

UNIVERSIDADE DO ESTADO DO RIO GRANDE DO NORTE
CAMPUS DE NATAL
DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO
CURSO DE CIÊNCIA DA COMPUTAÇÃO

AILTON FRANCISCO DE LUNA E SILVA

**BRILLEAPP: Um aplicativo educativo para auxiliar no processo de ensino-
aprendizagem do sistema Braille**

NATAL
2014

AILTON FRANCISCO DE LUNA E SILVA

**BRILLEAPP: Um aplicativo educativo para auxiliar no processo de ensino-
aprendizagem do sistema Braille**

Trabalho de Conclusão de Curso apresentado à
Universidade do Estado do Rio Grande do Norte
como pré-requisito para obtenção do grau de
bacharel em Ciência da Computação.

ORIENTADOR: Prof. Dr. Alberto Signoretti
Prof. Me. Raul Benites Paradedda

NATAL
2014

**Catálogo da Publicação na Fonte.
Universidade do Estado do Rio Grande do Norte.**

Silva, Ailton Francisco de Luna e
BRAILLEAPP: um aplicativo educativo para auxiliar no processo de
ensino-aprendizagem do sistema Braille. / Ailton Francisco de Luna e
Silva.

– Natal, RN, 2014.

74 f.

Orientador(a): Prof. Dr. Alberto Signoretti, Prof. Me. Raul Benites
Paradedá

Monografia (Graduação em Ciência da Computação). Universidade
do Estado do Rio Grande do Norte. Faculdade de Ciências Exatas.

1. Ciência da computação. 2. Sistema Braille – Acessibilidade –
Android. 3. Aplicativo educativo. I. Signoretti, Alberto. II. Universidade do
Estado do Rio Grande do Norte. III. Título.

UERN/BC

CDD 005

AILTON FRANCISCO DE LUNA E SILVA

BRILLEAPP: Um aplicativo educativo para auxiliar no processo de ensino-aprendizagem do sistema Braille

Trabalho de Conclusão de Curso apresentado à Universidade do Estado do Rio Grande do Norte como pré-requisito para obtenção do grau de bacharel em Ciência da Computação.

Aprovado em ___/___/___.

Banca Examinadora

Prof. Dr. Alberto Signoretti (Orientador)
Universidade do Estado do Rio Grande do Norte

Prof. Me. Raul Benites Paradedda (Orientador)
Universidade do Estado do Rio Grande do Norte

Prof. Ma. Bartira Paraguaçu Falcão Dantas Rocha
Universidade do Estado do Rio Grande do Norte

AGRADECIMENTOS

Agradeço primeiramente a minha Mãe (Maria Aparecida), pelo apoio incondicional, incentivo aos estudos deste sempre e por ser uma verdadeira MÃE, ao meu Pai (Francisco Pedro) e irmãos pela compreensão e ajuda que me deram, e a minha namorada, Angélica Carvalho, pelas inúmeras vezes que me incentivou e levantou minha “baixa” autoestima nos momentos de dificuldade.

A todos os amigos e colegas do curso, que ficaram ou que continuam no caminho, pela amizade, companheirismo e aprendizado. Em especial aos amigos João Figueiredo, Bruno Cabral e Damião Renato, por me ensinarem a jogar UNO e por me fazer reprovar na disciplina de Cálculo Numérico, ao amigo Serafim Junior e a amiga Mariêta Cunha, por reclamar sempre que preciso, e às vezes quando não era preciso, por melhorias pra turma.

As instituições nas quais fui estagiário (UERN, UNI-RN e COINE-SESED), por me proporcionar conhecimento e experiência profissional.

A todos os professores e funcionários, em especial aos meus orientadores Alberto Signoretti e Raul Paradedda, pelos ensinamentos, compreensão e dedicação.

Enfim, a todos que de alguma forma contribuíram para a realização deste trabalho.

Nós somos os campeões, meus amigos.
E nós vamos continuar lutando, até o
fim.

Freddie Mercury (1977)

RESUMO

Visando facilitar o processo de ensino-aprendizagem do sistema Braille (sistema universal de leitura e escrita usado por deficientes visuais), este trabalho propõe um aplicativo educativo para auxiliar pessoas, com e sem deficiência visual, que queiram aprendê-lo. Este aplicativo, denominado BrailleApp, foi criado para dispositivos móveis Android, seguindo os padrões de desenvolvimento de acessibilidade para torná-lo o mais acessível possível. Porém, seu uso não pretende substituir os métodos convencionais de ensino-aprendizagem do sistema Braille para pessoas com deficiência visual, mas sim de auxiliá-los. O desenvolvimento deste trabalho foi realizado em fases. Na fase inicial obteve-se o embasamento teórico necessário sobre o sistema Braille, por meio de pesquisas literárias e algumas visitas ao Instituto de Educação e Reabilitação de Cegos do Rio Grande do Norte (IERC/RN). Na fase seguinte, realizou-se a modelagem do sistema onde foram definidos seus requisitos. E por fim, a fase de construção propriamente dita e de testes. Em uma análise simples e empírica, os resultados obtidos foram satisfatórios em relação às funcionalidades e aos requisitos do sistema propostos, descritos na documentação do aplicativo, inclusive os requisitos no tocante a acessibilidade. Com isso, espera-se que o aplicativo BrailleApp possa auxiliar os usuários no processo de ensino-aprendizagem do sistema Braille de forma satisfatória, e que também contribua para a inserção social dos deficientes visuais estimulando sua interação com videntes.

Palavras-chave: Sistema Braille, acessibilidade, Android, aplicativo educativo.

ABSTRACT

To facilitate the teaching-learning Braille system (universal system of reading and writing used by the visually impaired), this paper proposes an educational application to help people with and without visual disabilities who want to learn it. This application, called BrailleApp, was created for Android mobile devices, following the development of accessibility standards to make it as accessible as possible. However, its use is not intended to replace conventional methods of teaching and learning Braille for people with visual disabilities, but rather to assist them. The development of this work was carried out in phases. In the initial phase afforded the necessary theoretical background on the Braille system, by means of literary research and a few visits to the Institute for Education and Rehabilitation of the Blind of Rio Grande do Norte (IERC/RN). In the next phase, we performed system modeling where their requirements were defined. Finally, the phase of actual construction and testing. In a simple, empirical analysis, the results were satisfactory in relation to the features and requirements of the proposed system, described in the application documentation, including requirements regarding accessibility. Thus, it is expected that the application BrailleApp can assist users in the teaching-learning Braille system satisfactorily process, and also to contribute to the social integration of visually impaired stimulating its interaction with seers.

Key words: Braille system, Android, accessibility, educational application.

LISTAS DE FIGURAS

Figura 1: Alfabeto Braille	19
Figura 2: Sinal simples (a) e composto (b)	19
Figura 3: Reglete e punção	20
Figura 4: Célula Braille em modo de escrita	20
Figura 5: Máquina de escrever Braille Perkins	21
Figura 6: Célula Braille em modo de leitura	21
Figura 7: <i>BrailleTouch</i> (Versão 1.2).....	30
Figura 8: <i>Touchscreen Braille Writer</i>	31
Figura 9: <i>Braille Writer</i> (versão 2.1)	31
Figura 10: <i>Braille Guide</i> (Versão 1.0.1)	32
Figura 11: Arquitetura do Android	35
Figura 12: Máquina virtual Dalvik	37
Figura 13: Número relativo de aparelhos com determinada versão	39
Figura 14: Emulador Android	40
Figura 15: Logomarca da FontStruct	41
Figura 16: Diagrama de casos de uso do BrailleApp.....	46
Figura 17: Diagrama ER do aplicativo BrailleApp	47
Figura 18: Tela principal	48
Figura 19: Transcritor Braille / Português.....	48
Figura 20: Enviar SMS.....	48
Figura 21: Transcritor Português / Braille.....	48
Figura 22: Lista de Sinais	49
Figura 23: Exercícios	49
Figura 24: Exercício de leitura Braille	49
Figura 25: Exercício de escrita Braille.....	49
Figura 26: Tela principal (2).....	73
Figura 27: Guia de Referência.....	73
Figura 28: Resultado dos exercícios	73
Figura 29: Configurações	73
Figura 30: Tamanho da fonte	74

Figura 31: Sobre	74
Figura 32: Compartilhar aplicativo	74

LISTAS DE QUADROS

Quadro 1: Requisitos funcionais do aplicativo BrailleApp.....	43
Quadro 2: Requisitos não funcionais do aplicativo BrailleApp	44
Quadro 3: Iniciar aplicativo.....	55
Quadro 4: Transcrever texto em Braille para Português	58
Quadro 5: Enviar SMS	62
Quadro 6: Transcrever texto em Português para Braille	64
Quadro 7: Listar sinais e seus significados	65
Quadro 8: Exibir Guia de Referência.....	66
Quadro 9: Exercitar a Leitura Braille.....	67
Quadro 10: Exercitar a escrita Braille.....	68
Quadro 11: Avaliar aplicativo.	69
Quadro 12: Compartilhar aplicativo	70
Quadro 13: Configura aplicativo (tamanho da fonte, vibração e volume).....	70

LISTA DE ABREVIATURAS E SIGLAS

2D	Bidimensional
3D	Tridimensional
ADT	<i>Android Development Tools</i>
API	<i>Application Programming Interface</i>
APK	<i>Android Package</i>
ASF	<i>Apache Software Foundation</i>
AVD	<i>Android Virtual Device</i>
CPU	<i>Central Processing Unit</i>
DP	<i>Density-independent pixel</i>
DEX	<i>Dalvik Executable</i>
ER	Entidade Relacionamento
IDC	<i>International Data Corporation</i>
IDE	<i>Integrated Development Environment</i>
JVM	<i>Java Virtual Machine</i>
MMS	<i>Multimedia Messaging Service</i>
OHA	<i>Open Handset Alliance</i>
RF	Requisito Funcional
RNF	Requisito Não Funcional
SDK	<i>Software Development Kit</i>
SE	<i>Software Educativo</i>
SMS	<i>Short Message Service</i>
TTS	<i>Text-to-Speech</i>
USB	<i>Universal Serial Bus</i>
WEB	<i>World Wide Web</i>
WI-FI	<i>Wireless Fidelity</i>

SUMÁRIO

1. INTRODUÇÃO	13
1.1 OBJETIVOS E CONTRIBUIÇÕES	14
1.2 METODOLOGIA	15
1.3 ESTRUTURA DO TRABALHO.....	15
2. FUNDAMENTAÇÃO TEÓRICA.....	17
2.1 O SISTEMA BRAILLE	17
2.1.1 Breve história.....	17
2.1.2 O Sistema Braille	18
2.1.3 Escrita em Braille	19
2.1.4 Leitura em Braille	21
2.1.5 Normas de aplicação	22
2.2 ACESSIBILIDADE NA PLATAFORMA ANDROID	22
2.2.1 Serviços e recursos de acessibilidade	23
2.2.2 Como criar aplicações acessíveis.....	24
2.2.2.1 Orientações e casos e considerações especiais	25
2.2.3 Testes de acessibilidade	26
2.2.3.1 Objetivos principais	27
2.2.3.2 Requisitos	27
2.2.3.3 Recomendações	28
2.2.3.4 Como testar recursos de acessibilidade.....	28
3. TRABALHOS RELACIONADOS	30
4. TECNOLOGIAS UTILIZADAS	34
4.1 ANDROID	34
4.1.1 Arquitetura.....	35
4.1.1.1 Sistema Operacional	36

4.1.1.2	Ambiente de execução	36
4.1.1.3	Bibliotecas	37
4.1.1.4	<i>Framework</i>	37
4.1.1.5	Aplicação.....	37
4.2	ANDROID SDK	38
4.2.1	Plataforma (versão do sistema operacional).....	38
4.2.2	Emulador Android	39
4.3	ECLIPSE COM <i>PLUGIN</i> ADT (AMBIENTE DE DESENVOLVIMENTO).....	40
4.4	FONTSTRUCT	41
4.5	CONCLUSÃO	41
5.	BRAILLEAPP	42
5.1	DOCUMENTAÇÃO DO PROJETO	43
5.1.1	Requisitos do sistema.....	43
5.1.2	Diagrama e Documentação de Casos de Uso	45
5.1.3	Diagrama ER do Banco de Dados	47
5.1.4	Interface de usuário.....	47
6.	RESULTADOS	50
7.	CONSIDERAÇÕES FINAIS	51
7.1	TRABALHOS FUTUROS	51
	REFERÊNCIAS	52
	APÊNDICE A – DOCUMENTAÇÃO DOS CASOS DE USO	55
	APÊNDICE B – TELAS DO APLICATIVO BRAILLEAPP	73

1. INTRODUÇÃO

A inserção de ferramentas educativas como um recurso pedagógico tem como objetivo principal facilitar os processos de ensino-aprendizagem, sendo desenvolvidas com a finalidade de levar o aluno/usuário a construção do conhecimento sobre um determinado conteúdo didático proposto. A inclusão da tecnologia na educação, sem dúvida, trouxe uma gama de novas possibilidades e formas facilitadoras nesse processo.

Um aplicativo ou *Software* Educativo (SE) deve atender a alguns requisitos. Segundo Oliveira et al (2001, p.74), as principais características de um *software* educativo são:

- Fundamentação pedagógica, que permita todo o seu entendimento;
- Finalidade didática, por levar o aluno/usuário a construir conhecimento relacionado ao conteúdo observado;
- Interação, entre aluno/usuário e o programa;
- E Facilidade de uso, não devendo exigir do aluno/usuário conhecimentos prévios, mas aplicar uma metodologia que permita que qualquer aluno/usuário, mesmo que em primeiro contato com a ferramenta, seja capaz de utilizá-la e desenvolver suas atividades de forma fácil e ágil.

O atendimento a esses requisitos devem proporcionar ao aluno/usuário a construção do conhecimento a partir da sua interação com o SE.

As ferramentas educativas podem não só levar ao aluno/usuário a construção do conhecimento, mas também desempenhar o papel intermediário de inclusão social, onde a comunicação escrita, verbal, gestual ou outra, é a base para que isso ocorra de forma satisfatória. Um deficiente visual, por exemplo, encontra muita dificuldade em se comunicar de forma escrita, tendo em vista que as pessoas com quem ele convive (familiares, amigos, colegas, professores de escola regular, entre outros) não conhecem o sistema Braille (sistema universal de leitura tátil e de escrita usado por deficientes visuais).

