartphones* por parte das pessoas com deficiência visual. Estas limitações estão relacionadas ao modo de operação dessas aplicações, que geralmente dependem do toque de tela para a navegação e execução das funções e as funcionalidades disponíveis em cada ferramenta.

Diante desta necessidade, este trabalho apresenta o desenvolvimento de um aplicativo para auxiliar pessoas com deficiência visual na utilização de *smartphones*. A principal característica deste aplicativo é a utilização de linguagem natural como meio de interação com

o usuário para a realização de ações como leitura de tela, abrir aplicações e envio de mensagens.

1.1 OBJETIVOS

O objetivo deste trabalho é o desenvolvimento de um aplicativo assistente que auxilie pessoas com deficiência visual a realizarem tarefas básicas em *smartphones*.

Os objetivos específicos do trabalho são:

- a) interpretar os comandos de voz, via processamento de linguagem natural, para intermediar a interação entre o usuário e algumas aplicações dos *smartphones* com a plataforma Android;
- b) fornecer *feedback* auditivo, ao usuário, do que está aparecendo na tela do *smartphone*, assim como, questionando-o sobre o que deseja fazer.

1.2 ESTRUTURA

Este trabalho está dividido em quatro capítulos. O primeiro capítulo apresenta a introdução e os objetivos do trabalho. O segundo capítulo apresenta a fundamentação teórica sobre tecnologia assistiva e processamento de linguagem natural. O terceiro capítulo demonstra o desenvolvimento do trabalho com requisitos, especificações, implementação, resultados e operacionalidade da aplicação. Por fim, o quarto capítulo relata as conclusões e possíveis extensões.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo está organizado em três seções. A seção 2.1 expõe informações referentes à tecnologia assistiva. A seção 2.2 apresenta alguns conceitos relevantes ao processamento de linguagem natural. Por fim, na seção 2.3 estão relacionados três trabalhos correlatos ao trabalho proposto.

2.1 TECNOLOGIA ASSISTIVA

Tecnologia Assistiva é uma expressão nova, que se refere a um conceito ainda em pleno processo de construção e sistematização (GARCÍA; GALVÃO FILHO, 2012, p. 12). De acordo com o autor, a utilização de recursos de Tecnologia Assistiva, entretanto, remonta aos primórdios da história da humanidade ou até mesmo da pré-história. Qualquer pedaço de pau utilizado como uma bengala improvisada, por exemplo, caracteriza o uso de um recurso de Tecnologia Assistiva (GARCÍA; GALVÃO FILHO, 2012, p. 12).

Pode-se sumarizar tecnologia assistiva como:

[...] toda e qualquer ferramenta, recurso ou estratégia e processo desenvolvido e utilizado com a finalidade de proporcionar maior independência e autonomia à pessoa com deficiência. São considerados como tecnologia assistiva, portanto, desde artefatos simples, como uma colher adaptada ou um lápis com uma empunhadura mais grossa para facilitar a preensão, até sofisticados programas especiais de computador que visam à acessibilidade (HAZARD; GALVÃO FILHO; REZENDE, 2007, p. 29).

Com o desenvolvimento da tecnologia, surgiu uma grande gama de recursos tecnológicos com um forte apelo visual, como os computadores por exemplo. Para utilizar tais recursos, pessoas portadoras de deficiência visual passaram a necessitar de *softwares* especiais ou suportes tecnológicos. São os chamados leitores de telas ou sintetizadores de voz, que proporcionam aos portadores de deficiência visual facilidade de acesso à informação (internet, editores de texto, planilhas eletrônicas, etc.) (HAZARD; GALVÃO FILHO; REZENDE, 2007, p. 56).

Segundo o Instituto Brasileiro de Geografia e Estatística (2010), em 2010 o Brasil possuía 35.774.392 milhões de pessoas com algum tipo de deficiência visual. A Tabela 1 mostra a quantidade de pessoas portadoras de deficiências de acordo com o Censo.

Tabela 1 – Censo Demográfico 2010 – Tipos de Deficiência

	Visual	Motora	Auditiva	Mental/ Intelectual	Total
Homens	14.919.686	4.979.618	4.908.611	1.409.597	26.217.512
Mulheres	20.854.706	8.285.981	4.808.707	1.201.938	35.151.332
Total	35.774.392	13.265.599	9.717.318	2.611.536	61.368.844

Fonte: Instituto Brasileiro de Geografia e Estatística (2010).

Observando mais alguns dados do Instituto Brasileiro de Geografia e Estatística (2010), do total de portadores de deficiência visual no Brasil, 6.562.910 milhões de pessoas possuem uma deficiência visual grave. A Tabela 2 mostra a quantidade de pessoas portadoras de deficiência visual grave no Brasil.

Tabela 2 – Censo Demográfico 2010 - Deficientes Visuais Graves no Brasil

Faixa Etária	Quantidade de pessoas no Brasil
0 a 5 anos	61.278
6 a 14 anos	302.724
15 a 59 anos	3.724.501
60 anos ou mais	2.474.407
Total	6.562.910

Fonte: Instituto Brasileiro de Geografia e Estatística (2010).

Tais números mostram que uma parcela da população precisa de algum tipo de auxílio para poder fazer uso das tecnologias disponíveis no mercado. Estes números também comprovam a importância que as tecnologias assistivas tem como meio de inclusão das pessoas com deficiência.

De acordo com Bersch (2013, p. 7), entre as tecnologias que visam auxiliar essa parcela da população com algum tipo de deficiência, seja ela sensorial (visual ou auditiva), intelectual ou motora, estão os *softwares* de reconhecimento de voz. Tais *softwares* são um meio de tornar o computador acessível às pessoas com essas privações. Porém, não são o único meio, podendo ser citados também teclados virtuais, mouses especiais, *softwares* para ajuste de cores e tamanho (efeito lupa), impressoras de braille, entre outros.

No caso dos *smartphones*, a interação geralmente se dá através de leitores de tela e de assistentes pessoais. Os leitores realizam a leitura de tela de acordo com a movimentação dos dedos e abrem as aplicações com cliques duplos. A digitação de texto se dá através do teclado virtual, ou por meio de ditado. Tais funcionalidades variam de acordo com o fabricante, podendo não estar disponíveis em algumas plataformas. Porém, há a possibilidade de aquisição de aplicativos de terceiros disponíveis no mercado para suprir essas necessidades (NAVARRO, 2012, p. 22). Já no caso dos assistentes pessoais, a interação ocorre através de comandos de voz. As funcionalidades disponíveis e o nível de interpretação de linguagem natural também variam de acordo com o fabricante. Apesar de não serem ferramentas

desenvolvidas especificamente para pessoas com deficiência visual, suas características de usabilidade as tornam um meio de interação acessível.

2.2 PROCESSAMENTO DE LINGUAGEM NATURAL

A ideia de dar aos computadores a habilidade de processar linguagem humana é tão velha quanto à ideia dos próprios computadores (JURAFSKY; MARTIN, 2009, p. 35). Tal ideia se mostra interessante, considerando que a linguagem é a marca da humanidade e da ciência. E a conversação ou o diálogo são o aspecto fundamental e mais privilegiado da linguagem (JURAFSKY; MARTIN, 2009, p. 847).

Conforme afirmou Braga (2008, p. 2), o uso de linguagem natural por máquinas possui inúmeras vantagens, entre elas:

- a) facilidade de aquisição de informação;
- b) associação a dispositivos móveis;
- c) acessibilidade à informação por parte de portadores de deficiência.

As aplicações de processamento de linguagem se diferenciam das demais aplicações de processamento de dados pelo uso do conhecimento sobre a língua (JURAFSKY; MARTIN, 2009, p. 36). Segundo Jurafsky e Martin (2009, p. 36), este conhecimento linguístico se resume nos seguintes pontos:

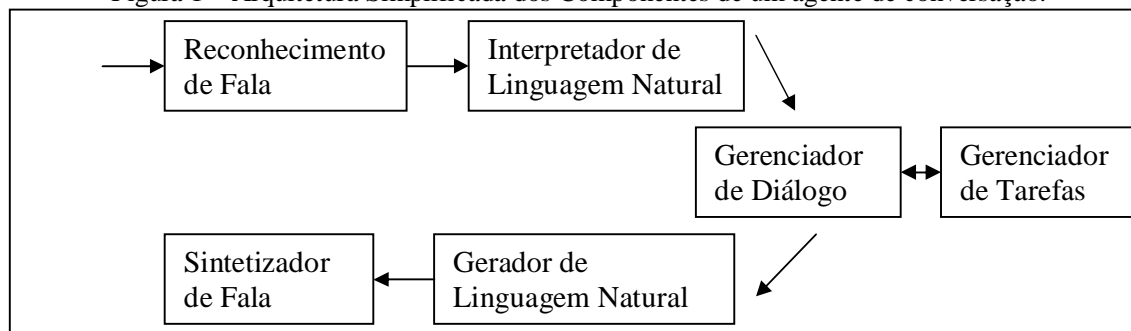
- a) fonética e fonologia: conhecimento sobre os sons da língua;
- b) morfologia: conhecimento sobre o significado dos componentes das palavras;
- c) sintaxe: conhecimento das estruturas relacionais entre palavras;
- d) semântica: conhecimento do significado das palavras;
- e) pragmática: conhecimento do significado das metas e das intenções do locutor;
- f) discurso: conhecimento linguístico sobre locuções mais longas do que uma única frase.

A maior parte do esforço em cada uma das categorias acima está voltado para a resolução de ambiguidade (JURAFSKY; MARTIN, 2009, p. 38). Tais aspectos mostram que a conversação entre humanos é uma complexa atividade conjunta. Por causa das limitações da tecnologia atual, conversas entre humanos e máquinas são muito mais simples e menos naturais do que as conversas entre humanos. Porém, antes de tentar modelar um agente falante para conversar com humanos, é crucial entender algumas coisas sobre como humanos conversam uns com os outros (JURAFSKY; MARTIN, 2009, p. 849).

O diálogo é caracterizado pela tomada de turnos. O locutor A fala alguma coisa, então o locutor B, então o locutor A e assim consecutivamente. Porém, um diálogo não é uma sequência de locuções independentes. Pelo contrário, é uma sequência de locuções relacionadas que dependem de uma base comum entre as partes. Esta base é aquilo que evidencia que o ouvinte compreendeu o sentido e a intenção daquilo que foi dito (JURAFSKY; MARTIN, 2009, p. 849).

Tradicionalmente, são duas as áreas incluídas no Processamento da Fala: a Síntese da Fala, sistema que permite a conversão de texto em fala, e o reconhecimento de voz, sistema que possibilita a conversão de voz em texto. A partir destas noções básicas sobre conversação entre humanos, parte-se para a modelagem de um sistema de conversação entre homem e máquina. A Figura 1 mostra uma arquitetura clássica para um sistema de diálogo.

Figura 1 – Arquitetura Simplificada dos Componentes de um agente de conversação.

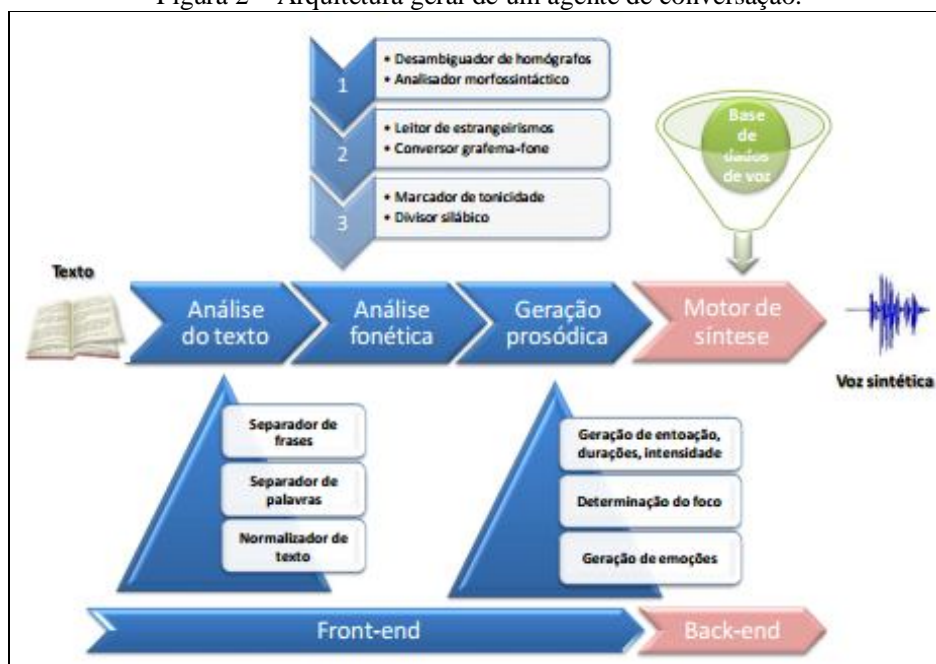


Fonte: adaptado de Jurafsky e Martin (2009, p. 857).