Com a finalidade de facilitar o processo de ensino-aprendizagem do sistema Braille, este projeto propõe um aplicativo educativo voltado para pessoas que enxergam, focado no público iniciante-aprendiz, professores e usuários do sistema Braille. Ele foi desenvolvido com base no conceito de *Desenho Universal*, dentre outras diretrizes de acessibilidade, o que

também permite sua utilização por deficientes visuais, através de serviços e recursos de acessibilidade, contribuindo para inclusão social.

1.1 OBJETIVOS E CONTRIBUIÇÕES

Dentro do contexto de inclusão de tecnologias na educação como ferramenta facilitadora, este trabalho tem como objetivos:

a) **Objetivos Gerais**

Facilitar o processo de ensino-aprendizagem do sistema Braille, através do desenvolvimento de um aplicativo acessível para auxiliar pessoas que queiram aprendê-lo. Neste caso, a acessibilidade se refere, principalmente, a pessoas com deficiência visual ou baixa visão.

b) **Objetivos Específicos**

- Obter uma base de conhecimento sobre o Sistema Braille, que incluem seus métodos de leitura e escrita e normas de aplicação à Língua Portuguesa.
- Realizar um estudo sobre as ferramentas, em especial para dispositivos móveis, que tenham o propósito de auxiliar o ensino-aprendizagem do Sistema Braille.
- Definir as funcionalidades do aplicativo proposto.
- Modelar e construir o aplicativo proposto de forma acessível, principalmente a deficientes visuais e com baixa visão.
- Verificar se todas as funcionalidades atendem aos requisitos predeterminados.

Como mencionado anteriormente, a principal contribuição deste trabalho é proporcionar uma ferramenta para dispositivos móveis acessível que possa auxiliar pessoas que queiram aprender o sistema Braille. A ferramenta proposta foi desenvolvida baseada no conceito de *Desenho Universal*, tal característica permite seu uso por qualquer pessoa, inclusive com deficiência visual, de forma indiferente. Com isso, esperasse incentivar o aumento do uso e prática do sistema Braille, ao mesmo tempo em que propicia a inserção social e ao mundo tecnológico aos deficientes visuais. Outra contribuição relevante é a possibilidade do envio de texto (SMS) por meio de um teclado virtual que adota as nuances do sistema Braille para a entrada de dados.

1.2 METODOLOGIA

Para o desenvolvimento deste trabalho, primeiramente realizou-se a fase de Estudo (concepção), onde se buscou, através de uma revisão literária e de algumas visitas ao Instituto de Educação e Reabilitação de Cegos do Rio Grande do Norte (IERC/RN), o embasamento teórico sobre o sistema Braille, assim como seu processo de ensino-aprendizagem, modo de leitura e escrita e suas ferramentas convencionais de escrita. Além de pesquisas sobre ferramentas correlatas à proposta por este trabalho e a verificação da viabilidade em desenvolver o que está sendo proposto.

Posteriormente, foi concebida a fase de Modelagem (elaboração), onde foram definidas as funcionalidades que seriam implementadas, requisitos, casos de usos e o modelo do banco de dados.

Em seguida realizou-se a fase de Construção, onde foram implementadas todas as funcionalidades definidas de acordo com os requisitos estabelecidos, inclusive os de acessibilidade. A plataforma adotada para o desenvolvimento deste trabalho foi a Android por, dentre outros motivos, ser a plataforma mais utilizada em dispositivos móveis (IDC, 2014), o que possibilita atingir um número maior de usuários.

E na última fase foram realizados testes simples e empíricos das funcionalidades e usabilidade, e testes sobre a acessibilidade do aplicativo proposto, definidos por Android Developer (2014).

1.3 ESTRUTURA DO TRABALHO

Este trabalho está organizado em 7 (sete) capítulos. Neste capítulo foi abordada a introdução, os objetivos gerais e específicos e a metodologia adotada no desenvolvimento deste trabalho.

O capítulo 2 discorre sobre a fundamentação teórica necessária a compreensão técnica deste trabalho. Nele são apresentados conceitos sobre o sistema Braille e a acessibilidade na plataforma Android.

No capítulo 3, são descritas as ferramentas corretas à proposta por este trabalho.

No capítulo 4, é feito uma abordagem sobre as principais tecnologias utilizadas no desenvolvimento deste trabalho, assim como seus conceitos, definições e motivação para uso.

O capítulo 5 discorre sobre o propósito deste trabalho, onde são descritos seus principais objetivos, funcionalidades que o aplicativo oferece e a documentação do mesmo.

No capítulo 6, são apresentados os resultados do desenvolvimento do aplicativo proposto, que incluem os resultados dos testes de acessibilidade.

E por fim, no capítulo 7, são feitas as considerações finais e apresentada as pretensões de trabalhos futuros.

2. FUNDAMENTAÇÃO TEÓRICA

Neste capítulo será apresentado o estudo sobre o referencial teórico que permite o entendimento sobre o sistema Braille e a acessibilidade na plataforma Android. A concepção desses conteúdos são fundamentais para a compreensão técnica deste projeto, pois estão diretamente ligados com o processo de seu desenvolvimento.

Este capítulo é dividido em duas seções principais. A primeira seção (2.1) é dividida em subseções relacionadas ao sistema Braille, que apresentam uma breve história sobre seu surgimento e sobre seu documento normatizador, sua definição, formas de representação e seu alfabeto, formas de escrita e leitura Braille, assim como as ferramentas tradicionais de escrita Braille, e comenta sobre suas normas de aplicação à língua portuguesa.

Na seção 2.2 são apresentadas subseções relacionadas à acessibilidade na plataforma Android, que incluem serviços e recursos de acessibilidade, orientações para criar um aplicativo acessível, e objetivos, requisitos e orientações que devem ser observados na aplicação dos testes de acessibilidade.

2.1 O SISTEMA BRAILLE

2.1.1 Breve história

O sistema Braille foi criado pelo francês Louis Braille em 1825. A partir de 1854, com a criação do Imperial Instituto dos Meninos Cegos, hoje chamado de Instituto Benjamin Constant, o sistema Braille foi adotado no Brasil em sua forma original (CERQUEIRA, 2006).

Em meados de 1940 foi necessário fazer algumas modificações no sistema Braille, de origem francesa, utilizado no país devido à reforma ortográfica da língua portuguesa ocorrida na época. Pela ausência de órgãos ou instituições governamentais que normatizassem o Braille, as alterações ficaram na responsabilidade de professores, técnicos especializados e de instituições ligadas à educação de cegos e à produção de livros em braile, que procuraram manter o sistema acessível e atualizado até a década de 1990 (CERQUEIRA, 2006).

Desde 1996, as Comissões de Braille do Brasil e de Portugal trabalham em conjunto na elaboração de um documento normatizador e de consulta do sistema Braille, hoje com

amparo legal no Protocolo de Colaboração Brasil/Portugal nas áreas de Uso e Mobilidades de Aplicações do Sistema Braille, firmado em Lisboa em 25 de maio de 2000 (CERQUEIRA, 2006).

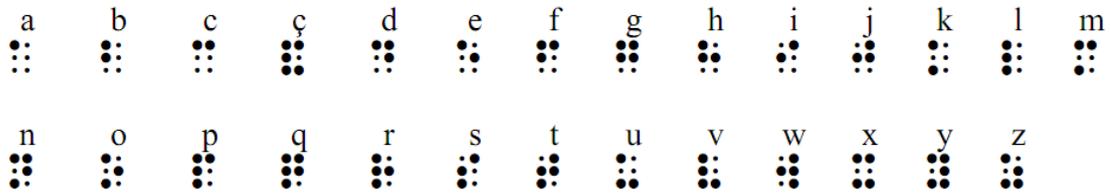
Na portaria nº 2.678 de 24 de setembro de 2002, o MINISTÉRIO DE ESTADO DA EDUCAÇÃO aprova o projeto da Grafia Braille para Língua Portuguesa e recomenda seu uso em todo o território nacional a partir de 01 de janeiro de 2003. Este é um documento normatizador e de consulta destinado a profissionais, como professores, transcritores e revisores, e usuários do sistema Braille. Em 2006 foi lançada sua 2ª edição, intitulado “Grafia Braille para Língua Portuguesa – Braille Integral”. Sua publicação trouxe algumas alterações, novos símbolos e novas normas e regras de aplicação (CERQUEIRA, 2006).

Segundo Cerqueira (2006, p.16), “O principal objetivo dos técnicos que elaboraram este documento foi permitir que o sistema Braille continue sendo o instrumento fundamental na educação, reabilitação e profissionalização das pessoas cegas”.

2.1.2 O Sistema Braille

Braille é o sistema universal de leitura tátil e de escrita em relevo mais eficiente e usado no mundo por deficientes visuais. O sistema Braille é baseado em 63 símbolos, ou sinais, formados pela combinação de seis pontos em relevo dispostos em duas colunas com três pontos cada (Figura 1). Esse agrupamento de pontos é chamado de *senal fundamental* e o espaço por ele ocupado chama-se *célula braille* ou *cela braille*. Com um sinal, ou agrupamentos deles, é possível representar letras, algarismo, símbolos matemáticos e científicos, sinais usados na música e em contexto informático. Pela sua eficiência e uso em escala mundial, o Braille é considerado o melhor meio de leitura e escrita para deficientes visuais (CERQUEIRA, 2006).

Para facilitar o estudo, os pontos que formam os sinais são numerados de um a seis, da esquerda para direita e de cima para baixo. Como dito anteriormente, o sinal braille é constituído de duas colunas com três pontos cada, onde a primeira coluna (esquerda, ⠠) é representada pelos números 1, 2 e 3 e a segunda coluna (direita, ⠡) é representada pelos números 4, 5 e 6. Alguns especialistas consideram que uma célula vazia (representado pelo número zero) também é um sinal braille, sendo assim o sistema passa de 63 para 64 sinais possíveis (CERQUEIRA, 2006).

Figura 1: Alfabeto Braille

Fonte: Elaborada pelo Autor (2014)

Os sinais do Sistema Braille podem ser classificados como simples ou compostos e podem receber significados diferentes dependendo da sua posição no contexto (CERQUEIRA, 2006). Os sinais simples são formados por apenas uma célula braille, já os sinais compostos são formados por dois ou mais sinais simples.

Figura 2: Sinal simples (a) e composto (b)

Fonte: Elaborada pelo Autor (2014)

A Figura 2 ilustra um sinal simples (a) e um sinal composto (b), seus valores e código (representação numérica de um sinal) correspondente. O número 1 (um) (b) é formado por dois sinais simples, o sinal de número (⠠) e o sinal correspondente à letra “a” (a). Nota-se neste caso, Figura (b), que o sinal ⠠ não representa a letra “a” e sim o número um, pois é antecedida do sinal indicador de número (⠠). Essa é uma das várias regras do sistema Braille.

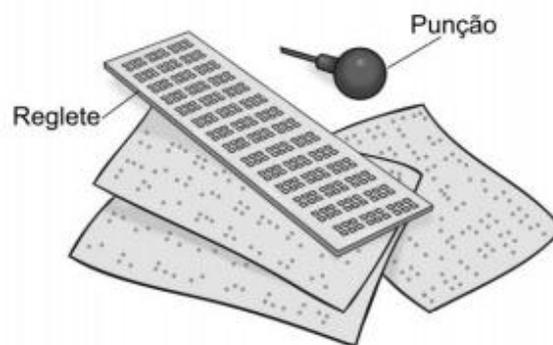
2.1.3 Escrita em Braille

O sistema Braille pode ser escrito basicamente em dois níveis ou graus distintos. No grau um (Braille Integral), as palavras são escritas por extenso, caractere por caractere (sinal por sinal). Já no grau dois (Braille abreviado ou Estenografado), as palavras podem ser escritas de forma abreviada, onde um caractere pode representar duas ou mais letras ou uma palavra inteira (CERQUEIRA, 2006).

Com o desenvolvimento tecnológico foram criados diferentes instrumentos computacionais de apoio à escrita Braille. Entretanto, os instrumentos mais comuns e tradicionais utilizados para a escrita Braille são a reglete e punção (Figura 3) e a máquina de escrever Braille (Figura 5).

Existem algumas variações de regletes com relação a seu tamanho, material de fabricação e acessórios, mas basicamente ela é formada por duas placas unidas por uma, ou mais, dobradiça(s), que permite introduzir a folha entre elas. A placa superior possui aberturas que correspondem às células braille e abaixo de cada abertura, na placa inferior, há uma configuração de célula braille em baixo relevo. Para a marcação dos pontos na reglete utiliza-se a punção, uma ferramenta com ponta adaptada para pressionar o papel na reglete (TECE, 2014).

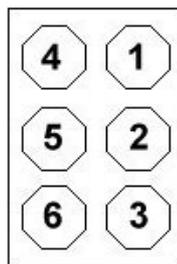
Figura 3: Reglete e punção



Fonte: TECE (2014)

Com a reglete e a punção a escrita Braille é feita ponto a ponto e cada ponto e palavra é escrito da direita para esquerda (Figura 4), pois a folha é pressionada no verso, de forma espelhada, mas a leitura é feita da esquerda para direita (Figura 6).

Figura 4: Célula Braille em modo de escrita



Fonte: SAC (2014)

Outro instrumento de escrita Braille bastante usado é a máquina de escrever Braille. Uma máquina especial de datilografia com modelos de 7, 8 e 9 teclas. No modelo de 9 teclas, há 3 teclas que correspondem à *quebra de linha*, *retocesso* e *espaço*, as demais teclas correspondem a uma célula braille (SAC, 2014).

A escrita na máquina Braille é feita caractere por caractere, da esquerda para a direita (no mesmo sentido da leitura) e as teclas podem ser pressionadas simultaneamente, gerando a combinação de pontos correspondente ao caractere desejado de forma mais rápida e fácil, se comparada a reglete, principalmente por não ser preciso escrever de forma invertida (espelhada). O modelo de máquina Braille mais utilizado é a da Perkins (Figura 5).