Na Figura 1, os componentes de Reconhecimento de Fala e Interpretador de Linguagem Natural são responsáveis por extrair o sentido da locução de entrada. Os componentes Síntetizador de Fala e Gerador de Linguagem Natural são responsáveis por transcrever a saída do sistema para linguagem natural. O componente Gerenciador de Diálogo controla todo o processo, juntamente com o componente Gerenciador de Tarefas que é quem detém todo o conhecimento específico em relação às atividades a serem realizadas (JURAFSKY; MARTIN, 2009, p. 857).

Segundo Braga (2008, p. 26) a arquitetura geral dos sistemas de conversação se divide em três grupos: pré-processamento de texto ou *front-end*, motor de síntese ou *back-end* e base de dados de voz ou *voice font*. A Figura 2 mostra a arquitetura proposta por Braga (2008, p. 26).

Figura 2 – Arquitetura geral de um agente de conversação.



Fonte: Braga (2008, p. 26).

O *front-end* é constituído por três componentes, sendo o primeiro o componente de análise de texto, que é responsável por separar frases, palavras e normalizar o texto. O segundo, o componente de análise fonética, que faz análise dos fonemas e o terceiro, o componente de geração prosódica que aproveita informações obtidas anteriormente como, tipo de frase, divisão silábica, marcação tônica para a geração de contornos prosódicos. O *back-end* por sua vez é composto pelo motor de síntese que interpreta a transcrição fonética gerada pelo *front-end* e a transforma em fala sintética (BRAGA, 2008, p. 26).

Braga (2008, p. xiii) também afirma que a dimensão e a qualidade da base de dados gerada dependem da técnica de síntese utilizada pelo *back-end*. Já o *front-end* depende das especificidades da língua e, no caso do português, ainda existem dificuldades na leitura de estrangeirismos, na desambiguação de homógrafos e mesmo na conversão grafema-fone.

2.3 TRABALHOS CORRELATOS

A seguir são apresentados três trabalhos correlatos ao proposto e uma comparação entre os mesmos. A seção 2.3.1 descreve a Siri, assistente pessoal inteligente de propriedade da Apple Inc (APPLE, 2015a). A seção 2.3.2 descreve o S Voice, assistente pessoal inteligente de propriedade da Samsung Inc (SAMSUNG, 2015a). A seção 2.3.3 apresenta o Cortana, assistente pessoal inteligente da Microsoft Corporation (MICROSOFT, 2015a). Por fim, na seção 2.3.4 é mostrado um quadro comparativo entre os trabalhos e uma discussão sobre suas características.

2.3.1 Siri

A Siri é um assistente pessoal de propriedade da Apple Inc. capaz de realizar várias tarefas por comandos de voz, tendo como foco ser um meio de interação mais amigável. Atualmente, é possível utilizar a Siri em português, porém nem todas as suas funcionalidades estão disponíveis no Brasil, ficando restritas a usuários de alguns países, como por exemplo, os Estados Unidos da América. Visando a facilidade de comunicação, a Siri é capaz de processar linguagem natural, diferente de outros assistentes pessoais do mercado que são capazes apenas de interpretar comandos de voz específicos (APPLE, 2015a). Devido a tais funcionalidades, a Siri é considerada como um dos recursos de acessibilidade dos dispositivos Apple, trabalhando integrada com o leitor de tela VoiceOver (APPLE, 2015b). A Figura 3 mostra um exemplo de resposta da Siri.

Figura 3 – Assistente pessoal Siri



Fonte: Apple (2015c).

A Siri possui uma série de funcionalidades, tais como o envio de mensagens de texto e e-mails ou ainda a consulta de ações na bolsa (APPLE, 2015c). Outra característica interessante, é que mesmo sem precisar de nenhum tipo de treinamento inicial, a Siri assimila características da voz e do dialeto falado pelo usuário, melhorando assim a sua capacidade de reconhecimento de voz com o passar do tempo (APPLE, 2015a).

2.3.2 S Voice

O S Voice também é um assistente pessoal de propriedade da Samsung Inc. Através de comandos de voz ele é capaz de executar uma variedade de tarefas, permitindo ao usuário

uma interação mais simples, sem o uso das mãos (SAMSUNG, 2015b). No momento, o S Voice possui suporte ao português e está disponível para uso apenas em alguns aparelhos da linha Galaxy. Apesar de interpretar linguagem natural, o S Voice depende de alguns comandos mais claros para poder executar tarefas (SCHAWARTZ, 2014, p. 511). O S Voice é considerado pela Samsung como uma das opções de acessibilidade dos dispositivos móveis da companhia (SAMSUNG, 2015c). Abaixo na Figura 4 é apresentado um exemplo do uso do S Voice.

Figura 4 – Assistente pessoal S Voice



Fonte: Samsung (2015d).

Entre as principais funcionalidades do S Voice estão o envio e recebimento de mensagens de texto, atender e efetuar chamadas, consultar dados dos contatos e consultar a previsão do tempo (SAMSUNG, 2015a). Contudo, apesar de todas as funcionalidades o S Voice ainda apresenta algumas inconsistências, em determinados momentos ignorando alguns dos comandos e fazendo aquilo que lhe parece adequado (SCHAWARTZ, 2014, p. 511).

2.3.3 Cortana

O Cortana é o assistente pessoal da Microsoft Corporation, similar ao S Voice e a Siri. Da mesma forma que os outros assistentes citados, executa diversas tarefas por comandos de voz. Por hora, o Cortana ainda não possui suporte ao idioma português (MICROSOFT, 2015d). Com o objetivo de ser um meio de interação mais fácil, o Cortana é capaz de interpretar linguagem natural (MICROSOFT, 2015a). A Figura 5 mostra a tela inicial do Cortana.

Figura 5 – Assistente pessoal Cortana



Fonte: adaptado de Microsoft (2015a).

O Cortana tem diversas funcionalidades, desde algumas mais básicas como o envio de *Short Message Service* (SMS) até algumas mais avançadas como a consulta da situação de vôos ou a própria conversa com o assistente (MICROSOFT, 2015b). Outro recurso interessante do Cortana é seu bloco de anotações. Nele, ficam registrados dados sobre os gostos e interesses do usuário. Essas anotações são utilizadas para que o assistente notifique o usuário sobre algo que lhe possa ser interessante (MICROSOFT, 2015c).

2.3.4 Comparação entre as características dos trabalhos correlatos

O Quadro 1 mostra uma breve comparação das características gerais dos trabalhos correlatos descritos nas subseções anteriores.

Quadro 1 – Comparação entre as características dos trabalhos relacionados

Trabalho/ Característica	Apple (2015)	Samsung (2015)	Microsoft (2015)
disponível em português	X	X	-
interpreta linguagem natural	X	X	X
utiliza sintetizador de voz	-	X	-
utiliza voz gravada	X	-	X
aprendizado contínuo	X	-	-

A partir do Quadro 1, percebe-se que os trabalhos possuem características semelhantes. As principais diferenças entre eles estão relacionadas à qualidade que cada um executa as funcionalidades disponibilizadas e as suas características de usabilidade.

Ao observar a característica disponível em português, percebe-se que apenas o Cortana (MICROSOFT, 2015) está atrás dos concorrentes, pois ainda não está disponível no idioma português. Porém, segundo Microsoft (2015a), em breve o Cortana estará disponível para outros idiomas, inclusive em português.

Analisando a característica de interpretação de linguagem natural, todos os trabalhos correlatos executam esta tarefa. Porém, a Siri (APPLE, 2015) destaca-se entre os três, pois, possui aprendizado contínuo, que faz com que sua capacidade de interpretação evolua de acordo com o seu uso.

Em relação às respostas de voz, o S Voice (SAMSUNG, 2015) está atrás do Cortana (MICROSOFT, 2015) e da Siri (APPLE, 2015), pois usa um sintetizador de voz enquanto os outros dois concorrentes utilizam voz humana gravada. O uso do sintetizador de voz não é um problema. Porém, não gera a mesma experiência e naturalidade proporcionada pelo uso de voz humana gravada.

3 DESENVOLVIMENTO

Este capítulo apresenta o desenvolvimento da aplicação. A seção 3.1 apresenta os requisitos da aplicação desenvolvida. A seção 3.2 exibe a especificação do trabalho. A seção 3.3 mostra a implementação das principais partes da aplicação, assim como as ferramentas utilizadas e a operacionalidade da aplicação. Por fim, a seção 3.4 discute os resultados obtidos.

3.1 REQUISITOS

O aplicativo desenvolvido deve:

- a) ler e utilizar um sintetizador de voz para indicar ao usuário o que está aparecendo na tela (Requisito Funcional - RF);
- b) capturar e transformar os comandos de voz humana, via processamento de linguagem natural, em comandos a serem executados pelo Android (RF);
- c) permitir que o usuário possa criar contatos na agenda telefônica (RF);
- d) permitir que o usuário possa consultar contatos da agenda telefônica (RF);
- e) permitir que o usuário possa usar a calculadora (RF);
- f) permitir que o usuário possa efetuar, atender ou recusar chamadas (RF);
- g) permitir que o usuário possa ler ou enviar mensagens (SMS, email, Whatsapp) (RF);
- h) permitir que o usuário possa adicionar ou consultar lembretes no calendário (RF);
- i) ser desenvolvida para a plataforma Android (Requisito Não-Funcional - RNF);
- j) ser de fácil utilização (RNF).

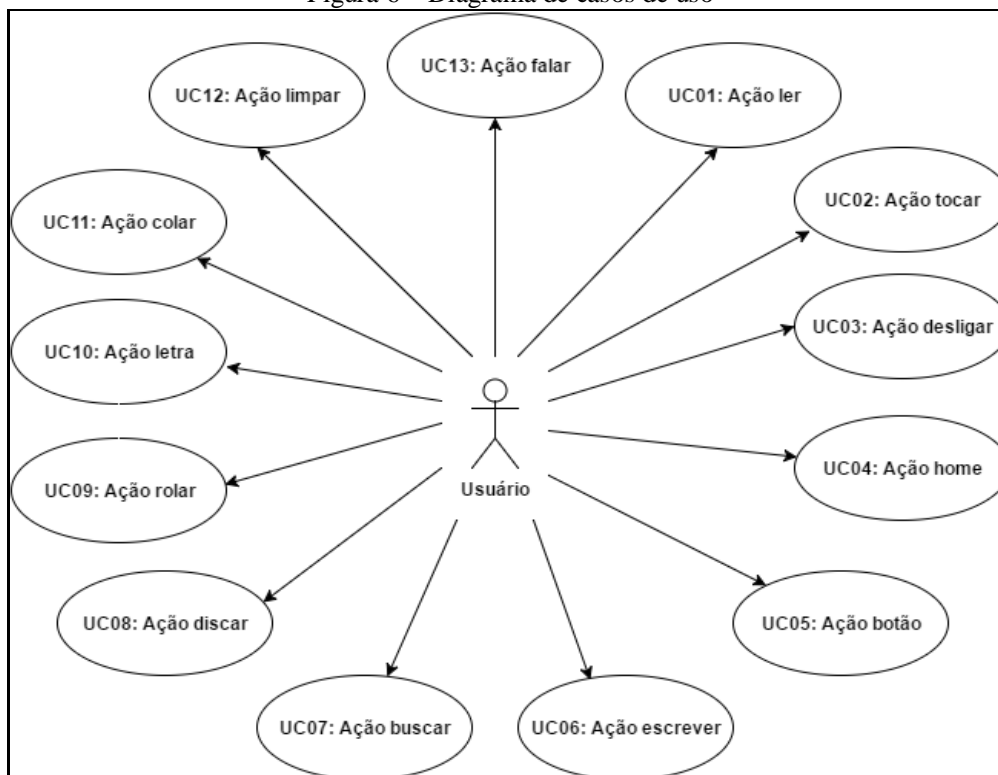
3.2 ESPECIFICAÇÃO

A especificação da aplicação foi desenvolvida de acordo com a *Unified Modeling Language* (UML). Para a elaboração dos diagramas foi utilizada a ferramenta draw.io.

3.2.1 Casos de uso

Os casos de uso que representam as funcionalidades da aplicação são exibidos na Figura 6. O detalhamento dos mesmos encontra-se no apêndice A.