Figura 5: Máquina de escrever Braille Perkins



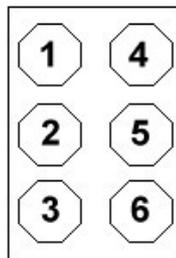
Fonte: CERQUEIRA (2006)

2.1.4 Leitura em Braille

A leitura Braille é feita da esquerda para direita utilizando-se geralmente o dedo indicador, mas alguns leitores também usam o dedo médio ou anular. Em média, os leitores conseguem ler 104 palavras por minutos. Usuários mais experientes conseguem usar as duas mãos chegando a duplicar sua velocidade de leitura (SAC, 2014).

Para uma leitura tátil rápida e eficiente, os pontos em relevo devem ser precisos e terem a mesma dimensão, adequada à ponta dos dedos utilizados para leitura (BRUNO, 2006).

Figura 6: Célula Braille em modo de leitura



Fonte: SAC (2014)

2.1.5 Normas de aplicação

Apesar do Braille ser um sistema universal, ele pode sofrer algumas modificações decorrentes de características particulares de cada língua. Palavras acentuadas e alguns sinais do sistema de pontuação da Língua Portuguesa, são exemplos de características particulares ligadas diretamente a uma língua específica.

Na aplicação do sistema Braille à Língua Portuguesa, a maioria dos sinais conserva seu significado original, originário da língua francesa, porém há alguns caracteres que são representados por sinais exclusivos do sistema Braille. Os sinais do sistema Braille aplicam-se, normalmente, em consonância às normas da ortográfica oficial da língua corrente. Porém, há observações e normas de aplicação que devem ser obedecidas (CERQUEIRA, 2006).

Em Cerqueira (2006) é apresentado todos os sinais do sistema Braille, assim como as observações, regras e normas de aplicação da grafia Braille para língua portuguesa. Segundo Cerqueira (2006), este é o documento normatizador e de consulta recomendado pelo MINISTÉRIO DE ESTADO DA EDUCAÇÃO. Ele foi adotado como base para o desenvolvimento deste trabalho no tocante ao processo de aplicação do sistema Braille à Língua Portuguesa.

2.2 ACESSIBILIDADE NA PLATAFORMA ANDROID

Muitos usuários do Android apresentam habilidades e limitações físicas, visuais, auditivas ou relacionadas a idade que influenciam nas suas percepções táteis, visuais e sonoras. As habilidades e limitações exigem que seus usuários interajam com o sistema de diferentes maneiras.

Uma das missões do Android é dispor recursos que permitam a apresentação de informação útil e universalmente acessível. Segundo Android Developer (2014), a acessibilidade é a medida do sucesso de um produto que pode ser usado por pessoas com capacidades variáveis. Sua missão se aplica a todos os usuários, incluindo pessoas com deficiência visual, de cor (daltonismo), de audição e com limitações motoras.

Um conceito importante na aplicação da acessibilidade é o de *Desenho Universal*, que consiste na prática de desenvolver produtos que são inerentemente acessíveis a todas as

peessoas, independentemente de suas características pessoais, idade ou habilidades. Sua ideia é de um “desenho” único para todos, evitando a necessidade de produtos especiais para pessoas com deficiência, e assegurando que todos possam utilizar com segurança e autonomia os diversos ambientes e produtos (CARLETTO e CAMBIAGHI, 2014).

Os padrões de projeto Android foram desenvolvidos de acordo com os princípios do *Desenho Universal*. Seu acompanhamento permite a construção de aplicativos que atendam aos requisitos básicos de acessibilidade. Com sua adoção e uso de ferramentas de acessibilidade do Android, os aplicativos se tornam mais acessíveis (ANDROID DEVELOPER, 2014).

2.2.1 Serviços e recursos de acessibilidade

O Android oferece várias ferramentas de acessibilidade para auxiliar pessoas com deficiência, principalmente visual, na execução de suas tarefas, facilitando a navegação no dispositivo e a exploração dos aplicativos. Elas incluem serviços e recursos, como *Leitor de tela*, *Explorar por toque*, *Navegar por gestos* e *Navegar por controladores direcionais*. A seguir são descritos os serviços e recursos citados acima, segundo Android Developer (2014):

- **Leitor de Tela:** É um serviço de acessibilidade voltada para deficientes visuais que permite, usando *feedback* falado, descrever eventos de interação do usuário, como navegar no dispositivo e abrir e explorar um aplicativo, e emitir avisos sonoros e táteis (vibração) de ações do sistema, como eventos de notificação. A *Google Inc*¹ fornece um leitor de tela padrão chamado *TalkBack*, que vem, normalmente, pré-instalado na maioria dos dispositivos Android com as versões 4.0 e superiores.
- **Explorar por toque:** É um recurso de acessibilidade do sistema que está disponível nas versões do Android 4.0 e superiores. Com ele é possível tocar e/ou arrastar o dedo na tela do dispositivo e ouvir o conteúdo ou descrição do que está abaixo dele. Este recurso só pode ser ativado se houver algum *Leitor de tela* ativado.
- **Navegar por gestos:** É um recurso de acessibilidade do sistema que permite a navegação no dispositivo e nas aplicações por meio de gestos específicos. Com ele é possível configurar gestos de “atalhos” para determinadas funções, como “Voltar”, “Abrir aplicativos recentes”, “Abrir notificações”, dentre outros. Esse recurso está

¹ *Google Inc* é uma empresa multinacional norte americana de *software* e serviços *online* (INFOESCOLA, 2014).

disponível em versões do Android 4.1 e superiores e só pode ser ativado se o serviço de *Leitor de tela* e o recurso *Explorar por toque* estiverem ativados.

- **Navegar por controladores direcionais:** O Android permite a navegação no dispositivo e nas aplicações por meio de controladores direcionais de *hardware* ou de *software*, como *Trackball*² e *D-pad*³. Eles interagem com o dispositivo e as aplicações de forma semelhante à navegação de 4 vias (direções) do controle remoto de uma TV.

Há também outros recursos de acessibilidade que permitem, dentre outros, aumentar o tamanho da fonte do texto, alterar opções de áudio, alterar velocidade da leitura de textos e definir o mecanismo de síntese de voz.

2.2.2 Como criar aplicações acessíveis

Os aplicativos desenvolvidos para o Android podem ser mais acessíveis aos usuários com limitações visuais, físicas ou relacionadas à idade quando os serviços e recursos de acessibilidade do aparelho são ativados. Esses serviços e recursos podem tornar as aplicações mais acessíveis, mesmo que não se adicione nenhum código de acessibilidade no projeto. No entanto, na maioria dos casos, a criação de uma aplicação acessível para Android não exige muitas mudanças no projeto (código), mas sim, um melhor detalhamento de como os usuários podem interagir com sistema e fornecer *feedbacks* que permitam melhor compreensão (ANDROID DEVELOPER, 2014).

Para aperfeiçoar a acessibilidade de uma aplicação e garantir uma usabilidade agradável para todos os usuários, especialmente na criação de interfaces de usuário usando componentes padrão do Android, são necessários alguns passos. Segundo Android Developer (2014), os passos necessários são:

- Adicionar texto descritivo para controles de interface do usuário, principalmente nos componentes de *imagem*, *botão* e *caixa de seleção*;
- Verificar se todos os elementos da interface de usuário que pode aceitar entradas (toques ou digitação) podem ser alcançados com um controlador direcional, como um *Trackball*, *D-pad* (físico ou virtual) ou com o *Navegar por gestos*;

² *Trackball* é dispositivo de controle de cursos semelhante a um *mouse*, que possui uma esfera, localizada normalmente na parte superior, utilizada para o usuário mover o cursor na tela (Trackball, 2014).

³ *D-pad* é um controlador direcional normalmente de 4 vias usado para navegação linear (D-pad, 2014).

- Verificar se todos os avisos de áudio são sempre acompanhados por outro alerta visual ou notificação, para ajudar os utilizadores que são surdos ou com deficiência auditiva;
- E testar o aplicativo usando apenas serviços e recursos de acessibilidade.

2.2.2.1 Orientações e casos e considerações especiais

Para garantir um nível considerado de acessibilidade em um aplicativo Android devem-se adotar algumas orientações para obtê-la, levando em conta situações específicas em que se devam tomar medidas para que isso seja cumprido. A seguir são descritas as orientações recomendadas por Android Developer (2014):

- Fazer uma navegação intuitiva: O usuário sempre deve saber onde está, independentemente de suas limitações. É preciso fazer um projeto bem estruturado que permita, a construção de um mapa mental do sistema, uma navegação amigável e que tenha um *feedback* tátil, sonoro ou visual. Ela também deve permitir a navegação por controladores direcionais;
- Usar tamanhos recomendados para elementos alvos de toque: O tamanho recomendado para elementos alvos é de 45 dp⁴. Porém, há casos que pode haver a necessidade de usar elementos alvos com dimensões maiores que as recomendações, como em aplicativos educacionais;
- E fornecer alternativas para exibição de elementos temporais: Elementos que são exibidos por um determinado tempo e depois somem, podem não ser percebidos pelos usuários. Um *Leitor de tela*, por exemplo, pode não ler um texto temporário a menos que o mesmo esteja em foco, caso contrário o usuário pode nem saber que aquele conteúdo esteve disponível. Recomenda-se não usar controles de tempo para tarefas importantes ou alterar o comportamento do aplicativo quando houver um serviço de acessibilidade ativado.

Também segundo Android Developer (2014), há casos e considerações especiais a serem observados que podem influenciar no desempenho da acessibilidade de uma aplicação. Na ocorrência de algum caso especial é preciso tomar medidas que as contornem considerando ações apropriadas. A seguir são descritas alguns casos especiais e considerações que devem ser adotadas:

⁴ dp (*Density-independent pixel*) é uma unidade virtual de pixel relativa à resolução da tela do dispositivo (ANDROID DEVELOPER, 2014).

- Dicas em campo de texto: Para campos de entrada de texto, como em *EdiText*, definir o atributo de *android:hint* para indicar ao usuário o conteúdo esperado quando este campo estiver vazio;
- Controles que mudam de função: Controles ou botões que mudam de função de acordo com o contexto da aplicação devem fornecer avisos de áudio para indicar seu estado atual;
- Imagens e gráficos decorativos: Imagens e gráficos meramente decorativos, que não apresentam nenhum conteúdo ou permitem a ação do usuário, não devem ter descrições de acessibilidade declaradas.

Há outros casos especiais, como a implementação de interface de usuário personalizado por meio direto da *API*⁵ do Android, que se deve observar e fazer considerações para que permita uma maior acessibilidade da aplicação.

O cumprimento dos passos e das orientações, incluindo os casos especiais e suas considerações, devem ser seguidos da etapa de testes de acessibilidade, descrita na próxima seção, para a verificação dos requisitos de acessibilidade.

2.2.3 Testes de acessibilidade

A fase de testes é uma etapa importante para tornar uma aplicação acessível a todos os usuários. Nesta fase podem-se encontrar problemas de acessibilidade que não foram percebidas durante o desenvolvimento do projeto.

Nas seções seguintes, serão descritos aspectos importantes da fase de testes de acessibilidade que incluem objetivos gerais, requisitos, recomendações, casos e recomendações especiais, e como testar uma aplicação usando os recursos de acessibilidade do Android.

Atender a todos esses aspectos não garante que a aplicação será totalmente acessível, mas será uma boa prática para alcançá-la. Outro aspecto importante é a realização dos testes, não só por parte do desenvolvedor, mas também, por outras pessoas que conheçam as

⁵ *API* (*Application Programming Interface* ou Interface de Programação de Aplicativos) é um conjunto de funções e padrões estabelecidos por um *software* para prover serviços a outros *softwares* de forma abstrata (FOLDOC, 2014).

diretrizes da fase de testes para que se possa avaliar o nível de acessibilidade da aplicação com mais precisão.

2.2.3.1 Objetivos principais

Segundo Android Developer (2014), os principais objetivos dos testes de acessibilidade devem permitir que os usuários possam, independentemente de suas limitações:

- Configurar e usar o aplicativo sem assistência ou conhecimento prévio;
- E que todas as tarefas possam ser facilmente usadas por meio de controladores direcionais que forneçam *feedback* apropriado e claro ao usuário.

2.2.3.2 Requisitos

Para garantir um mínimo de acessibilidade em uma aplicação, segundo Android Developer (2014), os testes a seguir devem ser realizados:

- Controladores direcionais: Verificar se o aplicativo pode ser operado sem o uso de uma tela sensível ao toque, apenas por controladores direcionais para realizar as principais tarefas do aplicativo;
- Avisos de áudio: Verificar se os elementos que fornecem informações (textos, imagens, gráficos etc) ou que permitam a ação do usuário têm descrições de áudio claras e precisas quando um recurso de acessibilidade estiver ativado, como um *Leitor de tela*;
- *Explorar por toque*: Verificar se os elementos que fornecem informações (textos, imagens, gráficos etc) ou que permitam a ação do usuário têm descrições de áudio claras e precisas quando o recurso *Explorar pelo toque* estiver ativado, não devendo haver elementos sem descrição de áudio;
- Tamanho de elementos tocáveis: Todos os elementos que o usuário possa selecionar ou executar alguma ação deve seguir as recomendações mínimas de 45 dp de comprimento e largura;
- Gestos específicos do aplicativo e *Leitores de tela*: Verificar se os gestos específicos do aplicativo, tais como *zoom* de imagem e lista de rolagem, continuam funcionando quando um *Leitor de tela* estiver ativado. Caso contrário, deve-se adotar uma interface alternativa para essas funções;

- Não fornecer apenas *feedback* de áudio: Todos os alertas sonoros também devem ser acompanhados por um sistema de notificação, *feedback* tátil ou alerta visual.

2.2.3.3 Recomendações

As recomendações de testes a seguir, segundo Android Developer (2014), são necessárias a fim de garantir a acessibilidade e qualidade de uma aplicação:

- Alertas de áudio repetitivos: Verificar se os elementos estreitamente ligados (como em listas) não repetem o mesmo aviso;
- E sobrecarga ou subcarga de alertas de áudio: Verificar se os elementos estritamente ligados fornece um nível adequado de informação de áudio que permita aos usuários compreender e agir sobre um elemento da tela. Um aviso muito curto ou muito longo pode tornar difícil sua compreensão.