Figura 6 – Diagrama de casos de uso

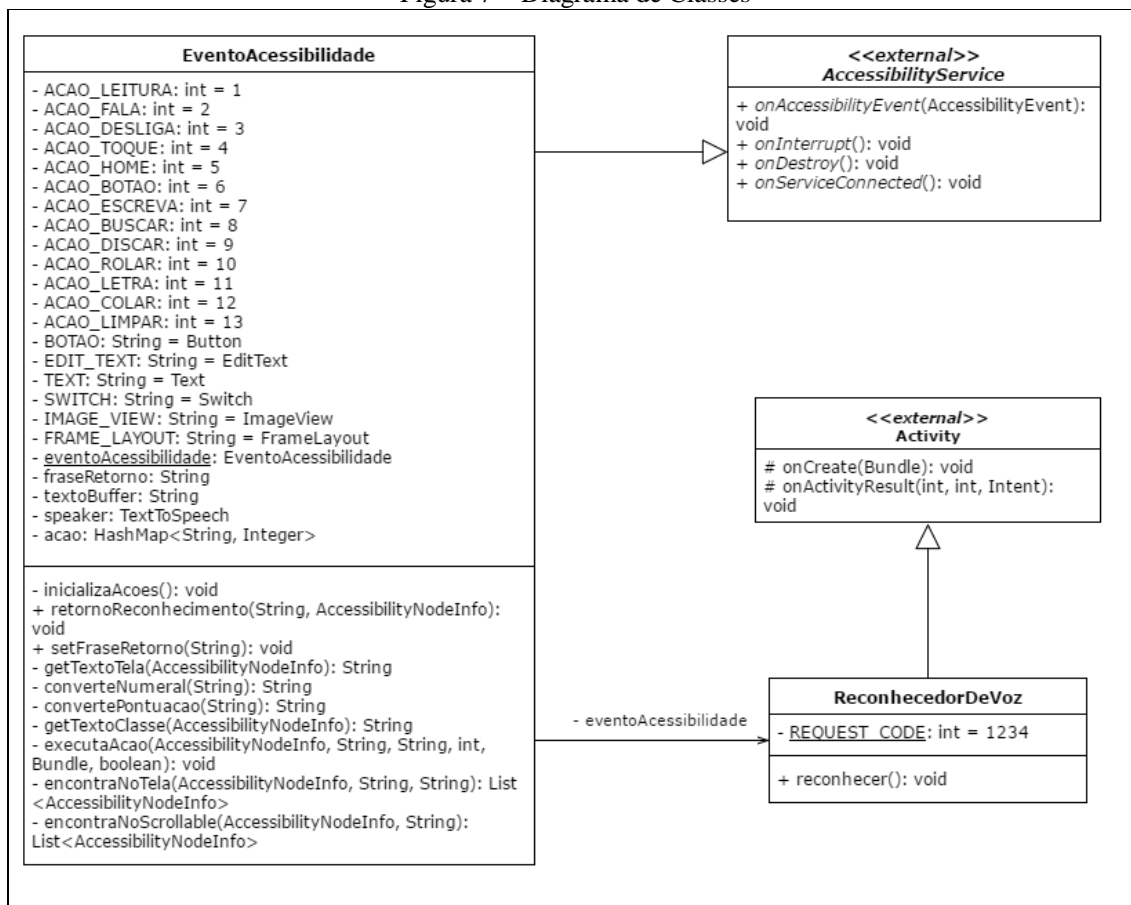


De acordo com a Figura 6, a aplicação possui apenas um ator, o *Usuário*. Este ator é responsável por realizar todas as treze ações existentes na aplicação. Cada ação resulta em uma interação diferente com o *smartphone*. Os casos de uso UC01: Ação ler e UC13: Ação falar resultam na exteriorização de voz. Os casos de uso UC02: Ação tocar, UC05: Ação botão e UC07: Ação buscar geram interações de toque em tela. Com objetivos mais específicos como parar a execução da aplicação, voltar para a tela inicial e realizar a rolagem de tela, existem os casos de uso UC03: Ação desligar, UC04: Ação home e UC09: Ação rolar respectivamente. Os demais casos de uso estão relacionados à digitação em tela e edição de campos de texto. A explicação detalhada do que é realizado por cada ação encontra-se mais adiante, na subseção 3.3.6.

3.2.2 Diagrama de Classes

Para a implementação da aplicação foram desenvolvidas duas classes, sendo elas *EventoAcessibilidade* e *ReconhecedorDeVoz*. A Figura 7 mostra o diagrama de classes da aplicação.

Figura 7 – Diagrama de Classes



A classe `EventoAcessibilidade` é uma implementação da classe abstrata `AccessibilityService` que pertence à biblioteca `android.accessibilityservice`. Ela é responsável por receber os eventos de acessibilidade do sistema operacional e realizar a tratativa dos mesmos. Também é de sua responsabilidade a exteriorização de voz, que ocorre através do atributo `speaker`, que é uma instância da classe `TextToSpeech`, pertencente à biblioteca `android.speech.tts`. Nela também está contido o mapa de verbos e ações utilizado durante o processo de tratativa da frase reconhecida. Este processo é detalhado na subseção 3.3.6.

A classe `ReconhecedorDeVoz` estende da classe `Activity` e é responsável por realizar o reconhecimento das frases ditas pelo usuário. O reconhecimento ocorre através do método `reconhecer`, onde é realizada uma chamada à classe `RecognizerIntent`, pertencente a biblioteca `android.speech`. Ao final do processo de reconhecimento, o resultado é retornado para a classe `EventoAcessibilidade`, através de uma referência singleton. O detalhamento deste processo está contido na subseção 3.3.5.

3.3 IMPLEMENTAÇÃO

A seguir são mostradas as técnicas e ferramentas utilizadas para o desenvolvimento da aplicação. São apresentadas também as principais rotinas envolvendo o reconhecimento de voz e a tomada de ações.

3.3.1 Técnicas e ferramentas utilizadas

A aplicação é constituída de um serviço que é executado ininterruptamente em segundo plano e de uma *activity*. O serviço tem os papéis de iniciar a *activity* no momento em que ocorrer um toque na tela, de identificar e executar a ação desejada pelo usuário. A *activity* é responsável por realizar o reconhecimento de voz e devolver para o serviço o resultado do reconhecimento. Tanto o serviço quanto a *activity* foram desenvolvidos na linguagem Java, utilizando a IDE Android Studio 1.4. As principais bibliotecas usadas no desenvolvimento do serviço foram: `accessibilityservice`, `accessibility` e `speech.tts` do Android. A principal biblioteca utilizada no desenvolvimento da *activity* foi a `android.speech`.

A biblioteca `android.accessibilityservice` é usada para o desenvolvimento de serviços de acessibilidade que provêm formas alternativas ou reforçadas de comunicação com o usuário (ANDROID, 2015a). Esta biblioteca é composta por duas classes, sendo elas: `AccessibilityService` e `AccessibilityServiceInfo`. A classe `AccessibilityService` executa em segundo plano e recebe chamadas do sistema operacional, que se referem a eventos, como por exemplo, um botão pressionado ou uma mudança de foco na tela. A classe `AccessibilityServiceInfo` tem a função de especificar quais tipos de eventos devem ser notificados pelo sistema operacional para a classe `AccessibilityService` (ANDROID, 2015a).

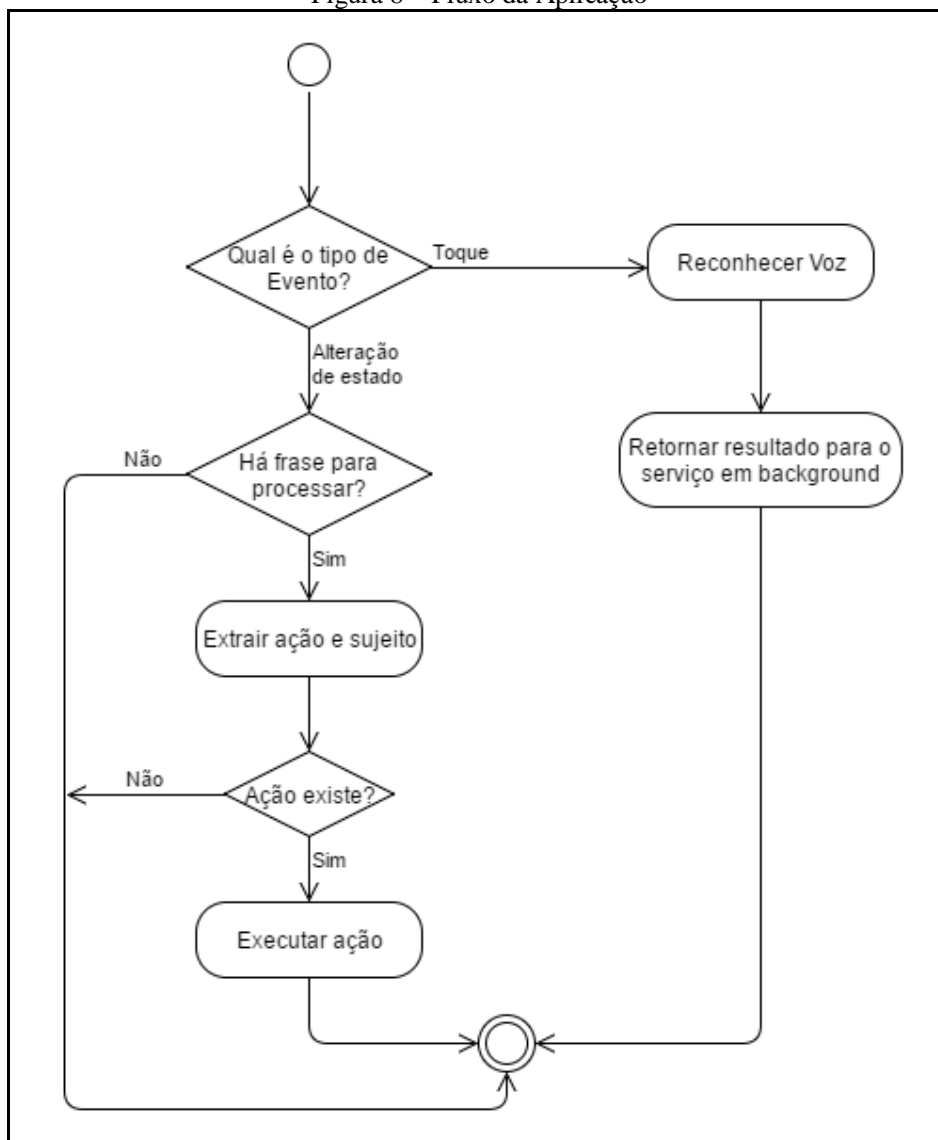
A biblioteca `android.speech` é utilizada para implementação de serviços de reconhecimento de fala (ANDROID, 2015b). Dentre as classes desta biblioteca, destaca-se a classe `SpeechRecognizer` que é responsável por prover os serviços de reconhecimento de fala. A biblioteca cria um canal de comunicação com o serviço de reconhecimento de fala, que é responsável por transformar a voz em texto. Para que seja possível criar este canal, é necessária a conexão com a internet. Porém, não é recomendado que se faça um reconhecimento de forma contínua através desta biblioteca, pois isso implica em um alto consumo de bateria e banda (ANDROID, 2015c). Através desta biblioteca é possível também tomar ações sobre a fala do usuário, como por exemplo, disparar uma busca na internet com o que foi dito (ANDROID, 2015b).

Através da biblioteca `android.speech.tts` é possível implementar síntese de voz (ANDROID, 2015d). A principal classe desta biblioteca é a `TextToSpeech` sendo responsável por transformar um texto em voz, para execução imediata ou ainda armazenamento em um arquivo de áudio (ANDROID, 2015d). Outra classe importante desta biblioteca é a `Voice`, através dela são configuradas as características da voz gerada, como timbre e qualidade (ANDROID, 2015e).

3.3.2 Fluxo geral da aplicação

Para facilitar a compreensão da estrutura da aplicação foi elaborado o diagrama de atividades da Figura 8. Este diagrama representa os dois principais fluxos da aplicação, sendo eles o reconhecimento da voz do usuário e a tratativa da frase reconhecida.

Figura 8 – Fluxo da Aplicação



Os fluxos exibidos na Figura 8 são executados de forma encadeada, pois, ao final do processo de reconhecimento de voz é disparado um novo evento que inicia o fluxo de tratamento de frases. O fluxo de reconhecimento de voz é responsável por realizar o reconhecimento e devolver o resultado para o que mesmo seja tratado. O fluxo de tratativa da frase reconhecida faz a extração da ação e do sujeito da frase reconhecida, para que então seja aplicada essa ação sob esse sujeito.

Nas próximas seções são detalhadas cada uma das etapas do fluxo apresentado acima. A seção 3.3.3 contém os parâmetros de configuração da aplicação. A seção 3.3.4 apresenta a tratativa de eventos utilizada na aplicação. A seção 3.3.5 exhibe a metodologia de reconhecimento de voz. A seção 3.3.6 detalha o tratamento da frase reconhecida. Por fim, a seção 3.3.7 mostra a operacionalidade da implementação.