2.2.3.4 Como testar recursos de acessibilidade

Para realizar todos os testes de acessibilidade descritos anteriormente é preciso configurar um ambiente (dispositivo) de testes que tenha um *Leitor de tela*, que permita *Explorar por toque* e *Navegar por gestos*, e um *Controlador direcional*, além de atender aos requisitos mínimos de *software* (versão da plataforma).

Para testar uma aplicação usando um *Leitor de tela*, é recomendado o serviço de acessibilidade *TalkBack*. Ele é fornecido pela *Google inc* e normalmente vem pré-instalado nas versões 4.0 e superiores, mas também pode ser baixada gratuitamente na loja oficial de aplicativos para Android (Google Play). Ele será parte dos testes de *Explorar por toque* e *Navegar por gesto*. Para habilitá-lo é preciso:

1. Abrir o aplicativo de ***Configurações*** do dispositivo.
2. Navegar até a categoria ***Acessibilidade*** e selecioná-la.
3. Selecionar ***Acessibilidade*** para habilitá-la.
4. Selecionar ***TalkBack*** para habilitá-lo.

Para testar o recurso *Explorar por toque* é necessário que o dispositivo tenha o *TalkBack* instalado. Para habilitá-lo é preciso:

1. Abrir o aplicativo de ***Configurações*** do dispositivo.

2. Navegar até a categoria ***Acessibilidade*** e selecioná-la.
3. Selecionar ***Acessibilidade*** para habilitá-la.
4. Selecionar ***TalkBack*** para habilitá-lo.

Obs: No Android 4.1 e superiores, o sistema oferece uma mensagem *pop-up* para permitir *Explorar por toque*. Em versões anteriores deve-se voltar à categoria ***Acessibilidade*** e selecionar ***Explorar por toque*** para ativá-lo.

Para testar *Navegar por gestos* é necessário um dispositivo Android 4.1, ou superior, e que tenha o *TalkBack* instalado. Quando os recursos *TalkBack* e *Explorar por toque* são ativados, automaticamente o *Navegar por gestos* é ativada. Os gestos podem apresentar diferentes ações de acordo com o serviço de *Leitor de tela* utilizado, mas suas ações podem ser configuradas seguindo os passos a seguir:

1. Abrir o aplicativo de ***Configurações*** do dispositivo.
2. Navegar até a categoria ***Acessibilidade*** e selecioná-la.
3. Selecionar ***Acessibilidade*** para habilitá-la.
4. Selecionar ***TalkBack*** para abri-lo.
5. Abrir a opção ***Configurações***.
6. Abrir a opção ***Gerenciar gestos***.

Para testar o *Navegar por controladores direcionais* pode-se utilizar o *Emulador Android* (descrito na seção 4.2.2). Ele fornece controladores direcionais, como *D-pad* e *Trackball*, simulado baseado em *hardware* que pode ser utilizado para testar a navegação do aplicativo. Pode-se também utilizar controladores direcionais baseados em *softwares*, como os fornecidos pelo aplicativo *Eyes-Free Keyboard* (<HTTPS://code.google.com/p/eyes-free/>).

Para mais informações de como configurar um ambiente de testes para acessibilidade, consulte a página de suporte para testes de acessibilidade no Android (HTTP://developer.android.com/tools/testing/testing_accessibility.html).

2.3 CONCLUSÃO

Neste capítulo foram abordados os principais referenciais teóricos necessários ao entendimento do processo de desenvolvimento deste trabalho, assim como os requisitos que um aplicativo deve atender para atingir um nível mínimo de acessibilidade e como avaliá-los.

3. TRABALHOS RELACIONADOS

Uma das etapas do desenvolvimento deste trabalho foi a pesquisa e estudo de ferramentas relacionados que tivessem o propósito de auxiliar o processo de ensino-aprendizagem ou que incentivasse, de alguma forma, o interesse da sociedade sobre o sistema Braille. A seguir são descritas as principais ferramentas estudadas relacionadas a este trabalho.

Figura 7: *BrailleTouch* (Versão 1.2)



Fonte: BRAILLETECH (2014)

A Figura 7 exibe a tela principal do aplicativo *BrailleTouch* desenvolvido pela BrailleTech ([HTTP://brailletouchapp.com](http://brailletouchapp.com)) para dispositivos *iPhone* e *iPod*. Há duas versões deste aplicativo disponíveis, uma gratuita e outra paga (completa).

Na versão gratuita, ele apresenta um teclado com o estilo baseado na célula braille que permite que o usuário forme caracteres e palavras ao pressionar os pontos correspondentes, além de ouvir os caracteres e palavras inseridas. Os pontos da célula são enumerados e dispostos em duas colunas com três pontos cada. O arranjo dos pontos adotado pelo *BrailleTouch* é a mesma utilizada na escrita braille por meio da ferramenta reglete, onde a enumeração dos pontos é feita de forma espelhada com relação a disposição convencional dos pontos na leitura braille. Na versão completa, além das funcionalidades encontradas na versão gratuita, ele permite o envio de mensagens de textos, *tweets* e *e-mails* através do teclado *touchscreen* braille (BRAILLETECH, 2014).

Figura 8: *Touchscreen Braille Writer*



Fonte: BONNINGTON (2014)

O *Touchscreen Braille Writer* (Figura 8) é um aplicativo para dispositivos móveis (*tablet* e celular) Android que apresenta uma configuração de teclas semelhante à de uma máquina Braille Perkins. Seu *layout* dispõe de 8 pontos, dos quais 6 são utilizados para compor a célula braille e 2 para excluir um caractere e retroceder uma linha. Os 8 pontos são definidos tocando na tela com os dedos simultaneamente para definir sua posição, mas o usuário pode redefini-los a qualquer momento. O tamanho dos pontos e a distâncias entre eles também podem ser configurados. Ele permite ao usuário digitar de forma rápida, segura e confortável, retornando *feedbacks* sonoros quando os caracteres e palavras são formados (BONNINGTON, 2014).

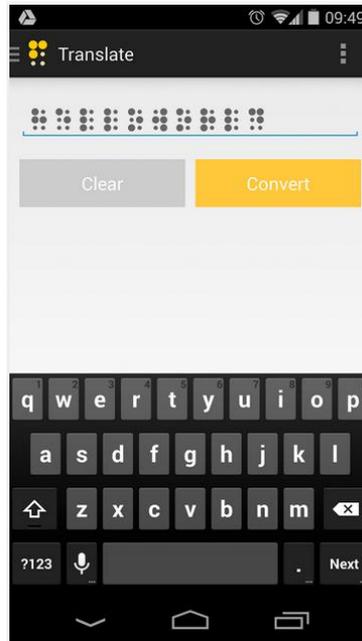
Figura 9: *Braille Writer* (versão 2.1)



Fonte: KIREV (2014)

O *Braille Writer* (Figura 9) é um aplicativo para Android 2.3 e superiores, desenvolvido por Nikola Kirev. Ele permite inserir textos com teclado convencional (padrão) ou com um “teclado braille”, e os exibe em “tinta” e em braille (KIREV, 2014).

Figura 10: *Braille Guide* (Versão 1.0.1)



Fonte: FISKUR (2014)

O *Braille Guide* (Figura 10), desenvolvido pela empresa Fiskur ([HTTP://fiskur.eu/](http://fiskur.eu/)), é um aplicativo gratuito desenvolvido para dispositivos Android 4.0 e superior e dispõe de alguns utilitários interessantes relacionados ao sistema Braille.

O *Braille Guide* oferece os seguintes utilitários: *Interativo*, permite que o usuário escreva um caractere através de uma célula braille interativa disposta visualmente na tela; *Exercício*, incentiva o usuário a exercitar a leitura braille, de forma visual, com a exibição de 10 palavras escritas em braille, onde o usuário deve identificar seus caracteres correspondentes e ao final é exibido o resultado do exercício; *Caracteres*, exibe o alfabeto, números e alguns sinais de pontuação com o seu símbolo correspondente em braille; e *Traduzir*, permite o usuário transcrever caracteres e palavras para o sistema Braille (FISKUR, 2014).

Essas ferramentas mostraram-se boas alternativas de estudo no sistema Braille para professores e alunos, principalmente na fase de alfabetização. Porém, elas foram desenvolvidas com base na grafia Braille para a língua inglesa, e como descrito na subseção 2.1.5, apesar do Braille ser um sistema universal, pode sofrer algumas modificações em cada

idioma. Logo, palavras acentuadas e alguns sinais do sistema de pontuação da língua portuguesa não são suportados por essas ferramentas. E principalmente, elas adotam regras de transcrição diferentes das utilizadas na grafia Braille para língua portuguesa, o que pode acarretar diferentes resultados no processo de transcrição para a mesma palavra dependendo do idioma utilizado.

O estudo realizado sobre as ferramentas descritas proporcionou o conhecimento sobre as mesmas e ajudou na identificação e definição das funcionalidades (serviços) que o BrailleApp dispõe, tomando-se como referências as boas funcionalidades encontradas e observando-se as necessidades na melhoria ou criação de novos serviços.

4. TECNOLOGIAS UTILIZADAS

Nas próximas seções deste capítulo, serão apresentadas as principais tecnologias utilizadas no desenvolvimento do aplicativo proposto por este trabalho, assim como os seus conceitos, definições e motivação de escolha.

4.1 ANDROID

Android é uma plataforma completa voltada para dispositivos móveis, formada por um conjunto de *softwares* sobre uma arquitetura flexível disposta em camadas que incluem, entre outros componentes, um sistema operacional, *middleware*⁶ e aplicativos (PEREIRA, 2009).

Com o Android, é possível integrar aplicações, sejam elas nativas ou não, de forma simples. Sua arquitetura é construída de modo a permitir aos desenvolvedores utilizar todos os recursos que um aparelho móvel possa dispor. Com isso, é possível criar aplicações que interajam com outros aplicativos ou utilizem recursos disponíveis nos aparelhos, como acessar a agenda telefônica, efetuar chamadas, enviar mensagens de texto, usar a câmera, usar o GPS, dentre outros (PEREIRA, 2009).

O Android é desenvolvido pela *Open Handset Alliance* (OHA), uma aliança feita por um grupo de empresas líderes do mercado de telefonia, liderada pela *Google inc*, com o objetivo de padronizar uma plataforma para dispositivos móveis de código aberto e livre, e que fosse moderna e flexível para o desenvolvimento de aplicações. Tudo isso para atender as expectativas e tendências do mercado e deixar os consumidores mais satisfeitos. A licença adotada pelo Android é a *Apache Software Foundation* (ASF), que permite que diversos desenvolvedores de todo o mundo possam alterar seu código fonte para contribuir na sua melhoria ou customizações pessoais, que não precisam ser compartilhadas com a comunidade de desenvolvimento (LECHETA, 2010).

O Android foi a plataforma escolhida para o desenvolvimento do aplicativo proposto neste trabalho. Dentre outros motivos, os principais foram: por ser uma plataforma livre, o que permite a sua utilização, a de suas ferramentas e *API's* sem custo financeiro; por dispor de uma *API* rica em recursos, inclusive de acessibilidade, que permite utilizar de forma abstrata

⁶ *Middleware* é uma camada de *software* mediadora entre *softwares* ou *software* e *hardware* que permite o uso de recursos e serviços de forma facilitada, abstrata (TELECO, 2014).

todas as funcionalidades que um dispositivo pode oferecer; por adotar a linguagem de programação Java, uma excelente, poderosa e uma das mais utilizadas linguagem de programação no mundo (TIOBE, 2014), para o desenvolvimento de suas aplicações; e por ser a plataforma mais usada em dispositivos móveis (*tablets, smartphones* etc.), segundo a *International Data Corporation* (IDC, 2014), o que possibilita atender a um número maior de usuários.

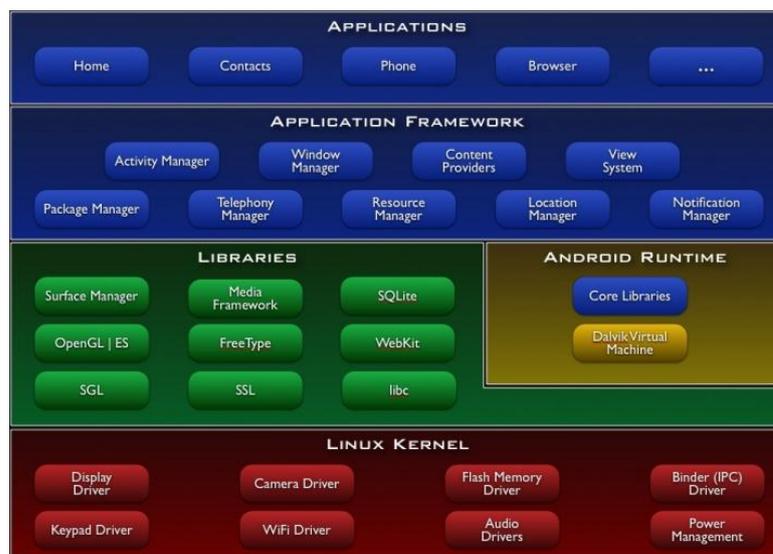
Segundo a IDC (2014), provedora global de serviços de consultoria e eventos para os mercados de tecnologia da informação, telecomunicação e tecnologia de consumo, em resultado divulgado em 12 de fevereiro de 2014, a plataforma mais usada em dispositivos móveis é o Android com 78,1%, seguida do iOS com 17,6%, Windows Phone com 3%, BlackBerry com 0,6% e outras 0,7% (IDC, 2014).

4.1.1 Arquitetura

A arquitetura do Android é estruturada em camadas, ou pilha, onde cada camada prover recursos e serviços a sua camada superior. Essa arquitetura facilita o desenvolvimento de aplicações, tendo em vista que é possível acessar serviços e recursos disponíveis nas camadas inferiores de forma abstrata através de bibliotecas em alto nível (PEREIRA, 2009).

A Figura 11 mostra a arquitetura do Android e como suas camadas são distribuídas. Nas seções seguintes será descrita cada camada que a compõe.

Figura 11: Arquitetura do Android



Fonte: ANDROID DEVELOPER (2014)

4.1.1.1 Sistema Operacional

O sistema operacional do Android é baseado no *Kernel 2.6* do Linux⁷. O *Kernel* é o responsável por gerenciar os principais recursos e serviços do sistema, tais como memória, processos, *threads*, pilhas de protocolos de rede e modelo de *drivers* e segurança. O *Kernel* também atua como uma camada de abstração entre o *hardware* e a pilha de *software* superior.