3.3.3 Configuração da aplicação

O Android possui uma série de recursos de acessibilidade que podem ser utilizados pelas aplicações. Para que uma aplicação possa utilizar esses recursos é necessário determinar alguns parâmetros em seu arquivo de configuração. O Quadro 2 mostra os parâmetros utilizados na aplicação desenvolvida.

Quadro 2 – Parâmetros de configuração

```

1 <accessibility-service
2   xmlns:android="http://schemas.android.com/apk/res/android"
3   android:accessibilityEventTypes="typeAllMask"
4   android:accessibilityFlags="flagReportViewIds|flagRequestTouchExplorationMode"
5   android:accessibilityFeedbackType="feedbackSpoken"
6   android:notificationTimeout="100"
7   android:canRetrieveWindowContent="true"
8   android:canRequestTouchExplorationMode="true"
8 />

```

O parâmetro existente na linha 2 (Quadro 2) define que a aplicação seja comunicada de todos os eventos de acessibilidade que venham a ocorrer. Na linha 3 são determinadas as *flags* para que a aplicação receba do sistema operacional os *ids* das *views* e para que o sistema operacional funcione no modo exploração por toque. Esse modo faz com que, por exemplo, para abrir ou selecionar itens seja necessário realizar um toque duplo, ao invés de um único toque como é de praxe. Na sequência são determinados o tipo de *feedback* dado pela aplicação, no caso falado (linha 4), tempo de *timeout* (linha 5), permissões para recuperar o conteúdo da tela (linha 6) e alterar o sistema operacional para o modo exploração por toque (linha 7).

3.3.4 Tratativa de Eventos

A classe `EventoAcessibilidade` possui o método `onAccessibilityEvent` que é invocado pelo sistema operacional toda vez que ocorre um evento de acessibilidade, como por exemplo, um toque na tela ou uma alteração no conteúdo da tela, conforme mostra o Quadro 3.

Quadro 3 – Tratamento de Evento

```

1  public void onAccessibilityEvent(AccessibilityEvent event) {
2      final int tipoEvento = event.getEventType();
3      switch (tipoEvento){
4          case AccessibilityEvent.TYPE_TOUCH_EXPLORATION_GESTURE_START:
5              if (speaker.isSpeaking()){
6                  speaker.stop();
7              }
8              Intent intent = new Intent();
9              intent.setClass(this, ReconhecedorDeVoz.class);
10             intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
11             startActivity(intent);
12             break;
13             case AccessibilityEvent.TYPE_WINDOW_STATE_CHANGED:
14                 if (event.getSource() != null){
15                     if (this.fraseRetorno != null &&
16                         (!event.getPackageName().equals(this.getPackageName()))){
17                         this.retornoReconhecimento(fraseRetorno, event.getSource());
18                     }
19                 }
20             break;

```

A partir do Quadro 3 é possível observar que a aplicação trata os eventos do tipo toque na tela (linha 4) e alteração de estado da tela (linha 13). Ao ocorrer um evento de toque na tela a aplicação inicia uma atividade para o reconhecimento de voz. A *activity* `ReconhecedorDeVoz` retornará para o serviço o texto reconhecido através do atributo estático `fraseRetorno`. Quando a *activity* de reconhecimento de voz for finalizada, o sistema operacional emitirá eventos de alteração do estado da tela, que serão capturados pelo método `onAccessibilityEvent`. Caso haja uma frase de retorno para ser processada e o evento não pertença ao *package* desta aplicação (linha 15), será invocado o método `retornoReconhecimento` para realizar o tratamento da frase dita pelo usuário. O método recebe como parâmetros a frase a ser tratada e o nó raiz da tela.

3.3.5 Reconhecimento de voz

Sempre que a *activity* `ReconhecedorDeVoz` for iniciada, será executado o método `reconhecer`. Este método é responsável por iniciar o serviço de reconhecimento de voz, disponibilizado pela biblioteca `android.speech`, através da classe `RecognizerIntent`, como pode ser visto no Quadro 4.

Quadro 4 – Reconhecimento de voz

```

1 public void reconhecer(){
2     Intent intent = new
      Intent(RecognizerIntent.ACTION_RECOGNIZE_SPEECH);
3     intent.putExtra(RecognizerIntent.EXTRA_LANGUAGE_MODEL,
      RecognizerIntent.LANGUAGE_MODEL_FREE_FORM);
4     intent.putExtra(RecognizerIntent.EXTRA_PROMPT, "Reconhecedor de
      Voz");
5     intent.putExtra(RecognizerIntent.EXTRA_MAX_RESULTS, 1);
6     startActivityForResult(intent, REQUEST_CODE);
7 }

```

Na linha 3 é definida a modalidade de linguagem a ser reconhecida. Optou-se pela modalidade *free*, que faz com que o serviço de reconhecimento trabalhe paralelamente com todas as linguagens suportadas. Também são definidos o título de exibição (linha 4) e a quantidade máxima de resultados a serem retornados (linha 5).

Ao final do reconhecimento, o resultado é devolvido para a classe *ReconhecedorDeVoz* no método *onActivityResult*, conforme pode ser visto no Quadro 5. Através de uma referência *singleton*, é repassado o resultado do reconhecimento para o serviço em *background* (linha 4). Por fim, na linha 7 a *activity* é finalizada, encerrando a etapa de reconhecimento de voz. Neste mesmo momento, são disparados eventos de alteração do estado da tela pelo sistema operacional.

Quadro 5 – Resultado do reconhecimento de voz

```

1 protected void onActivityResult(int requestCode, int resultCode,
      Intent data){
2     if (requestCode == REQUEST_CODE && resultCode == RESULT_OK){
3         ArrayList<String> resultados = data.getStringArrayListExtra(
      RecognizerIntent.EXTRA_RESULTS);
4         eventoAcessibilidade.setFraseRetorno(resultados.get(0));
5     }
6     super.onActivityResult(requestCode, resultCode, data);
7     this.finish();
8 }

```

Os eventos disparados serão capturados na classe *EventoAcessibilidade*. Quando um evento que atenda as condições estabelecidas, conforme exibe o Quadro 3, for capturado, será então executado o método *retornoReconhecimento*. Este método é responsável por fazer a tratativa da frase reconhecida.

3.3.6 Tratamento da frase reconhecida

O primeiro passo realizado no método *retornoReconhecimento* é a extração do sujeito e do verbo da frase reconhecida. A Figura 9 apresenta a convenção de estrutura estabelecida para as frases reconhecidas pela aplicação.

Figura 9 – Estrutura da frase



De acordo com a estrutura exibida na Figura 9, a primeira palavra da frase representa o verbo e a última palavra representa o sujeito. As palavras intermediárias, isto é, que não são a primeira nem a última da frase, podem ou não serem utilizadas, variando de acordo com o verbo. O Quadro 6 mostra a extração do sujeito (linha 6) e da ação (linha 9).

Quadro 6 – Encontrando ação e sujeito

```

1 public void retornoReconhecimento(String retorno,
  AccessibilityNodeInfo nodeInfo){
2     if(retorno == null){
3         return;
4     }
5     String[] frase = retorno.split(" ");
6     String sujeito = frase[frase.length - 1];
7     int acao;
8     try{
9         acao = acoes.get(frase[0].toLowerCase());
10    } catch (Exception e){
11        acao = 999;
12    }

```

Assim que é extraído o verbo da frase, conforme exibe o Quadro 6 na linha 9, busca-se em um mapa de ações, qual é o identificador de ação daquele verbo. O mapa de ações determina os diferentes verbos que resultam em uma ação na aplicação. Por exemplo, as palavras “diga” e “fale” resultam na ação falar. O apêndice B mostra a relação de todos os verbos reconhecidos pela aplicação e de suas respectivas ações.

3.3.6.1 Ações de exteriorização de voz

Depois de identificada a ação, parte-se para a execução da mesma. Cada ação usa a frase reconhecida de forma particular, utilizando as partes da frase que são necessárias. No Quadro 7 é exibida a estrutura da ação responsável por realizar a leitura dos textos presentes na tela.

Quadro 7 – Ação leitura

```

1  case ACAO_LEITURA:
2      String textoTela = "";
3      textoTela = getTextoTela(nodeInfo);
4      speaker.speak(textoTela, TextToSpeech.QUEUE_FLUSH, null);
5      break;

```

Para carregar todos os textos da tela, é realizada uma chamada ao método `getTextoTela` (Quadro 7) passando como parâmetro o nó raiz da tela (linha 3). Este método fará uma busca recursiva de todos os textos relacionados ao nó raiz. O texto retornado é então passado para o objeto `speaker`, para que seja exteriorizado através do sintetizador de voz.

A ação fala tem o intuito de repetir uma frase dita pelo usuário. Nesta ação não é aplicado o conceito de sujeito, pois é uma repetição das palavras ditas, conforme mostra o Quadro 8.

Quadro 8 – Ação fala

```

1  case ACAO_FALA:
2      String fraseSolicitada = "";
3      for (int i = 1; i < frase.length; i++){
4          fraseSolicitada = fraseSolicitada.concat(frase[i]).concat(" ");
5      }
6      speaker.speak(fraseSolicitada, TextToSpeech.QUEUE_FLUSH, null);
7      break;

```

A frase reconhecida fica armazenada dentro do *array* `frase` (Quadro 8). Posteriormente, é realizada uma varredura a partir da posição “1”, para que seja montada uma nova frase sem a primeira palavra da frase original, que representa o verbo. Desta forma, a frase exteriorizada pela aplicação não contém o verbo da frase original. Por exemplo, quando a frase original for “diga boa tarde” a saída será “boa tarde”.

3.3.6.2 Ações de toque

Para realizar interações com os componentes de tela, a aplicação faz uso das *actions* disponibilizadas pela biblioteca `android.accessibility`. Através delas é possível aplicar, por exemplo, toque nos componentes de tela. Cada componente pode receber um grupo limitado de ações e nem todas as ações são aplicáveis a todos os componentes. Por exemplo, não se pode aplicar uma ação do tipo *scroll* em um componente do tipo botão. O Quadro 9 mostra uma chamada do método `executaAcao` para aplicação de uma ação de toque.

Quadro 9 – Ação toque

```

1  case ACAO_TOQUE:
2      executaAcao(nodeInfo, sujeito, null,
3      AccessibilityNodeInfo.ACTION_CLICK, null, false);
4      break;

```

Os parâmetros passados na linha 2 (Quadro 9) representam respectivamente, o nó raiz da tela, o nome do componente de tela que deve receber a ação, o filtro de tipo de

componente, a ação que deve ser executada, os argumentos da ação e se a ação pode ser aplicada apenas em componentes focáveis. Os detalhes de execução do método `executaAcao` são exibidos no Quadro 20.

A ação botão é similar a ação toque vista no Quadro 9, aplicando da mesma forma um toque sobre um determinado componente de tela. Porém, nesta ação é passada a constante `BOTAO` no filtro de tipo de componente de tela. Isto faz com que apenas componentes da tela do tipo botão sejam considerados. O Quadro 10 exibe a chamada do método `executaAcao` da ação botão.

Quadro 10 – Ação botão

```
1 case ACAO_BOTAO:
2   executaAcao(nodeInfo, sujeito, BOTAO,
   AccessibilityNodeInfo.ACTION_CLICK, null, false);
3   break;
```

Entretanto, como a tela pode conter mais de um componente com o mesmo nome, foi desenvolvida essa estratégia de filtrar os componentes de acordo com o seu tipo. Dessa forma foi possível eliminar a ambiguidade na busca de componentes da tela. Em um cenário onde exista um campo de texto com o texto “3”, um botão com o nome “3” e o sujeito da frase seja “3”, a ação toque será aplicada no primeiro componente encontrado na busca, enquanto a ação botão será aplicada no primeiro componente do tipo botão encontrado na busca.

Da mesma forma que a ação botão filtra os componentes de tela, a ação buscar também faz uso dessa estratégia, porém utilizando o tipo `image_view`. O Quadro 11 mostra os parâmetros utilizados neste caso.

Quadro 11 – Ação buscar

```
1 case ACAO_BUSCAR:
2   executaAcao(nodeInfo, sujeito, IMAGE_VIEW,
   AccessibilityNodeInfo.ACTION_CLICK, null, false);
3   break;
```

O principal motivo para o desenvolvimento dessa ação foi o aplicativo Whatsapp. O mesmo, na sua lista de seleção de conversas, possui um componente do tipo `image_view`, onde é necessário que este componente receba a ação de toque para iniciar a conversa.

3.3.6.3 Ação de parada

A fim de fazer com que o aparelho volte a funcionar de maneira normal, foi desenvolvida uma ação para inativar a aplicação. No Quadro 12 pode ser vista a estrutura desta ação. O serviço em *background* não é parado de fato, mas é configurado de tal forma para que não tenha nenhum efeito no comportamento do sistema operacional.

Quadro 12 – Ação desliga

```

1  case ACAO_DESLIGA:
2      AccessibilityServiceInfo info = new AccessibilityServiceInfo();
3      info.flags = AccessibilityServiceInfo.DEFAULT;
4      info.feedbackType = AccessibilityServiceInfo.FEEDBACK_SPOKEN;
5      info.eventTypes = AccessibilityEvent.TYPE_VIEW_CLICKED;
6      info.notificationTimeout = 100;
7      this.setServiceInfo(info);
8      speaker.speak("Estou desligando, até mais",
9      TextToSpeech.QUEUE_FLUSH, null);
10     break;

```

Ao definir o atributo `flags` como `default` na linha 3 (Quadro 12), o sistema operacional para de se comportar no modo de exploração por toque, voltando ao modo padrão. Definindo o atributo `eventTypes` como `type_view_clicked` (linha 5), a aplicação passa a receber apenas eventos desse tipo, que não é um tipo tratado pela aplicação. Dessa forma, os eventos são ignorados e a aplicação não tem mais nenhum efeito prático. Para que a aplicação volte a funcionar, basta reiniciá-la através do menu de acessibilidade do aparelho.

3.3.6.4 Ação de retorno

Atualmente, os botões “voltar”, “home” e “aplicativos recentes” são virtuais. Porém, até pouco tempo atrás, estes botões eram físicos nos aparelhos com sistema operacional Android. Devido a essa característica, esses botões ficam inacessíveis através da hierarquia de tela. Para suprir essa necessidade foi desenvolvida a ação *home* exibida no Quadro 13, que realiza a mesma função do botão “home”.

Quadro 13 – Ação home

```

1  case ACAO_HOME:
2      Intent startMain = new Intent(Intent.ACTION_MAIN);
3      startMain.addCategory(Intent.CATEGORY_HOME);
4      startMain.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
5      startActivity(startMain);
6      break;

```

Para realizar a função do botão “home” é criada uma *intent* (linha 2 - Quadro 13), que faz com que o sistema operacional volte a tela de início. Esta *intent* é executada através do método `startActivity` (linha 5).

3.3.6.5 Ações para digitação

Uma das formas de interação da aplicação com campos de texto se dá através da ação *escreva*. Nesta ação, além da primeira e da última palavra da frase, também são utilizadas as palavras intermediárias. As mesmas tornam-se o texto a ser escrito em um determinado campo. Por exemplo, se a frase reconhecida for “escreva boa tarde mensagem”, a aplicação

escreverá o texto “boa tarde” em um campo da tela chamado “mensagem”. O Quadro 14 mostra a lógica de execução desta ação.

Quadro 14 – Ação escreva

```

1  case ACAO_ESCREVA:
2      String fraseEditText = "";
3      for (int i = 1; i < frase.length - 1; i++){
4          fraseEditText = fraseEditText.concat(frase[i]).concat(" ");
5      }
6      ClipboardManager clipboard = (ClipboardManager)
7      getSystemService(Context.CLIPBOARD_SERVICE);
8      ClipData clip = ClipData.newPlainText("label", fraseEditText);
9      clipboard.setPrimaryClip(clip);
10     executaAcao(nodeInfo, sujeito, TEXT,
11     AccessibilityNodeInfo.ACTION_FOCUS, null, true);
12     executaAcao(nodeInfo, sujeito, TEXT,
13     AccessibilityNodeInfo.ACTION_PASTE, null, true);
14     break;

```

Como pode ser visto no Quadro 14, usa-se um laço de repetição (linha 3) para extrair do *array* frase, as palavras intermediárias da frase reconhecida, ou seja, que não são nem a primeira nem a última da frase. Então esse texto extraído é passado para o *clipboard* (linha 7) para que possa ser colado no campo. Aplica-se então foco no campo desejado (linha 9) para que finalmente possa ser colado o texto no mesmo (linha 10). A aplicação do foco se faz necessária, devido ao fato que a biblioteca não permite que se colem textos em campos que não detenham o foco.

Visando a realização de ligações, foi desenvolvido a ação discar. O conceito padrão de sujeito não é aplicado nesta ação. Nela todas as palavras ditas após o verbo são consideradas como sujeitos. Por exemplo, na frase “discar asterisco oito zero zero zero”, serão considerados cinco sujeitos e um evento de toque será aplicado em cada um dos cinco. Optou-se por realizar a ação discar desta forma, pois o teclado virtual não respeita o modo exploração por toque. O Quadro 15 exhibe a lógica de execução da ação.

Quadro 15 – Ação discar

```

1  case ACAO_DISCAR
2      String numeroDiscar = "";
3      for (int i = 1; i < frase.length; i++){
4          numeroDiscar += converteNumeral(frase[i]);
5      }
6      char[] numeros = numeroDiscar.toCharArray();
7      executaAcao(nodeInfo, "espaço", BOTAO,
AccessibilityNodeInfo.ACTION_LONG_CLICK, null, false);
8      for (int i = 0; i < numeros.length; i++){
9          if (numeros[i] != '-' && numeros[i] != ' '){
10             executaAcao(nodeInfo, String.valueOf(numeros[i]), FRAME_LAYOUT,
AccessibilityNodeInfo.ACTION_CLICK, null, false);
11         }
12     }
13     executaAcao(nodeInfo, "discar", BOTAO,
AccessibilityNodeInfo.ACTION_CLICK, null, false);
14     break;

```

A biblioteca de reconhecimento de voz, em alguns casos, retorna os números em forma cardinal. Por esta razão, a primeira tratativa a ser feita, conforme exibe o Quadro 15, é converter esses números de cardinais para numerais (linha 4). A conversão é realizada devido ao fato de que os botões do teclado virtual estão no formato numeral. Após a conversão, é aplicado um toque longo no botão “espaço” do teclado virtual (linha 7), garantindo assim que não haverá nenhum número previamente digitado no teclado. Então se aplicam os eventos de toque nos botões do teclado que correspondem aos números ditos (linha 10). Por fim, aplica-se um evento de toque no botão “discar” (linha 13) para iniciar a chamada.

3.3.6.6 Ação para rolagem

A ação rolar tem o objetivo de fazer o movimento de rolagem da tela. Nesta ação existem apenas dois sujeitos possíveis, sendo eles “frente” e “trás”. Estes implicam respectivamente, no movimento de rolagem para frente e para trás. O Quadro 16 mostra a estrutura desta ação.

Quadro 16 – Ação rolar

```

1  case ACAO_ROLAR:
2      List<AccessibilityNodeInfo> nos = encontraNoScrollable(nodeInfo,
3      null);
4      if (nos.size() > 0){
5          for (int i = 0; i < nos.size(); i++){
6              if("frente".equalsIgnoreCase(sujeito)){
7                  nos.get(i).performAction(AccessibilityNodeInfo.ACTION_SCROLL_FORWARD);
8              } else if ("trás".equalsIgnoreCase(sujeito)){
9                  nos.get(i).performAction(AccessibilityNodeInfo.ACTION_SCROLL_BACKWARD);
10             }
11         }
12     break;

```

A partir do nó raiz da tela, é realizada uma chamada ao método `encontraNoScrollable` na linha 2 (Quadro 16). Este método retorna todos os componentes ligados ao nó raiz que podem executar uma ação de rolagem. Então de acordo com o sujeito é realizada a rolagem para frente (linha 6) ou para trás (linha 8). Decidiu-se por fazer a rolagem de todos os componentes roláveis em paralelo, pois no momento da leitura da tela, se tem a impressão de que as diferentes listas são colunas, uma ao lado da outra. Por exemplo, a aplicação Whatsapp contém três listas, exibindo chamadas, conversas e contatos, uma ao lado da outra. Ao fazer a rolagem em paralelo, se mantém essa impressão de colunas de texto.

3.3.6.7 Ações de ditado

Alguns textos podem ser complexos de serem escritos através do reconhecimento de frases contínuas, como por exemplo, endereços de e-mail. A ação letra visa facilitar a elaboração destes textos, através de uma estratégia de ditado. Esta estratégia foi inspirada na expressão comum “n de navio”. Nesta ação, a primeira letra do sujeito é concatenada em uma variável de texto e armazenada em memória para ser utilizada posteriormente. No Quadro 17 pode ser vista a lógica de execução da ação letra.

Quadro 17 – Ação letra

```

1  case ACAO_LETRA:
2      if (frase.length > 1){
3          speaker.speak(frase[1], TextToSpeech.QUEUE_FLUSH, null);
4          frase[1] = converteNumeral(frase[1]);
5          frase[1] = convertePontuacao(frase[1]);
6          char[] palavra = frase[1].toCharArray();
7          textoBuffer =
8          textoBuffer.concat(String.valueOf(palavra[0]).toLowerCase());
9      }
10     break;

```

Para que se tenha certeza de qual letra será armazenada, conforme exhibe o Quadro 17, exterioriza-se a palavra reconhecida (linha 3). Caso a palavra reconhecida seja um número ou um sinal de pontuação, a mesma será convertida para sua representação simbólica (linhas 4 e

5). Por exemplo, na frase “letra ponto”, será concatenado na variável `textoBuffer` o texto “.”. Ou ainda, caso a frase seja “letra nove”, será concatenado o texto “9”. Em um cenário onde a frase seja “letra pato”, será concatenado o texto “p”.

Da mesma forma que existe a ação `letra` para adicionar textos na variável `textoBuffer`, a ação `limpar` tem o objetivo de tirar textos dessa mesma variável. No Quadro 18 é exibida a estrutura da ação `limpar`.

Quadro 18 – Ação `limpar`

```

1  case ACAO_LIMPAR:
2    if ("tudo".equalsIgnoreCase(sujeito)){
3      textoBuffer = "";
4    } else if ("letra".equalsIgnoreCase(sujeito)){
5      if (textoBuffer.length() > 0){
6        textoBuffer = textoBuffer.substring(0, textoBuffer.length() - 1);
7      }
8    }
9    break;

```

A ação `limpar` tem apenas dois sujeitos possíveis, sendo eles “tudo” e “letra”. Conforme pode ser visto no Quadro 18, o sujeito “tudo” faz com que a variável `textoBuffer` seja inicializada (linha 3). Já o sujeito “letra” faz com que seja removido o último caractere contido nesta mesma variável.

Para que seja possível colar o texto contido na variável `textoBuffer` nos componentes de tela, foi desenvolvida a ação `colar`. O Quadro 19 exibe a estrutura desta ação.

Quadro 19 – Ação `colar`

```

1  case ACAO_COLAR:
2    ClipboardManager clipboardColar = (ClipboardManager)
3    getSystemService(Context.CLIPBOARD_SERVICE);
4    ClipData clipColar = ClipData.newPlainText("label", textoBuffer);
5    clipboardColar.setPrimaryClip(clipColar);
6    executaAcao(nodeInfo, sujeito, TEXT,
7    AccessibilityNodeInfo.ACTION_FOCUS, null, true);
8    executaAcao(nodeInfo, sujeito, TEXT,
9    AccessibilityNodeInfo.ACTION_PASTE, null, true);
10   break;

```

Assim como foi feito na ação `escreva` e, como pode ser visto no Quadro 19, o texto a ser escrito é passado para o `clipboard` (linha 3), executa-se a ação de foco (linha 5) e por fim, cola-se o texto no componente da tela. É importante ressaltar que nesta ação, não são usadas palavras intermediárias, diferentemente da ação `escreva`.

3.3.6.8 Métodos auxiliares

Visando o reaproveitamento de código, criou-se o método `executaAcao` para aplicação das ações recebidas por parâmetro nos componentes da tela. O mesmo se encarrega de

encontrar os componentes e aplicar sob eles a ação desejada. Sua estrutura é exibida no Quadro 20.