Diversas aplicações e processos podem ser executados simultaneamente no Android, onde cada aplicativo dispara um novo processo no sistema operacional, e o *Kernel* é o responsável por realizar o controle de acesso aos recursos disponíveis e os compartilhados entre as aplicações. Quando necessário, o sistema operacional pode encerrar algum processo para liberar memória e/ou outro recurso, e posteriormente reiniciá-lo, se for preciso. É no sistema operacional onde está também o sistema de gerenciamento de energia do Android, recurso muito importante para dispositivos móveis (LECHETA, 2010).

4.1.1.2 Ambiente de execução

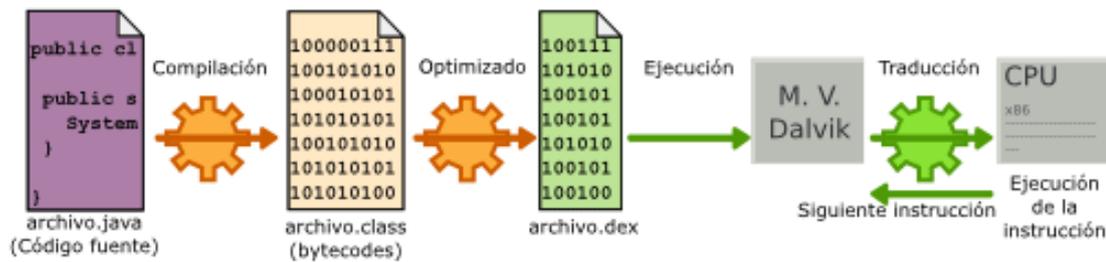
Para cada aplicação executada no Android é criada uma instância da máquina virtual Dalvik no ambiente de execução (*Android Runtime*), porém ela não é uma Máquina Virtual Java⁸ (JVM) convencional. Dalvik é uma máquina virtual otimizada para dispositivos móveis, mas com melhor desempenho, maior integração com a nova geração de *hardware* e projetado para executar várias máquinas virtuais simultaneamente. Foi desenvolvida para funcionar em dispositivos com poucos recursos de *hardware* e projetada para consumir o mínimo de memória, bateria e CPU possível (LECHETA, 2010).

A Figura 12 exemplifica o processo de execução da Dalvik, que vai desde a codificação da aplicação à execução do arquivo *.DEX* (*Dalvik Executable*). Inicialmente o código fonte é compilado, gerando o *bytecode* (*.class*). Em seguida, o *bytecode* é otimizado e convertido para o formato *.DEX*. E por fim, o arquivo *.DEX* é passado à Dalvik. Esse processo é realizado através da ferramenta *DX* distribuída juntamente ao SDK do Android (PEREIRA, 2009).

⁷ Linux é um sistema operacional, um conjunto de programas responsáveis pela comunicação entre *hardware* e *software* e funcionamento de um dispositivo (computador, celular, *tablet*, ect) (VIVA AO LINUX, 2014).

⁸ Máquina Virtual Java, ou JVM (*Java Virtual Machine*) é um *software* que carrega e executa programas escritos em Java, convertendo os *bytecodes* (código do programa compilado) em código executável de máquina. A JVM é a principal responsável pela portabilidade dos códigos Java, de forma que, os programas escritos em Java podem ser executados em qualquer plataforma, desde que tenha uma JVM (ALECRIM, 2005).

Figura 12: Máquina virtual Dalvik



Fonte: RODRIGUEZ (2014)

Nessa camada também está o *Core Libraries* (núcleo de bibliotecas) que inclui um conjunto de bibliotecas do núcleo da linguagem Java (PEREIRA, 2009).

4.1.1.3 Bibliotecas

A camada de bibliotecas inclui um conjunto de bibliotecas nativas do Android escritas na linguagem C/C++ utilizadas pelo sistema. Estão incluídas nesse conjunto, bibliotecas que fornecem recursos para manipular objetos multimídia, visualização 2D e 3D, funções gráficas, funções para navegadores *web*, funções de aceleração de *hardware*, renderização 3D, fontes bitmap e vetorizadas e funções de acesso ao banco de dados SQLite (PEREIRA, 2009).

4.1.1.4 Framework

A camada *Framework* (*Application Framework*) é formada por todas as *API's* que fornecem os serviços e recursos usados pelos aplicativos para realizar suas funções. Essas *API's* dispõem de classes visuais, que incluem elementos como lista, caixa de textos, botões e outros elementos visuais, provedor de conteúdo (*Content Provider*), que possibilita que uma aplicação possa acessar informações de outras aplicações, serviços de comunicação via *Bluetooth*, *Wi-Fi* e conexão *USB*, serviços de localização e de sensores, dentre outros recursos (PEREIRA, 2009).

4.1.1.5 Aplicação

A camada de Aplicação está no topo da arquitetura do Android. Esta é a camada de interação entre o dispositivo Android e o usuário. Nela encontram-se todos os aplicativos nativos e de desenvolvidos por terceiros, como Cliente de Mensagens (SMS e MMS), Cliente de *E-mail*, Calculadora, Agenda Telefônica, Navegadores, dentre outros. A linguagem adota para desenvolver aplicações para o Android é a Java (PEREIRA, 2009).

4.2 ANDROID SDK

Android SDK (*Software Development Kit*) ou Kit de Desenvolvimento de Software Android é um conjunto de ferramentas e tecnologias utilizadas para o desenvolvimento, testes e depuração de aplicações no Android. Inclui um emulador para simular um aparelho móvel (celular, tablete etc), ferramentas utilitárias, uma *API* para a linguagem Java completa com todas as classes necessárias para desenvolver aplicações, e uma documentação para que se construam aplicações de acordo com um padrão de desenvolvimento estabelecido (LECHETA, 2010).

O Android SDK inclui dois utilitários essenciais para o desenvolvimento de aplicações, o *SDK Manager* e o *AVD Manager*. O *SDK Manager* (Gerenciador do SDK) é o utilitário usado para baixar as ferramentas, plataformas e outros componentes. E o *AVD Manager* (Gerenciador de AVD) é o utilitário usado para criar e gerenciar Dispositivos Android Virtuais (AVDs) por meio de uma interface gráfica.

Com os AVDs é possível configurar ambientes de desenvolvimento semelhante ao de um aparelho real, especificando a plataforma (versão do sistema operacional), a arquitetura do processador, a capacidade de armazenamento interno, a resolução da tela, se há câmeras frontais e/ou traseiras e outras configurações que serão simuladas pelo emulador (ANDROID DEVELOPER, 2014).

4.2.1 Plataforma (versão do sistema operacional)

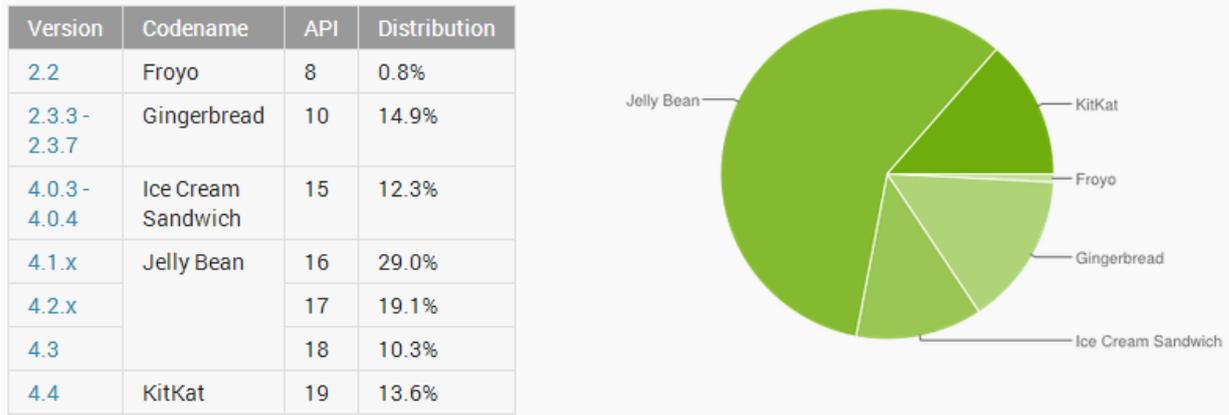
No Android, plataforma é o nome dado a uma versão do sistema operacional. Cada plataforma tem um código identificador, chamado de *API Level*, que indica qual a versão da *API* utilizada (LECHETA, 2010).

Com a evolução tecnológica e as exigências dos usuários no mercado de dispositivos móveis, as atualizações dos aparelhos são inevitáveis. O que implica na criação de novas versões com novas funcionalidades e conseqüentemente novas *API's*.

A escolha de uma plataforma alvo para desenvolvimento é de suma importância e deve ser baseada em alguns critérios. Os critérios adotados para a escolha da versão alvo do sistema operacional deste trabalho foram: os recursos disponíveis nas *API's*, que permitissem a criação da aplicação de forma mais acessível, e o número relativos de aparelhos que a utilizam, com a pretensão de atingir o maior número de usuários possível.

No desenvolvimento do aplicativo proposto, foram utilizados recursos criados na *API Level 14* (versão 4.0). Logo, as versões suportadas pelo aplicativo proposto são a 4.0 (mínima) e superiores, contemplando mais de 84% dos aparelhos Android.

Figura 13: Número relativo de aparelhos com determinada versão

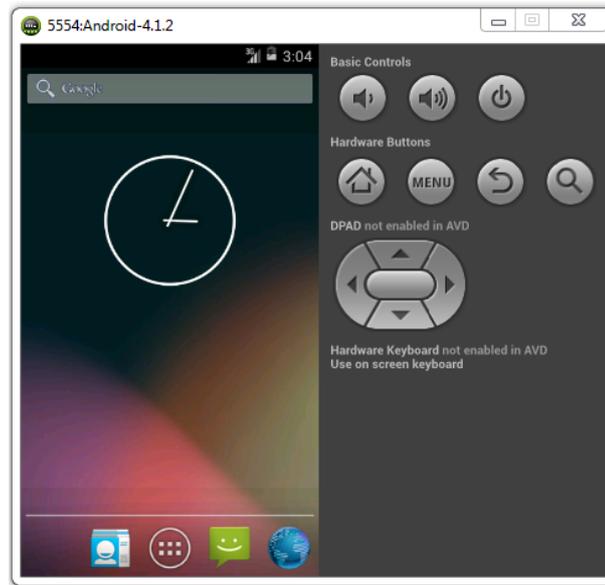


Fonte: ANDROID DEVELOPER (2014)

A Figura 13 ilustra através de uma tabela e um gráfico o número relativo de dispositivos que utilizam determinada versão do Android. Esses dados foram coletados pela *Google Play App Store* durante 7 dias encerrado em 4 de junho de 2014. As versões com distribuição inferior a 0,1% não foram exibidas (ANDROID DEVELOPER, 2014).

4.2.2 Emulador Android

Uma das ferramentas que acompanham o Android SDK é o emulador de dispositivos móveis, Emulador Android. Com ele é possível desenvolver e testar aplicações sem o uso de dispositivos reais físicos. Para iniciar o emulador é preciso especificar um AVD, que indica, dentre outros atributos, a plataforma, a capacidade de armazenamento interno e a resolução da tela do dispositivo (ANDROID DEVELOPER, 2014).

Figura 14: Emulador Android

Fonte: Elaborada pelo Autor (2014)

4.3 ECLIPSE COM *PLUGIN* ADT (AMBIENTE DE DESENVOLVIMENTO)

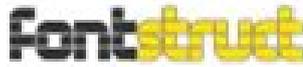
Para aumentar a produtividade no desenvolvimento de aplicações para Android, é recomendado o uso da IDE (*Integrated Development Environment*) Eclipse integrada com o *plugin* ADT (*Android Development Tools*). O *plugin* ADT amplia os recursos do Eclipse para que se possa configurar de forma fácil e rápida, novos projetos, construir interfaces gráficas, adicionar novas *API's*, depurar aplicativos usando as ferramentas do Android SDK, e até exportar pacotes de aplicativos (*APKs*) assinados, ou não, para distribuição. Seu uso é o recomendado, porém pode-se utilizar outras IDEs ou até mesmo usar diretamente as ferramentas do Android SDK para construir e depurar as aplicações (ANDROID DEVELOPER, 2014).

A IDE Eclipse integrada com o *plugin* ADT foram umas das ferramentas de desenvolvimento adotadas neste projeto por proporcionar um ambiente completo e por ser uma referência para desenvolvimento de aplicativos Android.

4.4 FONTSTRUCT

FontStruct é uma ferramenta *web* gratuita de criação de fontes, concebida e desenvolvida pela *FontShop International* ([HTTP://www.fontshop.com/](http://www.fontshop.com/)). O principal utilitário da FontStruct é o FontStructor, uma aplicação que permite que qualquer pessoa construa fontes de forma fácil e rápido a partir de formas geométricas básicas que são dispostas em um padrão de grade. As fontes criadas por essa ferramenta, podem ser baixadas como fontes *Truetype*⁹ e utilizadas por qualquer aplicação que suporte esse tipo fonte (FONTSTRUCT, 2014).

Figura 15: Logomarca da FontStruct



Fonte: FONTSTRUCT (2014)

Ela foi utilizada para a criação da fonte Braille do aplicativo proposto, porém, além da criação da fonte, foi preciso fazer análises e adaptações no processo de transcrição do Braille para a língua portuguesa e vice-versa. Pois, somente com a fonte não é possível mapear todos os caracteres da língua portuguesa nem seguir a todas as regras e normas de aplicação.

A FontStruct foi escolhida por ser uma ótima ferramenta para criação de fontes de forma fácil, rápida e gratuita. Além de permitir sua distribuição num formato suportado pela plataforma Android.

4.5 CONCLUSÃO

Neste capítulo foram descritas as principais tecnologias usadas no desenvolvimento deste projeto, assim como a motivação para seu uso. Com o estudo realizado sobre elas, foi possível a criação do aplicativo proposto.

⁹ *Truetype* é um tipo de fonte desenvolvido pela *Apple Computer* nos anos 80, que é atualmente umas das mais usadas nas plataformas *Microsoft Windows* e *Mac Os* (BEAR'S, 2014).

5. BRAILLEAPP

Este trabalho propõe um aplicativo desenvolvido para dispositivos móveis Android 4.0 e superiores com o objetivo de auxiliar as pessoas, com e sem deficiência visual, que queiram aprender o sistema Braille, chamado de BrailleApp. Seu desenvolvimento foi baseado no conceito de *Desenho Universal*, o que permite seu uso por qualquer pessoa, que podem ter habilidades e limitações diferentes, de forma indistinta. Mas, para que isso pudesse se consolidar foi necessário seguir os requisitos de acessibilidade recomendados por Android Developer (2014).

Uma observação importante é que o BrailleApp não pretende substituir os métodos convencionais de ensino-aprendizagem do sistema Braille para pessoas com deficiência visual, mas sim de auxiliá-los.

No tocante à aplicação do sistema Braille, o BrailleApp foi baseado, especificamente, na grafia Braille para língua portuguesa e dispõe das seguintes principais funcionalidades:

- **Transcrição da grafia Braille para língua portuguesa:** Transcreve texto em Braille, inseridas através de uma célula braile virtual interativa desenhada na tela do aparelho, para língua portuguesa. Nesta opção também é possível enviar SMS com o texto transcrito (em português).
- **Transcrição da grafia da língua portuguesa para Braille:** Transcreve texto em português, inseridas através de um teclado convencional, para Braille.
- **Listagem dos sinais braile:** Lista todos os sinais Braille e seus respectivos significados.
- **Guia de referência sobre a grafia Braille para língua portuguesa:** Documento de consulta que apresenta o código Braille na grafia da língua portuguesa, suas regras e normas de aplicação.
- **Exercício de leitura Braille:** Permite exercitar a leitura em Braille, onde são exibidas 10 palavras, individualmente, escritas em Braille e o usuário tenta transcrevê-las para português.
- **Exercício de escrita Braille:** Permite exercitar a escrita em Braille, onde são exibidas 10 palavras, individualmente, escritas em português e o usuário tenta transcrevê-las para Braille, por meio de uma célula braile virtual interativa desenhada na tela do aparelho.

Na seção seguinte será apresentada a documentação do aplicativo proposto.

5.1 DOCUMENTAÇÃO DO PROJETO

Segundo Guedes (2009), um sistema de informação, independente da sua complexidade, deve ser documentado detalhadamente para que possa ser mantido com maior controle e segurança, e que uma maneira muito eficaz de documentar um sistema é modelá-lo.

Como parte da documentação do aplicativo proposto, serão descritos nas seções seguintes, seus requisitos funcionais e não funcionais, diagrama e documentação dos casos de uso, diagrama ER do banco de dados e as interfaces de usuário.

5.1.1 Requisitos do sistema

Uma das etapas mais importantes no desenvolvimento de *software* são o levantamento e análise de requisitos. Nessas etapas são identificadas as necessidades que o sistema deve suprir e se elas são possíveis de se desenvolver (GUEDES, 2009). Os requisitos podem ser classificados em funcionais e não funcionais. Os requisitos funcionais (RF) descrevem as funcionalidades que o sistema deve oferecer e como ele deve se comportar com relação a entradas específicas e em determinadas situações. Já os requisitos não funcionais (RNF) são características específicas do sistema, como segurança, usabilidade, validação, restrição e interface de usuário (SOMMERVILLE, 2007). A seguir são descritos os requisitos funcionais e não funcionais do aplicativo BrailleApp.

a) Requisitos Funcionais

Quadro 1: Requisitos funcionais do aplicativo BrailleApp

IDENTIFICAÇÃO	REQUISITOS FUNCIONAIS	PRIORIDADE
RF 01	Iniciar aplicativo	ESSENCIAL
RF 02	Transcrever texto em Braille para Português	ESSENCIAL
RF 03	Enviar SMS	ESSENCIAL
RF 04	Transcrever texto em Português para Braille	ESSENCIAL

RF 05	Listar sinais	ESSENCIAL
RF 06	Exibir o guia de referência sobre o Braille	ESSENCIAL
RF 07	Exercitar a leitura Braille	ESSENCIAL
RF 08	Exercitar a escrita braile	ESSENCIAL
RF 09	Avaliar aplicativo	ESSENCIAL
RF 10	Compartilhar aplicativo	ESSENCIAL
RF 11	Configurar aplicativo (tamanho da fonte, vibração e volume)	ESSENCIAL

Fonte: Elaborado pelo Autor (2014)

b) Requisitos não funcionais

Quadro 2: Requisitos não funcionais do aplicativo BrailleApp

IDENTIFICAÇÃO	REQUISITOS NÃO FUNCIONAIS	PRIORIDADE
RNF 01	Criar o Banco de Dado ao iniciar o aplicativo pela primeira vez.	ESSENCIAL
RNF 02	Ao iniciar o aplicativo, verificar se há o pacote de voz <i>TTS</i> português instalado no dispositivo. Caso não haja, informar a necessidade em instalá-lo e indicar o local de seu <i>download</i> .	ESSENCIAL
RNF 03	Para instalar o pacote de voz <i>TTS</i> português é necessário acesso a <i>internet</i> .	ESSENCIAL
RNF 04	Para que o aplicativo possa “falar” (converter texto em voz) em português é necessário que o pacote de voz <i>TTS</i> português esteja instalado no dispositivo.	ESSENCIAL
RNF 05	O envio de SMS só é possível em dispositivos	ESSENCIAL

	que disponham deste serviço.	
RNF 06	O envio de SMS pode ser tarifado.	ESSENCIAL
RNF 07	Para avaliar o aplicativo é necessário que o dispositivo tenha um serviço que permita acesso à loja da <i>Google Play</i> e tenha acesso a <i>internet</i> .	ESSENCIAL
RNF 08	Para compartilhar o aplicativo é preciso que haja no dispositivo um aplicativo para compartilhamento de texto. O envio do texto é passível de tarifação, dependendo do aplicativo utilizado.	ESSENCIAL
RNF 09	O aplicativo deverá dispor de uma interface de usuário acessível, segundo as recomendações de desenvolvimento de acessibilidade estabelecidas por Android Developer (2014), que permita o uso de ferramentas de acessibilidade do Android.	ESSENCIAL
RNF 10	Os serviços e recursos de acessibilidade do Android devem ser ativados pelo usuário quando estes necessitarem de <i>feedbacks</i> sonoros e táteis para melhorar a acessibilidade na exploração do aplicativo.	ESSENCIAL
RNF 11	O aplicativo deverá funcionar em dispositivos móveis com o sistema operacional Android 4.0 e superiores.	ESSENCIAL

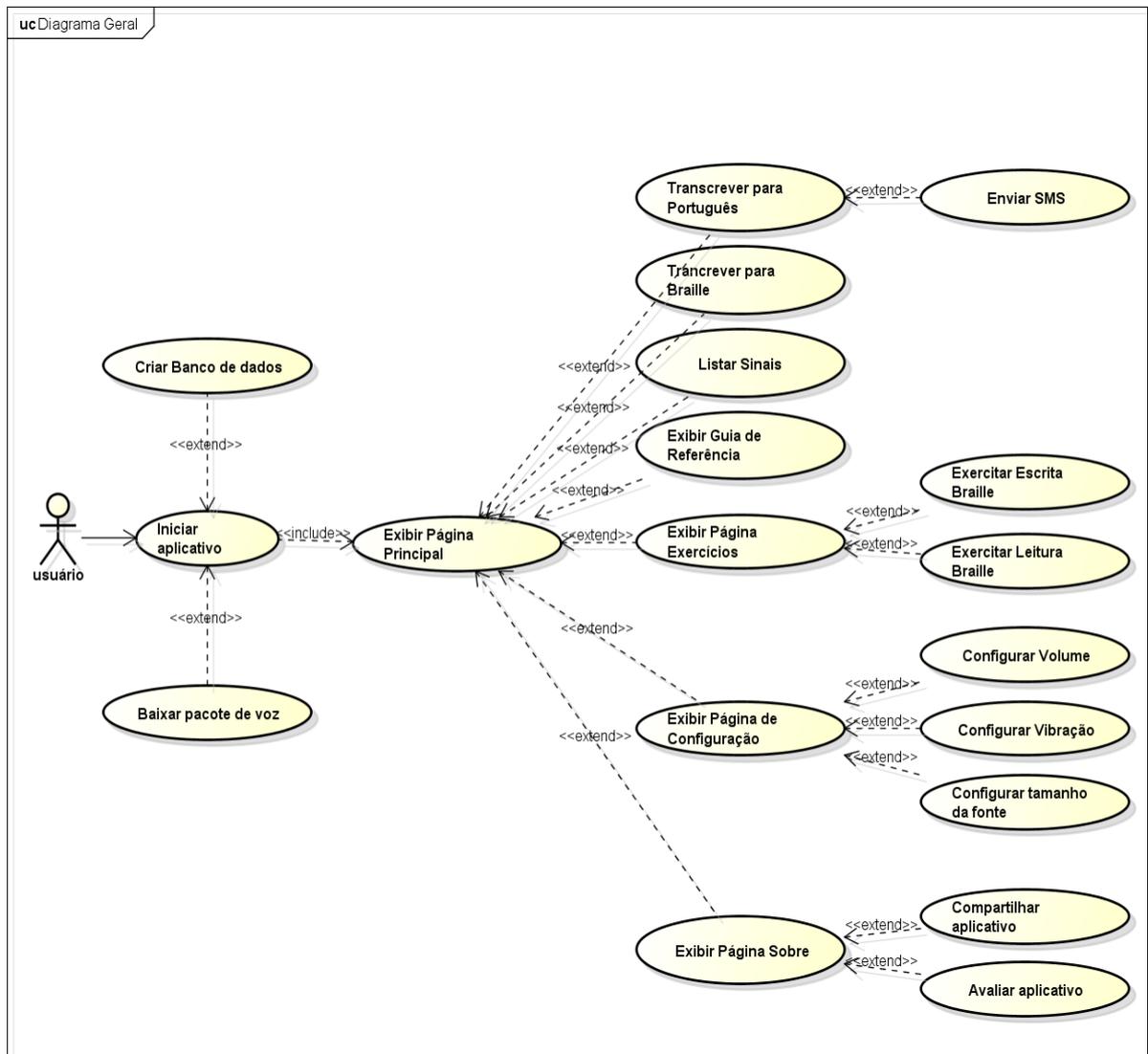
Fonte: Elaborado pelo Autor (2014)

5.1.2 Diagrama e Documentação de Casos de Uso

O diagrama de caso de uso é normalmente utilizado nas etapas de levantamento e análise de requisitos, onde seu objetivo é descrever as funcionalidades do sistema do ponto de vista do usuário, através de uma linguagem simples para possibilitar sua compreensão por

qualquer pessoa (GUEDES, 2009). O diagrama de caso de uso do aplicativo BrailleApp (Figura 16) foi criado com a ferramenta Astah UML (HTTP://astah.net/).

Figura 16: Diagrama de casos de uso do BrailleApp



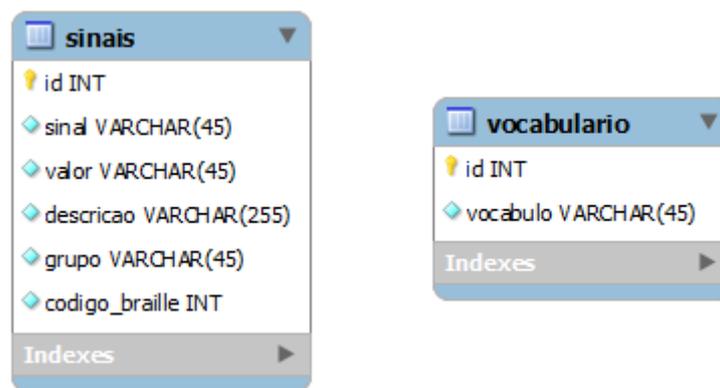
Fonte: Elaborado pelo Autor (2014)

A documentação dos casos de uso descreve de forma mais detalhada a função do caso de uso, normalmente, por meio de uma linguagem simples que possa ser compreendida por um leigo. Segundo Guedes (2009), a documentação de um caso de uso também fornece, geralmente, informações como quais atores estão envolvidos, quais etapas os atores e o sistema devem executar para realizar sua função e quais validações e restrições o caso de uso deve obedecer. A documentação dos casos de uso do BrailleApp estão descritas no Apêndice A.

5.1.3 Diagrama ER do Banco de Dados

O diagrama ER (entidade relacionamento) é um documento que descreve o modelo de dados de um sistema, por meio de representações gráficas de forma intuitivas, onde devem ser apresentadas todas as entidades (tabelas) e o relacionamento entre elas de modo preciso, não ambíguo (HEUSER, 1998). O aplicativo BrailleApp possui duas tabelas (Figura 17) que são usadas apenas para consulta, são elas: “sinais”, que armazena as informações sobre os sinais braille, e “vocabulario”, que armazena os vocábulos que são utilizadas na opção “Exercícios” do aplicativo.

Figura 17: Diagrama ER do aplicativo BrailleApp



Fonte: Elaborada pelo Autor (2014)

5.1.4 Interface de usuário

A interface de usuário é a parte visível de um *software*, por meio da qual o usuário interage com o sistema e realiza suas tarefas. Desenvolver uma interface de usuário “amigável” é de fundamental importância para que se obtenha um nível de usabilidade satisfatório. Segundo Souza (2009), a usabilidade, em sistemas de informação, está relacionada diretamente a qualidade da interação do sistema com os usuários e depende de diversos aspectos como: Facilidade de uso; Facilidade de aprendizado; Satisfação do usuário; Flexibilidade; e Produtividade.