Quadro 20 – Método executaAcao

```

1 List<AccessibilityNodeInfo> nos = encontraNoTela(nodeInfo, sujeito,
  tipoComponente);
2 AccessibilityNodeInfo no = null;
3 if (nos != null){
4   try{
5     if (focus){
6       for (int i = 0; i < nos.size();i++) {
7         if (nos.get(i).isFocusable()){
8           no = nos.get(i);
9           i = nos.size();
10        }
11      }
12    } else {
13      no = nos.get(0);
14    }
15    if (no != null){
16      if (arguments == null){
17        no.performAction(action);
18      } else {
19        no.performAction(action, arguments);
20      }
21    }
22  } catch (IndexOutOfBoundsException indexEx) {
23    speaker.speak("Não encontrei na tela " + sujeito,
  TextToSpeech.QUEUE_FLUSH, null);
24  } catch (Exception ex){
25    speaker.speak("Erro de tela desconhecido",
  TextToSpeech.QUEUE_FLUSH, null);
26 }

```

Para encontrar os componentes desejados, como pode ser visto no Quadro 20, é realizada uma chamada ao método `encontraNoTela` (linha 1). A estrutura deste método é mostrada no Quadro 21. Caso tenha sido encontrado algum nó, a partir dos parâmetros passados, será selecionado um nó para receber a ação. Do contrário, será exteriorizado o texto “não encontrei na tela” e o sujeito da frase (linha 23). Quando o parâmetro `focus` for verdadeiro, será selecionado o primeiro nó que pode receber foco da lista de nós retornados (linhas 6 a 9). Já se este parâmetro for falso, será selecionado o primeiro nó da lista (linha 13). Por fim, é executada a ação sob o nó selecionado sem argumentos adicionais (linha 17) ou com argumentos adicionais (linha 19).

O objetivo do método `encontraNoTela` é retornar uma lista com todos os nós da tela que contém o texto recebido por parâmetro. Para tal, é realizada uma busca recursiva em todos os nós, a partir do nó raiz, conforme exhibe o Quadro 21.

Quadro 21 – Método encontraNoTela

```

1  if (nodeInfo == null){
2      return nos;
3  }
4  if (nodeInfo.getChildCount() > 0) {
5      for (int i = 0; i < nodeInfo.getChildCount(); i++) {
6          textoNo = "";
7          if (nodeInfo.getChild(i) != null){
8              if (nodeInfo.getChild(i).getText() != null){
9                  textoNo += nodeInfo.getChild(i).getText().toString();
10             }
11             if (nodeInfo.getChild(i).getContentDescription() != null){
12                 textoNo +=
nodeInfo.getChild(i).getContentDescription().toString();
13             }
14             textoNo = textoNo.toLowerCase();
15             if (textoNo.contains(texto) || texto.equals("")){
16                 if (tipoComponente == null){
17                     nos.add(nodeInfo.getChild(i));
18                 } else if
(nodeInfo.getChild(i).getClassName().toString().contains(tipoComponente))){
19                     nos.add(nodeInfo.getChild(i));
20                 }
21             }
22             nos.addAll(encontraNoTela(nodeInfo.getChild(i), texto,
tipoComponente));
23         }
24     }
25 }
26 return nos;

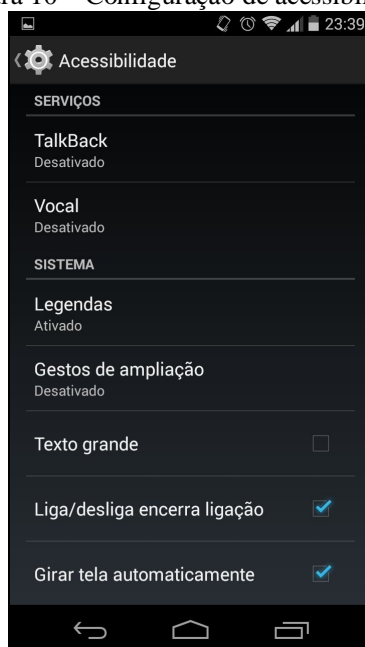
```

De acordo com o Quadro 21, quando o nó corrente atender as condições (linha 1 e linha 4), ocorrerá uma iteração em todos os seus filhos (linha 5). Serão então concatenados os textos presentes no nó filho na variável `textoNo` (linhas 9 e 12). Então, é comparado o valor contido nesta variável com o texto recebido por parâmetro (linha 15). Se o texto recebido por parâmetro for vazio ou a variável `textoNo` contiver o mesmo e o nó corresponder ao tipo de componente desejado, ele será então adicionado a lista `nos` (linhas 16 a 20). O mesmo procedimento será repetido nos nós filhos, através da chamada recursiva ao método (linha 22). Por fim, é retornada a lista de nós que atendem as condições (linha 26).

3.3.7 Operacionalidade da implementação

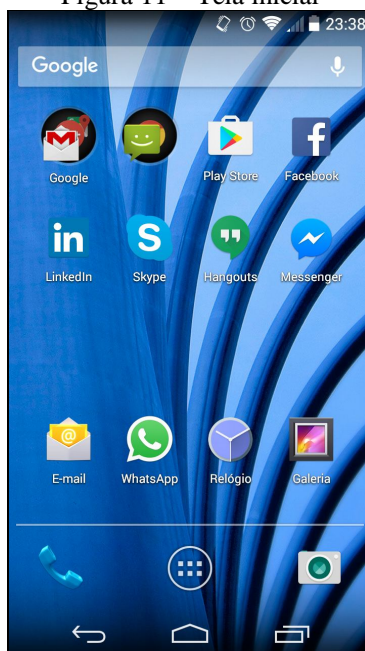
Após a instalação da aplicação, a mesma ficará disponível no menu de configurações do dispositivo, no item acessibilidade. Para que a aplicação entre em funcionamento é necessário ativá-la através deste menu. A Figura 10 exibe a aplicação aguardando ativação.

Figura 10 – Configuração de acessibilidade



Após a ativação, a aplicação permanecerá executando em segundo plano. Não há uma diferença visível na tela durante a execução. A Figura 11 mostra a tela inicial do aparelho enquanto é executada a aplicação em segundo plano.

Figura 11 – Tela inicial



Ao ocorrer um toque na tela, é iniciado o serviço de reconhecimento de voz. Este serviço abre uma tela, conforme exibe a Figura 12. Ao final do processo de reconhecimento, esta tela será fechada, fazendo com que o dispositivo volte para a tela anterior.

Figura 12 – Tela de reconhecimento de voz



Assim que ocorrer o retorno para a tela anterior, será executada a ação solicitada pelo usuário. Geralmente, o primeiro comando a ser dado será para ler a tela, através do comando “ler tela”, para que o usuário possa identificar o que está sendo exibido. Realizando a leitura da tela inicial serão falados os nomes dos componentes presentes na tela, entre eles o componente “aplicativos”. Com o usuário realizando um novo toque na tela e dizendo a frase “toque em aplicativos”, o resultado será a abertura do menu de aplicativos, conforme exibe a Figura 13a.

Figura 13 – Interação com aplicações



Realizando um novo toque na tela, será iniciado novamente o serviço de reconhecimento de voz. O usuário poderá então efetuar novamente a leitura da tela, para identificar quais aplicativos estão visíveis. Uma vez identificada que a aplicação “calculadora” está presente na tela, pode ser dita a frase “toque em calculadora”, que fará com que essa aplicação seja aberta, conforme exibe a Figura 13b.

3.4 RESULTADOS E DISCUSSÕES

Nesta seção são apresentados os testes realizados com o aplicativo. A seção 3.4.1 apresenta o experimento de usabilidade com pessoas sem deficiência visual, detalhando a metodologia utilizada e a análise dos resultados. A seção 3.4.2 relata o experimento com uma pessoa portadora de deficiência visual. Por fim, a seção 3.4.3 traz um comparativo entre os trabalhos correlatos e o trabalho desenvolvido.

3.4.1 Experimento 01: Avaliação de usabilidade com pessoas sem deficiência visual

O experimento de usabilidade foi realizado com quatro voluntários que não possuem deficiência visual para avaliar o comportamento da aplicação.

3.4.1.1 Metodologia

O experimento ocorreu durante o mês de junho através de um teste individual com os voluntários. Para realização dos testes foi disponibilizado um *smartphone* Moto G de propriedade do autor. Cada voluntário recebeu um questionário de perfil, uma lista de tarefas e um questionário de usabilidade que estão disponíveis no apêndice C. Os voluntários foram orientados a responder o questionário através da ferramenta Google Forms.

3.4.1.2 Aplicação do teste

Inicialmente, os voluntários foram orientados sobre os objetivos dos testes, dos conceitos envolvendo a aplicação, do seu modo de operação e das tarefas que deveriam ser executadas. Após a realização das tarefas os voluntários foram orientados a responder o questionário informando os resultados obtidos. As tarefas executadas durante os testes buscaram contemplar as principais funcionalidades da aplicação.

A lista de tarefas contém oito questões objetivas. O questionário de usabilidade foi composto por seis questões objetivas e uma questão descritiva. As perguntas procuravam obter informações sobre o uso da aplicação e as impressões dos usuários. Os resultados obtidos são apresentados na subseção 3.4.4.

3.4.1.3 Análise dos resultados obtidos

A primeira análise foi realizada sobre os dados de perfil dos voluntários. O Quadro 22 mostra os perfis dos voluntários envolvidos no experimento.

Quadro 22 – Perfil dos voluntários

Sexo	100% Masculino
Idade	50% 21 – 30 anos 50% 31 – 40 anos
Escolaridade	25% Ensino superior incompleto 75% Ensino superior completo
Você utiliza <i>smartphones</i> /computadores com qual frequência?	100% Frequentemente

A partir do Quadro 22 é possível observar que todos os voluntários já tiveram contato com o ensino superior assim como todos utilizam *smartphones* e computadores com frequência. Tal ponto facilita a adaptação e o uso da aplicação desenvolvida.

Após a análise dos perfis, foi realizada a avaliação dos resultados obtidos a partir da lista de tarefas. O Quadro 23 exibe os resultados obtidos.

Quadro 23 – Lista de tarefas

Tarefa	A tarefa foi executada?
Vá para a tela inicial	100% Sim
Realize a leitura de tela	100% Sim
Encontre e abra a aplicação "Calculadora"	100% Sim
Realize uma operação de soma	100% Sim
Encontre e abra a aplicação "Whatsapp"	100% Sim
Abra uma conversa e envie uma mensagem	100% Sim
Encontre e abra a aplicação "Telefone"	100% Sim
Realize uma ligação telefônica	100% Sim

Conforme o Quadro 23, percebe-se que todas as tarefas propostas foram executadas com sucesso por todos os voluntários. A partir disto pode-se afirmar que a aplicação desenvolvida é funcional.

Por fim, foi realizada a avaliação de usabilidade da aplicação. O Quadro 24 mostra os resultados obtidos a partir das questões objetivas.

Quadro 24 – Avaliação de usabilidade

A aplicação é de fácil utilização?	100% Sim, completamente
O vocabulário utilizado pela aplicação é coerente?	75% Sim, completamente 25% Sim, parcialmente
A aplicação permite que você se localize durante a navegação?	75% Sim, completamente 25% Sim, parcialmente
É possível identificar se uma tarefa foi executada?	25% Sim, completamente 50% Sim, parcialmente 25% Regularmente
A aplicação executou as tarefas da forma esperada?	50% Sim, completamente 50% Sim, parcialmente
Você acredita que a aplicação possa auxiliar pessoas com deficiência visual no uso de <i>smartphones</i> ?	100% Sim

A partir do Quadro 24 é possível perceber a aplicação desenvolvida obteve resultados satisfatórios quanto à usabilidade. Destaca-se o ponto de que todos os voluntários acharam a aplicação desenvolvida de fácil utilização. O ponto que teve a pior avaliação foi a identificação da execução das tarefas, pois há a necessidade de uma leitura de tela após a execução da tarefa para se identificar se a tarefa foi executada ou não, já que o usuário não recebe nenhum retorno automático após a execução da tarefa. Ressalta-se também o fato de que todos os voluntários acreditam que a aplicação desenvolvida pode ajudar pessoas com deficiência visual na utilização de *smartphones*.

Além das questões objetivas, os voluntários também foram questionados sobre a maior dificuldade durante o uso da aplicação. As principais dificuldades relatadas foram a existência de componentes na tela com o mesmo texto, por exemplo, contatos com o mesmo nome, já que apenas o primeiro é selecionado e o tempo de resposta para identificação de voz. Foi sugerido que a aplicação tivesse uma forma de identificar esses componentes com o mesmo texto para que fosse possível interagir com eles de forma clara.

3.4.2 Experimento 02: Avaliação de usabilidade por pessoas com deficiência visual

Este experimento foi realizado por uma pessoa portadora de deficiência visual com o objetivo de comprovar a funcionalidade da aplicação. Neste experimento foi utilizada a mesma metodologia e modelo de aplicação utilizado no experimento anterior.

A execução da lista de tarefas se deu de forma conduzida, sendo realizada uma explicação antes da execução de cada tarefa. Das oito tarefas propostas na lista, sete foram realizadas com sucesso, não sendo possível realizar a ligação telefônica. A dificuldade na realização da ligação se deve ao fato de que o teclado é minimizado dependendo do ponto em que ocorre o toque na tela.