A interface do aplicativo BrailleApp foi desenvolvida sobre os padrões de desenvolvimento definidos por Android Developer (2014). Esses padrões definem características do aplicativo que envolve questões como a estrutura, *layout*, navegação, barra de ação, notificações e acessibilidade, dentre outros. Ela foi modelada, criada e testada principalmente no tocante a acessibilidade, de forma que, o usuário com deficiência visual possa a utilizar sem muita dificuldade e restrição, ao ativar os recursos de acessibilidade do

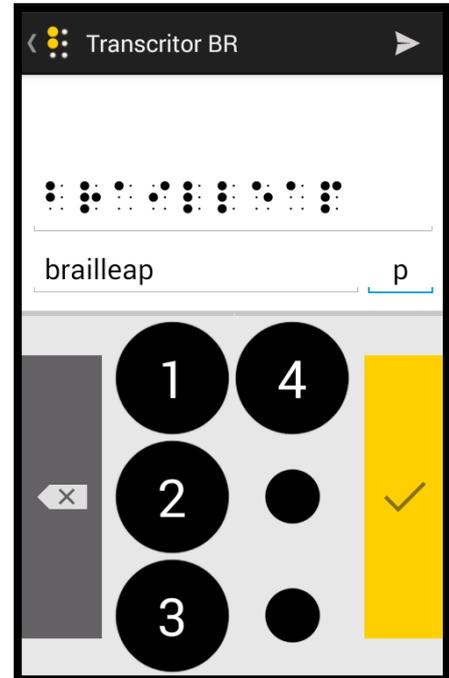
Android. A seguir são ilustradas as principais telas do aplicativo BrailleApp, as demais telas são exibidas no Apêndice B.

Figura 18: Tela principal



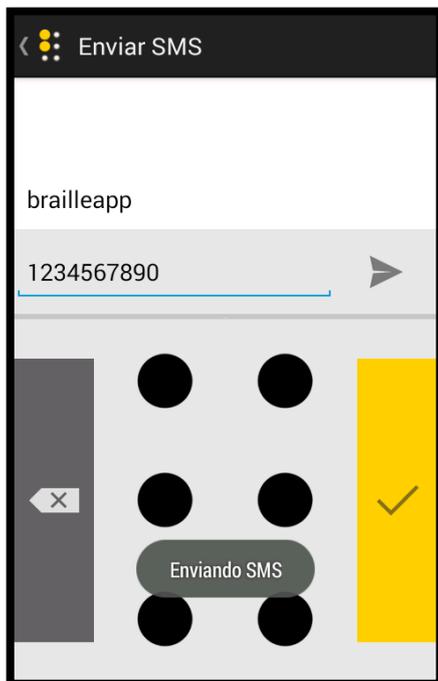
Fonte: Elaborada pelo Autor (2014)

Figura 19: Transcritor Braille / Português



Fonte: Elaborada pelo Autor (2014)

Figura 20: Enviar SMS



Fonte: Elaborada pelo Autor (2014)

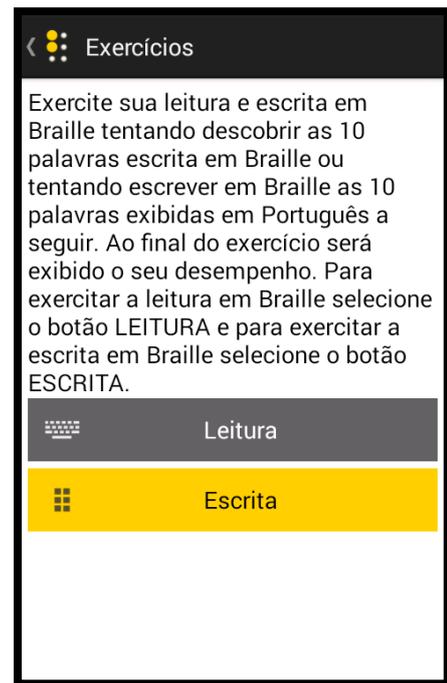
Figura 21: Transcritor Português / Braille



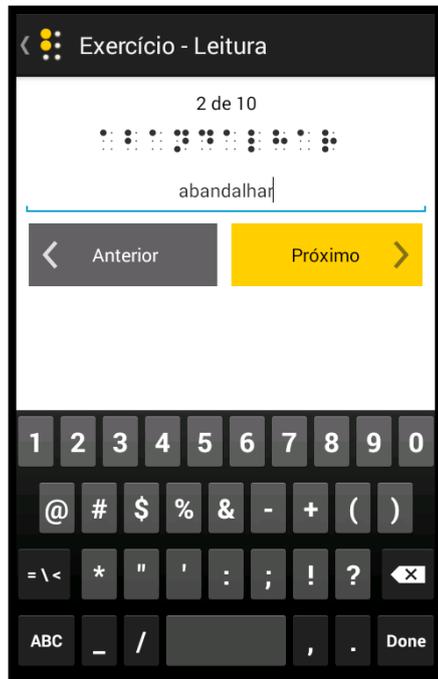
Fonte: Elaborada pelo Autor (2014)

Figura 22: Lista de Sinais

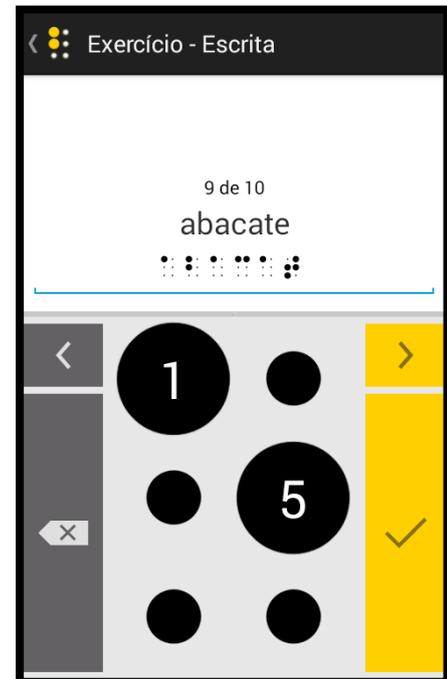
Fonte: Elaborada pelo Autor (2014)

Figura 23: Exercícios

Fonte: Elaborada pelo Autor (2014)

Figura 24: Exercício de leitura Braille

Fonte: Elaborada pelo Autor (2014)

Figura 25: Exercício de escrita Braille

Fonte: Elaborada pelo Autor (2014)

6. RESULTADOS

Em uma análise simples e empírica, pode-se afirmar que o desenvolvimento do aplicativo BrailleApp está em conformidade às funcionalidades e requisitos propostos, descritas na sua documentação, em especial, na implementação das principais normas, regras de aplicação e transcrição da grafia Braile para língua portuguesa definidas no documento normatizador em Cerqueira (2006), além de atender aos requisitos de um *Software* educativo definidos por Oliveira et al (2001, p.74).

Todos os testes de acessibilidade recomendados por Android Developer (2014), descritos na seção 2.2.3, foram realizados e o resultado foi satisfatório, tendo em vista que todos os requisitos de acessibilidade foram atendidos. Porém, isso não garante que o aplicativo esteja totalmente acessível.

Para uma avaliação mais precisa, não só em termos de acessibilidade, é imprescindível à opinião dos usuários finais. Para isso, em complemento a este trabalho, pretende-se coletar, por meio do mecanismo de avaliação de aplicativos da loja do *Google Play*, quando o BrailleApp estiver disponível na loja, a opinião (avaliação) dos usuários, e realizar testes de acessibilidade e usabilidade com um grupo de deficientes visuais.

7. CONSIDERAÇÕES FINAIS

O desenvolvimento deste trabalho foi possível devido aos conhecimentos adquiridos durante toda a graduação do curso de Ciência da Computação, além de experiências profissionais na área e de muita pesquisa e estudo sobre as ferramentas e tecnologias adotadas na sua implementação. Logo, foi desenvolvido neste trabalho o aplicativo BrailleApp atendendo a todos os seus requisitos estabelecidos, inclusive os de acessibilidade.

Apesar do BrailleApp não ter sido avaliado pelos usuários finais, foram feitas simulações durante toda a fase de desenvolvimento, que incluíram testes de acessibilidade, para torná-la uma ferramenta fácil de usar, que incentiva a interação do usuário com o aplicativo, que adota uma metodologia que facilita seu entendimento a partir de uma fundamentação teórica bem estruturada, e que é acessível a todas as pessoas independentemente das suas limitações.

Acredita-se que o uso do BrailleApp auxiliará os usuários, incluindo os com deficiência, no processo de ensino-aprendizagem no sistema Braille de forma satisfatória e contribuirá na inserção social dos deficientes visuais, estimulando sua interação com videntes e proporcionando-lhes mais independência e qualidade de vida.

7.1 TRABALHOS FUTUROS

Como trabalhos futuros, pretendesse melhorar as funcionalidades existentes no aplicativo BrailleApp, em especial a “Guia de Referência”, e adicionar novos utilitários para deficientes visuais. Os principais utilitários são:

- Teclado Braille virtual: Um serviço para entrada de dados que pode ser usado por qualquer aplicação no Android. Quando uma aplicação solicitar ao sistema operacional um recurso de entrada de dados, o “Teclado Braille virtual” poderá ser utilizado para realizar essa tarefa. Seu *layout* poderá ser configurado de forma semelhante à ferramenta *reglete* ou a uma máquina de escrever Perkins.
- Editor de texto Braille: Um editor de texto que permitirá criar, editar, salvar e excluir arquivos de texto, através do “Teclado Braille virtual”.
- Desenvolver uma versão *Web* do BrailleApp.
- Tabuleiro Braille: Dispositivo que possibilite a leitura de livros digitais de forma tátil.

REFERÊNCIAS

ANDROID DEVELOPER. Disponível em: <<http://developer.android.com/>>. Acesso em: 14 de abril de 2014.

ALECRIM, Emerson. **Máquina Virtual Java (Java Virtual Machine).** VIVA AO LINUX, 2005. Disponível em: <[http://www.vivaolinux.com.br/artigo/Maquina-Virtual-Java-\(Java-Virtual-Machine\)](http://www.vivaolinux.com.br/artigo/Maquina-Virtual-Java-(Java-Virtual-Machine))>. Acesso em: 14 de maio de 2014.

BEAR'S, Jacci Howard. **TrueType Fonts.** Disponível em: <<http://desktoppub.about.com/cs/basic/g/truetypefonts.htm>>. Acesso em: 17 de agosto de 2014.

BONNINGTON, Christina. **Touchscreen Braille Writer permite cegos escreverem em Tablet.** Disponível em: <<http://www.wired.com/2011/10/touchscreen-braille-writer/>>. Acesso em: 21 de agosto de 2014.

BRAILLETECH. Disponível em: <<http://brailletouchapp.com/>>. Acesso em: 22 de agosto de 2014.

Brasil. Ministério da Educação. Secretaria de Educação Especial. **Grafia Braille para a Língua Portuguesa** / elaboração : CERQUEIRA, Jonir Bechara... [et al.]. Secretaria de Educação Especial. Brasília: SEESP, 2006.

BRUNO, Marilda Moraes Garcia. **Educação infantil: saberes e práticas da inclusão: dificuldades de comunicação sinalização: deficiência visual.** [4. ed.] / elaboração prof^a Marilda Moraes Garcia Bruno – consultora autônoma. – Brasília: MEC, Secretaria de Educação Especial, 2006.

CARLETTO, Ana Cláudia e GAMBIAGHI, Silvana. **Desenho Universal, um conceito para todos.** Disponível em: <http://www.rinam.com.br/files/REFERENCIAS_DesenhoUniversalumconceitoparatodos.pdf>. Acesso em: 17 de junho de 2014.

D-pad. Disponível em: <<http://en.wikipedia.org/wiki/D-pad>>. Acesso em: 22 de agosto de 2014.

FOLDOC. **API**. Disponível em: <<http://foldoc.org/Application+Program+Interface>>. Acesso em: 22 de agosto de 2014.

GUEDES, G. T. A. **UML2 uma abordagem prática**. 1ª ed. São Paulo -SP: Novatec, 2009.

HEUSER, Carlos Alberto. **Projeto de Banco de Dados**. 1998.

PEREIRA, Lúcio Camilo Oliva. **Android para desenvolvedores** / Lúcio Camilo Oliva, Michel Lourenço da Silva, Rio de Janeiro: Brasport, 2009.

SAC, Sociedade de Assistência aos Cegos. **O SISTEMA BRAILLE**. Disponível em: <http://www.sac.org.br/APR_BR2.htm>. Acesso em: 15 de maio de 2014.

TECE. Tecnologia e Ciência Educacional Ltda. **Manual de uso de regletes**. Disponível em: <<http://www.tece.com.br>>. Acesso em: 14 de junho de 2014.

IDC, International Data Corporation. Disponível em:

<<http://www.idc.com/getdoc.jsp?containerId=prUS24676414>>. Acesso em: 15 de maio de 2014.

INFOESCOLA, Navegando e aprendendo. **Google**. Disponível em:

<<http://www.infoescola.com/empresas/google/>>. Acesso em: 25 de junho de 2014.

KIREV, Nikola. Disponível em: <<http://nikolakirev.com/>>. Acesso em: 22 de agosto de 2014.

LECHETA, Ricardo R. Google Android: **Aprenda a criar aplicações para dispositivos móveis com Android SDK** / Ricador R. Lecheta. – 2. Ed. Ver. e ampl. – São Paulo: Novatec Editora, 2010.

OLIVEIRA, Celina Couto; José Wilson da Costa, Mércia Moreira. **Ambientes Informatizados de Aprendizagem: Produção e Avaliação de software Educativo**. Campinas, SP: Papyrus, 2001.

RODRÍGUEZ, Aurora. **Arquitetura do Anrdoid**. Disponível em:

<<http://androideity.com/2011/07/07/la-maquina-virtual-dalvik/>>. Acesso em: 16 de junho de 2014.

SOMMERVILLE, Ian. **Engenharia de Software - 8ª edição**. ed. São Paulo, SP – Brasil: Pearson Education do Brasil, 2007.

SOUZA, C.S. **Projeto de interfaces de usuário: perspectivas cognitiva e semiótica**. In: XIX Congresso da Sociedade Brasileira de Computação, 1999, Rio de Janeiro. Anais. Rio de Janeiro. Julho de 1999.

TELECO, Inteligência em Telecomunicações. **Middleware**. Disponível em: <<http://www.teleco.com.br/>>. Acesso em: 20 de agosto de 2014.

TIOBE. Disponível em:

<<http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html>>. Acesso em: 12 de abril de 2014.

Trackball. Disponível em: <<http://pt.wikipedia.org/wiki/Trackball>>. Acesso em: 22 de agosto de 2014.