Em relação aos pontos de usabilidade da aplicação, relatou-se que a mesma é de fácil utilização, possui um vocabulário adequado, permite a identificação da execução de uma tarefa e na maior parte das vezes executou as tarefas conforme o esperado. O ponto que obteve a pior avaliação no questionário foi a capacidade de se localizar durante a navegação da aplicação, sendo avaliado como regular, devido ao fato da aplicação fazer sempre a leitura da tela inteira. Foi relatado também como dificuldade no uso, o fato da pontuação não estar integrada com a ação de escrita e o uso do teclado numérico por conta de sua minimização.

Foram sugeridas como melhorias para a aplicação, a criação de comandos para leitura de partes específicas da tela e que no momento em que está ocorrendo a sintetização de voz, o toque na tela apenas a interrompa, sem iniciar o serviço de reconhecimento de voz. Apesar da necessidade de implementação dessas melhorias e das limitações encontradas, relatou-se que a aplicação pode sim auxiliar pessoas com deficiência visual no uso de *smartphones*.

3.4.3 Comparação com os trabalhos correlatos

O Quadro 25 apresenta um comparativo entre o aplicativo desenvolvido e os trabalhos correlatos. As características comparadas são baseadas nos pontos avaliados nos testes.

Quadro 25 – Comparação com os trabalhos correlatos

Características / Trabalhos	Apple (2015)	Samsung (2015)	Microsoft (2015)	Trabalho Proposto
Reconhecimento de Voz	X	X	X	X
Realizar cálculos	X	X	X	X
Enviar mensagens	X	X	X	X
Realizar a leitura de tela	-	-	-	X
Realizar ligações	X	X	X	X
Dependência de rede	X	X	X	X

Conforme mostra o Quadro 25, a vantagem do aplicativo em relação aos trabalhos correlatos esta na realização da leitura de tela. Contudo, na Siri (APPLE, 2015) há a possibilidade de trabalho integrado com o aplicativo VoiceOver, que também realiza a leitura de tela. Porém, no aplicativo VoiceOver, a leitura de tela não é feita através de comandos de voz, mas assim através da metodologia exploração por toque.

Em relação à característica de reconhecimento de voz, apesar de o aplicativo desenvolvido realizar a identificação do que foi dito com precisão satisfatória, a interpretação das frases reconhecidas não está no mesmo nível dos trabalhos correlatos. Enquanto o aplicativo desenvolvido depende de uma estrutura fixa de frases com verbo e sujeito, os trabalhos correlatos não possuem esse tipo de dependência, pois tratam-se de agentes de

conversação mais evoluídos. A única dependência presente nos trabalhos correlatos é a conexão com a rede para a realização do reconhecimento, dependência presente também no trabalho desenvolvido.

4 CONCLUSÕES

Este trabalho apresentou o desenvolvimento de um aplicativo Android para o auxílio no uso de *smartphones* por parte de pessoas portadoras de deficiência visual. Os comandos de voz reconhecidos pelo aplicativo são transformados em uma interação com o dispositivo.

A aplicação foi desenvolvida utilizando a linguagem de programação Java no ambiente Android Studio e as bibliotecas `accessibilityservice`, `accessibility`, `speech` e `speech.tts` do Android. As ferramentas se mostraram funcionais, não apresentando nenhum problema durante o período de desenvolvimento.

Os resultados alcançados foram satisfatórios. Os objetivos de realizar o reconhecimento de voz, fornecer *feedback* auditivo e realizar as ações solicitadas pelo usuário foram alcançados pelo aplicativo. Com as treze ações contidas no aplicativo é possível realizar diversas atividades no *smartphone*, sendo possível inclusive realizar atividades que não estão previstas nos requisitos funcionais, como enviar e-mails ou mensagens através do aplicativo Whatsapp.

Em relação ao reconhecimento de voz, a biblioteca utilizada mostrou-se satisfatória, apresentando bons resultados mesmo no reconhecimento de frases longas, com mais de dez palavras. Em contrapartida, a utilização da aplicação em ambientes barulhentos ou com múltiplas vozes, chega a se tornar inviável, pois não há nenhuma diferenciação entre a voz do usuário e as demais vozes do ambiente. Notou-se também que a dependência de conexão com a internet para realizar o reconhecimento pode gerar uma demora no processo de reconhecimento, caso não haja uma largura de banda adequada.

As limitações encontradas foram a dificuldade ou impossibilidade de interação com alguns componentes da tela. Isso ocorre por motivos como vários componentes na tela com o mesmo texto ou componentes sem texto algum. Também se encontrou dificuldade na operação de alguns aplicativos específicos, como a agenda de contatos, devido à realização de tarefas automaticamente quando ocorre uma troca de tela ou mudanças de estado causadas por essa troca. Essas dificuldades fizeram com que não fosse possível através da aplicação, atender chamadas telefônicas nem criar lembretes na agenda, ambos requisitos funcionais. É importante ressaltar também que há cenários em que as tarefas não serão executadas de acordo com o esperado, devido a características do sistema operacional ou das aplicações de terceiros envolvidas. Por exemplo, o teclado numérico é minimizado em alguns casos ao ocorrer uma troca de telas ou ainda um campo de texto pode ter desabilitado o recurso de colagem.

A partir dos resultados pode-se concluir que a aplicação desenvolvida, apesar das limitações, é uma opção de auxílio aos portadores de deficiência visual no uso de *smartphones*. Por fim, o trabalho desenvolvido pode servir de base ou inspiração para a criação de novas tecnologias assistivas.

4.1 EXTENSÕES

Algumas extensões possíveis para este trabalho são:

- a) adaptação para outras línguas além do português;
- b) criar ciclos de interação com mais de uma ação, para facilitar ou até possibilitar a interação com componentes de sub-menu e teclados;
- c) adicionar mais ações, como checar a carga da bateria ou as horas;
- d) desenvolver uma inteligência artificial para conversação;
- e) incluir um identificador de voz, para que em ambientes barulhentos seja possível diferenciar a voz do usuário em meio a outras vozes;
- f) encontrar uma forma de interação com componentes de tela que não possuem nenhum texto atribuído.

REFERÊNCIAS

- APPLE. **Sobre a Siri**. [S. l.], 2015a. Disponível em: <<https://support.apple.com/pt-br/HT4992>>. Acesso em: 22 ago. 2015.
- _____. **iOS: Uma série de recursos para uma série de necessidades**. [S. l.], 2015b. Disponível em: <<http://www.apple.com/br/accessibility/ios/>>. Acesso em: 22 ago. 2015.
- _____. **Siri**. 2015c. Disponível em: <<http://www.apple.com/br/ios/siri/>>. Acesso em: 22 ago. 2015.
- _____. **Android.accessibilityservice**. [S. l.], 2015a. Disponível em: <<http://developer.android.com/intl/pt-br/reference/android/accessibilityservice/package-summary.html>>. Acesso em: 02 nov. 2015.
- _____. **Android.speech**. [S. l.], 2015b. Disponível em: <<http://developer.android.com/reference/android/speech/package-summary.html>>. Acesso em: 02 nov. 2015.
- _____. **SpeechRecognizer**. [S. l.], 2015c. Disponível em: <<http://developer.android.com/intl/pt-br/reference/android/speech/SpeechRecognizer.html>>. Acesso em: 02 nov. 2015.
- _____. **Android.speech.tts**. [S. l.], 2015d. Disponível em: <<http://developer.android.com/reference/android/speech/tts/package-summary.html>>. Acesso em: 14 nov. 2015.
- _____. **Voice**. [S. l.], 2015e. Disponível em: <<http://developer.android.com/reference/android/speech/tts/Voice.html>>. Acesso em: 14 nov. 2015.
- BERSCH, Rita. **Introdução à tecnologia assistiva**. Porto Alegre: Assistiva, 2013. 20 p. Disponível em: <http://www.assistiva.com.br/Introducao_Tecnologia_Assistiva.pdf>. Acesso em: 19 jun. 2016.
- BRAGA, Daniela Filipa Macedo Moreira da Silva. **Algoritmos de processamento da linguagem natural para sistemas de conversão texto-fala em português**. 2008. 228 f. Tese (Doutorado) - Curso de Filologia, Galego-português, Francês e Linguística, Universidade da Coruña, Coruña, 2008. Disponível em: <<http://ruc.udc.es/dspace/handle/2183/1011>>. Acesso em: 28 fev. 2016.
- GARCÍA, Jesus Carlos Delgado; GALVÃO FILHO, Teófilo Alves. **Pesquisa nacional de tecnologia assistiva**. São Paulo: Instituto de Tecnologia Social (ITS Brasil), 2012. 66 p. Disponível em: <<http://www.ufjf.br/acessibilidade/files/2009/07/miolopesqnacional-grafica-1.pdf>>. Acesso em: 28 fev. 2016.
- HAZARD, Damian; GALVÃO FILHO, Teófilo Alves; REZENDE, André Luiz Andrade. **Inclusão digital de e social de pessoas com deficiência: textos de referência para monitores de telecentros**. Brasília: Unesco, 2007. 72 p. Disponível em: <<http://unesdoc.unesco.org/images/0016/001600/160012POR.pdf>>. Acesso em: 28 fev. 2016.
- INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATÍSTICA. **Estatísticas de Gênero**. 2010. Disponível em: <<http://www.ibge.gov.br/apps/snig/v1/index.html?loc=0>>. Acesso em: 18 set. 2015.
- JURAFSKY, Daniel; MARTIN, James H. **Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition**. 2. ed. New Jersey: Pearson Education, 2009.

MICROSOFT. **Conheça o Cortana**. [S. l.], 2015a. Disponível em: <<http://www.windowsphone.com/pt-br/how-to/wp8/cortana/meet-cortana>>. Acesso em: 23 ago. 2015.

_____. **O que eu posso dizer para o Cortana?** [S. l.], 2015b. Disponível em: <<http://www.windowsphone.com/pt-br/how-to/wp8/cortana/what-can-i-say-to-cortana>>.

Acesso em: 23 ago. 2015.

_____. **Meus interesses e o Cortana**. [S. l.], 2015c. Disponível em: <<http://www.windowsphone.com/pt-br/how-to/wp8/cortana/my-interests-and-cortana>>.

Acesso em: 23 ago. 2015.

_____. **Disponibilidade de recursos e serviços**. [S. l.], 2015d. Disponível em: <<http://www.windowsphone.com/pt-br/how-to/wp8/basics/feature-and-service-availability>>.

Acesso em: 09 set. 2015.

NAVARRO, Juliana Jobim. **A inclusão social dos deficientes visuais e a publicidade brasileira**: um breve panorama. 2012. 62 f. TCC (Graduação) - Curso de Publicidade e Propaganda, Audiovisuais e Publicidade, Faculdade de Comunicação Social da Unb, Brasília, 2012. Disponível em:

<http://bdm.unb.br/bitstream/10483/4259/1/2012_JulianaJobimNavarro.pdf>. Acesso em: 20 jun. 2016.

RADABAUGH, Mary Pat. **Study on the Financing of Assistive Technology Devices of Services for Individuals with Disabilities - A report to the president and the congress of the United State, National Council on Disability**. 1993. Disponível em

<<https://www.ncd.gov/publications/1993/Mar41993#16>>. Acesso em: 30 ago. 2015.

SAMSUNG. **How do I use S Voice on my Samsung Gear Circle?** [S. l.], 2015a. Disponível em: <<http://www.samsung.com/us/support/answer/ANS00038540/238333/SM-R130NZKSXAR>>. Acesso em: 23 ago. 2015.

_____. **Go Hands-Free With S Voice for Your Mobile Device**. [S. l.], 2015b. Disponível em: <<http://www.samsung.com/ae/discover/go-hands-free-with-s-voice-for-your-mobile-device>>. Acesso em: 23 ago. 2015.

_____. **Acessibilidade móvel: tecnologia acessível a todos**. [S. l.], 2015c. Disponível em: <<http://www.samsung.com/br/mobileaccessibility/>>. Acesso em: 23 ago. 2015.

_____. **S Voice**. [S. l.], 2015d. Disponível em:

<<http://www.samsung.com/global/galaxys3/svoice.html>>. Acesso em: 01 set. 2015.

SCHAWARTZ, Steve. **My Samsung Galaxy S5**. Indianápolis: Que Publishing, 2014.

APÊNDICE A – Descrição dos casos de uso

Os casos de uso da aplicação estão descritos a seguir:

- a) UC01 - Ação ler, este caso de uso descreve como o Usuário realiza a leitura dos textos exibidos na tela do *smartphone*;
- b) UC02 - Ação tocar, este caso de uso descreve como o Usuário aplica os eventos de toque nos componentes da tela;
- c) UC03 - Ação desligar, este caso de uso descreve como o Usuário inativa a aplicação;
- d) UC04 - Ação home, este caso de uso descreve como o Usuário faz com que o *smartphone* volte para a tela inicial;
- e) UC05 - Ação botão, este caso de uso descreve como o Usuário aplica eventos de toque nos componentes da tela do tipo botão;
- f) UC06 - Ação escrever, este caso de uso descreve como o Usuário insere textos nos campos de texto da tela;
- g) UC07 - Ação buscar, este caso de uso descreve como o Usuário aplica eventos de toque nos componentes da tela do tipo *image_view*;
- h) UC08 - Ação discar, este caso de uso descreve como o Usuário realiza chamadas telefônicas;
- i) UC09 - Ação rolar, este caso de uso descreve como o Usuário faz a rolagem da tela;
- j) UC10 - Ação letra, este caso de uso descreve como o Usuário monta um texto através de um ditado e o armazena em memória;
- k) UC11 - Ação colar, este caso de uso descreve como o Usuário cola o texto armazenado em memória nos campos de texto da tela;
- l) UC12 - Ação limpar, este caso de uso descreve como o Usuário apaga o texto armazenado em memória;
- m) UC13 - Ação falar, este caso de uso descreve como o Usuário solicita a repetição de uma frase dita.

A partir do Quadro 26 até o Quadro 38 são demonstradas as condições, cenários e exceções de cada caso de uso.

Quadro 26 – UC01 Ação ler

Número	01
Caso de Uso	Ação ler
Ator	Usuário
Pré-condições	1. Tocar na tela para iniciar o reconhecimento de voz
Cenário Principal	1. O usuário fala um dos verbos referentes à ação ler. 2. A aplicação exterioriza todos os textos da tela através do sintetizador de voz.
Exceção	1. O verbo dito pelo usuário não esta presente na lista de verbos mapeados pela aplicação.

Quadro 27 – UC02 Ação toque

Número	02
Caso de Uso	Ação toque
Ator	Usuário
Pré-condições	1. Tocar na tela para iniciar o reconhecimento de voz
Cenário Principal	1. O usuário fala uma frase onde a primeira palavra é um dos verbos referentes à ação toque e a última palavra é o nome do componente que receberá a ação. 2. A aplicação procura um componente na tela com o nome dito. 3. A aplicação aplica a ação de toque sob o componente encontrado.
Exceção	1. O verbo dito pelo usuário não esta presente na lista de verbos mapeados pela aplicação. 2. Não há um componente na tela com o nome dito pelo usuário.

Quadro 28 – UC03 Ação desligar

Número	03
Caso de Uso	Ação desligar
Ator	Usuário
Pré-condições	1. Tocar na tela para iniciar o reconhecimento de voz
Cenário Principal	1. O usuário fala um dos verbos referentes à ação desligar. 2. A aplicação se torna inativa, fazendo o sistema operacional voltar a operar de forma normal.
Exceção	1. O verbo dito pelo usuário não esta presente na lista de verbos mapeados pela aplicação.

Quadro 29 – UC04 home

Número	04
Caso de Uso	Ação home
Ator	Usuário
Pré-condições	1. Tocar na tela para iniciar o reconhecimento de voz
Cenário Principal	1. O usuário fala um dos verbos referentes à ação <i>home</i> . 2. A aplicação faz com que o sistema operacional volte para a tela inicial.
Exceção	1. O verbo dito pelo usuário não esta presente na lista de verbos mapeados pela aplicação.

Quadro 30 – UC05 Ação botão

Número	05
Caso de Uso	Ação botão
Ator	Usuário
Pré-condições	1. Tocar na tela para iniciar o reconhecimento de voz
Cenário Principal	1. O usuário fala uma frase onde a primeira palavra é um dos verbos referentes à ação botão e a última palavra é o nome do botão que receberá a ação. 2. A aplicação procura o botão com o nome dito. 3. A aplicação aplica a ação de toque sob o botão encontrado.
Exceção	1. O verbo dito pelo usuário não esta presente na lista de verbos mapeados pela aplicação. 2. Não há um botão na tela com o nome dito pelo Usuário.

Quadro 31 – UC06 Ação escrever

Número	06
Caso de Uso	Ação escrever
Ator	Usuário
Pré-condições	1. Tocar na tela para iniciar o reconhecimento de voz
Cenário Principal	1. O usuário fala uma frase contendo um verbo referente à ação escrever, o texto a ser escrito e o nome do campo que receberá o texto. 2. A aplicação procura um campo de texto com o nome dito pelo usuário. 3. A aplicação cola o texto dito no campo de texto encontrado.
Exceção	1. O verbo dito pelo usuário não está presente na lista de verbos mapeados pela aplicação. 2. Não há um campo de texto na tela com o nome dito pelo Usuário.

Quadro 32 – UC07 Ação buscar

Número	07
Caso de Uso	Ação buscar
Ator	Usuário
Pré-condições	1. Tocar na tela para iniciar o reconhecimento de voz
Cenário Principal	1. O usuário fala uma frase onde a primeira palavra é um dos verbos referentes à ação buscar e a última palavra é o nome do componente que receberá a ação. 2. A aplicação procura um componente do tipo <code>image_view</code> na tela com o nome dito. 3. A aplicação aplica a ação de toque sob o componente encontrado.
Exceção	1. O verbo dito pelo usuário não está presente na lista de verbos mapeados pela aplicação. 2. Não há um componente do tipo <code>image_view</code> na tela com o nome dito pelo Usuário.

Quadro 33 – UC08 Ação discar

Número	08
Caso de Uso	Ação buscar
Ator	Usuário
Pré-condições	1. Estar com o teclado do telefone aberto 2. Tocar na tela para iniciar o reconhecimento de voz
Cenário Principal	1. O usuário fala um dos verbos referentes à ação discar e os algarismos a serem digitados. 2. A aplicação procura os botões referentes aos números ditos. 3. A aplicação aplica a ação de toque sob os botões encontrados e inicia a ligação.
Exceção	1. O verbo dito pelo usuário não está presente na lista de verbos mapeados pela aplicação. 2. Não há um botão com o nome dito pelo usuário.

Quadro 34 – UC09 Ação rolar

Número	09
Caso de Uso	Ação rolar
Ator	Usuário
Pré-condições	1. Tocar na tela para iniciar o reconhecimento de voz
Cenário Principal	1. O usuário fala um dos verbos referentes à ação rolar e o sentido da rolagem. 2. A aplicação realiza a rolagem de acordo com o sentido dito.
Exceção	1. O verbo dito pelo usuário não está presente na lista de verbos mapeados pela aplicação.

Quadro 35 – UC10 Ação letra

Número	10
Caso de Uso	Ação letra
Ator	Usuário
Pré-condições	1. Tocar na tela para iniciar o reconhecimento de voz
Cenário Principal	1. O usuário fala um dos verbos referentes à ação letra e uma palavra. 2. A aplicação concatena a primeira letra da palavra dita em uma variável de memória. 3. A aplicação repete através do sintetizador de voz a palavra dita.
Exceção	1. O verbo dito pelo usuário não esta presente na lista de verbos mapeados pela aplicação.

Quadro 36 – UC11 Ação colar

Número	11
Caso de Uso	Ação colar
Ator	Usuário
Pré-condições	1. Tocar na tela para iniciar o reconhecimento de voz
Cenário Principal	1. O usuário fala um dos verbos referentes à ação colar e o nome do campo que deve receber o texto. 2. A aplicação procura o campo de texto com o nome dito. 3. A aplicação cola o texto armazenado em memória no campo de texto encontrado.
Exceção	1. O verbo dito pelo usuário não esta presente na lista de verbos mapeados pela aplicação. 2. Não há um campo de texto na tela com o nome dito pelo usuário.

Quadro 37 – UC12 Ação limpar

Número	12
Caso de Uso	Ação limpar
Ator	Usuário
Pré-condições	1. Tocar na tela para iniciar o reconhecimento de voz
Cenário A	1. O usuário fala um dos verbos referentes à ação limpar e a palavra “letra”. 2. A aplicação remove a última letra da variável de memória.
Cenário B	1. O usuário fala um dos verbos referentes à ação limpar e a palavra “tudo”. 2. A aplicação remove todo o conteúdo da variável de memória.
Exceção	1. O verbo dito pelo usuário não esta presente na lista de verbos mapeados pela aplicação.

Quadro 38 – UC13 Ação falar

Número	13
Caso de Uso	Ação falar
Ator	Usuário
Pré-condições	1. Tocar na tela para iniciar o reconhecimento de voz
Cenário Principal	1. O usuário fala um dos verbos referentes à ação falar e o texto a ser repetido pela aplicação. 2. A aplicação exterioriza o texto solicitado através do sintetizador de voz.
Exceção	1. O verbo dito pelo usuário não esta presente na lista de verbos mapeados pela aplicação.

APÊNDICE B – Relação de verbos e ações

A aplicação possui mapeadas vinte e oito palavras como verbos responsáveis por desencadear uma ação. A relação de palavras é exibida no Quadro 39.

Quadro 39 – Verbos e ações

Verbo	Ação
Leia	ACAO_LEITURA
Ler	ACAO_LEITURA
Le	ACAO_LEITURA
Fala	ACAO_FALAR
Fale	ACAO_FALAR
Falar	ACAO_FALAR
Diga	ACAO_FALAR
Desliga	ACAO_DESLIGAR
Desligar	ACAO_DESLIGAR
Parar	ACAO_DESLIGAR
Toque	ACAO_TOQUE
Clique	ACAO_TOQUE
Abra	ACAO_TOQUE
Abre	ACAO_TOQUE
Início	ACAO_HOME
<i>Home</i>	ACAO_HOME
Casa	ACAO_HOME
Botão	ACAO_BOTAO
Escreva	ACAO_ESCREVA
Escrever	ACAO_ESCREVA
Escreve	ACAO_ESCREVA
Buscar	ACAO_BUSCAR
Discar	ACAO_DISCAR
Rolar	ACAO_ROLAR
Ir	ACAO_ROLAR
Letra	ACAO_LETRA
Colar	ACAO_COLAR
Limpar	ACAO_LIMPAR

APÊNDICE C – Lista de tarefas e questionário de Avaliação de usabilidade

Neste apêndice constam o questionário de perfil de usuário, a lista de tarefas executadas pelos usuários e o questionário de usabilidade aplicado ao final do experimento.

Quadro 40 – Questionário de perfil de usuário

PERFIL DO USUÁRIO**Sexo:**

- Feminino
- Masculino

Idade:

- 15 – 20 anos
- 21 – 30 anos
- 31 – 40 anos
- Mais de 40 anos

Escolaridade:

- Ensino fundamental incompleto
- Ensino fundamental completo
- Ensino médio incompleto
- Ensino médio completo
- Ensino superior incompleto
- Ensino superior completo

Você utiliza *smartphones*/computadores com qual frequência?

- Nunca
- Às vezes
- Frequentemente

Quadro 41 – Lista de tarefas

INSTRUÇÕES

Com este questionário buscamos avaliar a utilização da aplicação no auxílio a pessoas com deficiência visual na utilização de *smartphones*.

Você pode utilizar a aplicação por tempo indeterminado para se habituar ao funcionamento.

Com base nas instruções de uso da aplicação, solicitamos que prossiga com os testes conforme as orientações abaixo.

Lista de tarefas:**Vá para a tela inicial.**

A tarefa foi executada?

Sim

Não

Realize a leitura de tela.

A tarefa foi executada?

Sim

Não

Encontre e abra a aplicação "Calculadora".

A tarefa foi executada?

Sim

Não

Realize uma operação de soma.

A tarefa foi executada?

Sim

Não

Encontre e abra a aplicação "Whatsapp".

A tarefa foi executada?

Sim

Não

Abra uma conversa e envie uma mensagem.

A tarefa foi executada?

Sim

Não

Encontre e abra a aplicação "Telefone".

A tarefa foi executada?

Sim

Não

Realize uma ligação telefônica.

A tarefa foi executada?

Sim

Não

Quadro 42 – Avaliação de usabilidade

QUESTIONÁRIO DE USABILIDADE**A aplicação é fácil de usar?**

- Sim, completamente
- Sim, parcialmente
- Regularmente
- Não, parcialmente
- Não, completamente

O vocabulário utilizado pela aplicação é coerente?

- Sim, completamente
- Sim, parcialmente
- Regularmente
- Não, parcialmente
- Não, completamente

A aplicação permite que você se localize durante a navegação?

- Sim, completamente
- Sim, parcialmente
- Regularmente
- Não, parcialmente
- Não, completamente

É possível identificar se uma tarefa foi executada?

- Sim, completamente
- Sim, parcialmente
- Regularmente
- Não, parcialmente
- Não, completamente

A aplicação executou as tarefas da forma esperada?

- Sim, completamente
- Sim, parcialmente
- Regularmente
- Não, parcialmente
- Não, completamente

Qual a sua maior dificuldade durante a utilização?

Você acredita que a aplicação possa auxiliar pessoas com deficiência visual no uso de *smartphones*?

- Sim
- Não