VIVA AO LINUX. **O que é Linux**. Disponível em: <<http://www.vivaolinux.com.br/>>. Acesso em: 21 de agosto de 2014.

APÊNDICE A – DOCUMENTAÇÃO DOS CASOS DE USO

Quadro 3: Iniciar aplicativo

Nome do Caso de Uso	Iniciar aplicativo.
Ator Principal	Usuário.
Resumo	Descreve as etapas para iniciar o aplicativo.
Pré-condição	É preciso que o aplicativo esteja instalado no dispositivo.
Pós-condição	Exibir a tela principal com as opções do aplicativo.
FLUXO PRINCIPAL	
Ações do Ator	Ações do Sistema
1. O usuário inicia o aplicativo.	
	2. O sistema verifica se banco de dados já foi criado.
	3. Caso o banco de dados não tenha sido criado ainda, é executado um <i>script</i> para cria-lo e popular suas tabelas.
	4. O usuário é redirecionado para a tela principal do aplicativo.
	5. O sistema verifica se o pacote de voz <i>TTS</i> português está instalado no dispositivo.
	6. Se o pacote de voz <i>TTS</i> português não estiver instalado, o sistema mostra um <i>pop-up</i> informando a necessidade em instalá-lo e as opções para realizar o seu <i>download</i> ou de cancelar o <i>pop-up</i> .
7. O usuário seleciona a opção de <i>download</i> .	
	8. O aplicativo é encerrado e o sistema operacional

	chama um aplicativo do dispositivo que permita acesso à loja da <i>Google Play</i> e o redirecionada para a mesma para realizar o <i>download</i> .
FLUXO ALTERNATIVO I	
1. O usuário inicia o aplicativo.	
	2. O sistema verifica se o banco de dados já foi criado.
	3. Caso o banco de dados não tenha sido criado ainda, é executado um <i>script</i> para cria-lo e popular suas tabelas.
	4. O usuário é redirecionado para a tela principal do aplicativo.
	5. O sistema verifica se o pacote de voz <i>TTS</i> português está instalado no dispositivo.
	6. Se o pacote de voz <i>TTS</i> português não estiver instalado, o sistema mostra um <i>pop-up</i> informando a necessidade em instalá-lo e as opções para realizar o seu <i>download</i> ou a de cancelar o <i>pop-up</i> .
7. O usuário cancela o <i>pop-up</i> .	
	8. O sistema continua na página principal.
FLUXO ALTERNATIVO II	
1. O usuário inicia o aplicativo.	
	2. O sistema verifica se banco de dados já foi criado.
	3. Caso o banco de dados não tenha sido criado ainda, é executado um <i>script</i> para cria-lo e popular suas tabelas.
	4. O usuário é redirecionado para a tela principal do aplicativo.

	5. O sistema verifica se o pacote de voz <i>TTS</i> português está instalado no dispositivo.
	6. Se o pacote de voz <i>TTS</i> português já estiver instalado no dispositivo, o sistema continua na página principal.
FLUXO ALTERNATIVO III	
1. O usuário inicia o aplicativo.	
	2. O sistema verifica se banco de dados já foi criado.
	3. Caso o banco de dados já tenha sido criado, o usuário é redirecionado para a tela principal do aplicativo.
	4. O sistema verifica se o pacote de voz <i>TTS</i> português está instalado no dispositivo.
	5. Se o pacote de voz <i>TTS</i> português não estiver instalado, o sistema mostra um <i>pop-up</i> informando a necessidade em instalá-lo e as opções para realizar o seu <i>download</i> ou de cancelar o <i>pop-up</i> .
6. O usuário seleciona a opção de <i>download</i> .	
	7. O aplicativo é encerrado e o sistema operacional chama um aplicativo do dispositivo que permita acesso ao <i>Google Play</i> e o redirecionada para o mesmo para realizar o <i>download</i> .
FLUXO ALTERNATIVO IV	
1. O usuário inicia o aplicativo.	
	2. O sistema verifica se banco de dados já foi criado.
	3. Caso o banco de dados já tenha sido criado, o usuário é redirecionado para a tela principal do aplicativo.
	4. O sistema verifica se o pacote de voz <i>TTS</i> português

	está instalado no dispositivo.
	5. Se o pacote de voz <i>TTS</i> português não estiver instalado, o sistema mostra um <i>pop-up</i> informando a necessidade em instalá-lo e as opções para realizar o seu <i>download</i> ou a de cancelar o <i>pop-up</i> .
6. O usuário cancela o <i>pop-up</i> .	
	7. O sistema continua na página principal.
FLUXO ALTERNATIVO V	
1. O usuário inicia o aplicativo.	
	2. O sistema verifica se banco de dados já foi criado.
	3. Caso o banco de dados já tenha sido criado, o usuário é redirecionado para a tela principal do aplicativo.
	4. O sistema verifica se o pacote de voz <i>TTS</i> português está instalado no dispositivo.
	5. Se o pacote de voz <i>TTS</i> português já estiver instalado no dispositivo, o sistema continua na página principal.

Quadro 4: Transcrever texto em Braille para Português

Nome do Caso de Uso	Transcrever texto em Braille para Português.
Ator Principal	Usuário.
Resumo	Descreve as etapas para transcrever texto em Braille para Português.
Pré-condição	É preciso que o usuário esteja na tela principal do aplicativo.
FLUXO PRINCIPAL	
Ações do Ator	Ações do Sistema
1. O usuário seleciona a opção	

transcrever texto em Braille para Português.	
	2. É apresentada uma tela com os campos de saída de texto “texto em Braille”, “texto em Português” e “caractere corrente”, um botão para adicionar e outro para remover sinais, uma célula Braille virtual para entrada de dados, e um botão na barra de ação para enviar SMS.
3. O usuário toca nos pontos da célula Braille virtual para formar os sinais correspondentes ao(s) caractere(s) desejado(s).	
	4. O Sistema verifica o código Braille do sinal formado e retorna o seu caractere (em Português) correspondente, o exibindo no campo “caractere corrente”, se existir.
	5. Se o pacote de voz <i>TTS</i> português estiver instalado e o volume da aplicação não estiver em “mudo”, o número correspondente a cada ponto é falado ao ser selecionado.
6. O usuário aperta o botão “adicionar sinal”	
	7. Se o sinal formado existir, ele é adicionado à lista de sinais do campo “texto em Braille” e verificado qual o seu caractere correspondente (em Português) e o adiciona no campo “texto em Português”.
	8. Se o pacote de voz <i>TTS</i> português estiver instalado e o volume da aplicação não estiver em “mudo”, o caractere (corrente) é falado.
FLUXO ALTERNATIVO I	

1. O usuário seleciona a opção transcrever texto em Braille para Português.	
	2. É apresentada uma tela com os campos de saída de texto “texto em Braille”, “texto em Português” e “caractere corrente”, um botão para adicionar e outro para remover sinais, uma célula Braille virtual para entrada de dados, e um botão na barra de ação para enviar SMS.
3. O usuário toca nos pontos da célula Braille virtual para formar os sinais correspondentes ao(s) caractere(s) desejado(s).	
	4. O Sistema verifica o código Braille do sinal formado e retorna o seu caractere (em Português) correspondente, o exibindo no campo “caractere corrente”, se existir.
	5. Se o pacote de voz <i>TTS</i> português estiver instalado e o volume da aplicação não estiver em “mudo”, o número correspondente a cada ponto é falado ao ser selecionado.
6. O usuário aperta o botão “adicionar sinal”	
	7. Se o sinal formado não existir, ele não é adicionado e um alerta visual é emitido.
	8. Se o pacote de voz <i>TTS</i> português estiver instalado e o volume da aplicação não estiver em “mudo”, o texto do alerta visual é falado.

FLUXO ALTERNATIVO II

1. O usuário seleciona a opção transcrever texto em Braille para Português.	
	2. É apresentada uma tela com os campos de saída de texto “texto em Braille”, “texto em Português” e “caractere corrente”, um botão para adicionar e outro para remover sinais, uma célula Braille virtual para entrada de dados, e um botão na barra de ação para enviar SMS.
3. O usuário aperta o botão “remover sinal”.	
	4. O último sinal adicionado é removido, se existir.
FLUXO ALTERNATIVO III	
1. O usuário seleciona a opção transcrever texto em Braille para Português.	
	2. É apresentada uma tela com os campos de saída de texto “texto em Braille”, “texto em Português” e “caractere corrente”, um botão para adicionar e outro para remover sinais, uma célula Braille virtual para entrada de dados, e um botão na barra de ação para enviar SMS.
3. O usuário aperta por longo tempo o botão “remover sinal”.	
	4. Todos os sinais são removidos.
FLUXO ALTERNATIVO IV	
1. O usuário seleciona a opção transcrever texto em Braille para Português.	

	2. É apresentada uma tela com os campos de saída de texto “texto em Braille”, “texto em Português” e “caractere corrente”, um botão para adicionar e outro para remover sinais, uma célula Braille virtual para entrada de dados, e um botão na barra de ação para enviar SMS.
3. O usuário aperta botão “SMS”.	
	4. O sistema redireciona o usuário para página de Enviar SMS.

Quadro 5: Enviar SMS

Nome do Caso de Uso	Enviar SMS.
Ator Principal	Usuário.
Resumo	Descreve as etapas para enviar SMS.
Pré-condição	É preciso que o usuário tenha entrado página de transcrever texto em Braille para Português, inserido algum texto e pressionado o botão “SMS”.
FLUXO PRINCIPAL	
Ações do Ator	Ações do Sistema
	1. É apresentada uma tela com a mensagem a ser enviada, o “destinatário”, botão de “Enviar SMS” e uma célula Braille virtual para entrada de dados (número do destinatário).
2. O usuário toca nos pontos da célula Braille virtual para formar os sinais correspondentes ao(s) número(s) desejado(s).	

	3. Se o pacote de voz <i>TTS</i> português estiver instalado e o volume da aplicação não estiver em “mudo”, o número correspondente a cada ponto é falado ao ser selecionado.
4. O usuário aperta o botão “adicionar sinal”	
	5. Se o sinal formado corresponder a um número, é verificado qual o seu número correspondente e o adiciona no campo “destinatário”.
FLUXO ALTERNATIVO I	
	1. É apresentada uma tela com a mensagem a ser enviada, o “destinatário”, botão de “Enviar SMS” e uma célula Braille virtual para entrada de dados (número do destinatário).
2. O usuário toca nos pontos da célula Braille virtual para formar os sinais correspondentes ao(s) número(s) desejado(s).	
	3. Se o pacote de voz <i>TTS</i> português estiver instalado e o volume da aplicação não estiver em “mudo”, o número correspondente a cada ponto é falado ao ser selecionado.
4. O usuário aperta o botão “Adicionar sinal”	
	5. Se o sinal formado não corresponder a um número ou não existir, ele não é adicionado e um alerta visual é emitido.
	6. Se o pacote de voz <i>TTS</i> português estiver instalado e o volume da aplicação não estiver em “mudo”, o texto

	do alerta visual é falado.
FLUXO ALTERNATIVO II	
	1. É apresentada uma tela com a mensagem a ser enviada, o “destinatário”, botão de “Enviar SMS” e uma célula Braille virtual para entrada de dados (número do destinatário).
2. O usuário pressiona o botão “Enviar SMS”	
	3. A mensagem é enviada.
	4. Caso o ocorra algum erro, é emitido um alerta visual e sonoro (se o pacote de voz <i>TTS</i> português estiver instalado e o volume da aplicação não estiver em “mudo”).

Quadro 6: Transcrever texto em Português para Braille

Nome do Caso de Uso	Transcrever texto em Português para Braille.
Ator Principal	Usuário.
Resumo	Descreve as etapas para transcrever texto em Português para Braille.
Pré-condição	É preciso que o usuário esteja na tela principal do aplicativo.
FLUXO PRINCIPAL	
Ações do Ator	Ações do Sistema
5. O usuário seleciona a opção transcrever texto em Português para Braille.	
	6. É apresentada uma tela com os campos de saída de texto “texto em Braille”, um campo de entrada de texto em Português e um botão de “limpar tudo”.

7. O usuário insere o texto	
	8. O Sistema transcreve o texto inserido automaticamente para Braille e os exibe no campo “texto em Braille”.
FLUXO ALTERNATIVO I	
1. O usuário seleciona a opção transcrever texto em Português para Braille.	
	2. É apresentada uma tela com os campos de saída de texto “texto em Braille”, um campo de entrada de texto em Português e um botão de “limpar tudo”.
3. O usuário aperta o botão “limpar”	
	4. O texto e os sinais são apagados.

Quadro 7: Listar sinais

Nome do Caso de Uso	Listar sinais.
Ator Principal	Usuário.
Resumo	Descreve as etapas para listar os sinais Braille e seus significados.
Pré-condição	É preciso que o usuário esteja na tela principal do aplicativo.
FLUXO PRINCIPAL	
Ações do Ator	Ações do Sistema
1. O usuário seleciona a opção “Lista de sinais”.	
	2. O sistema busca no banco de dados todos os sinais (símbolo, valor e descrição).

	3. É apresentada uma tela com a lista dos sinais braile separada por categorias.
4. O usuário pressiona um item da lista.	
	5. Se o pacote de voz <i>TTS</i> português estiver instalado e o volume da aplicação não estiver em “mudo”, a descrição do sinal é falada.
6. O usuário pressiona por longo tempo um item da lista.	
	7. Um alerta visual é exibido com código braile do sinal correspondente.
	8. Se o pacote de voz <i>TTS</i> português estiver instalado e o volume da aplicação não estiver em “mudo”, o código braile do sinal é falado.

Quadro 8: Exibir Guia de Referência

Nome do Caso de Uso	Exibir Guia de Referência.
Ator Principal	Usuário.
Resumo	Descreve as etapas para exibir o Guia de Referência sobre o Braille.
Pré-condição	É preciso que o usuário esteja na tela principal do aplicativo.
FLUXO PRINCIPAL	
Ações do Ator	Ações do Sistema
1. O usuário seleciona a opção “Guia de Referência”.	
	2. É apresentada uma tela com o texto que descreve o a grafia Braille para língua portuguesa.

Quadro 9: Exercitar a leitura Braille.

Nome do Caso de Uso	Exercitar Leitura Braille.
Ator Principal	Usuário.
Resumo	Descreve as etapas para exercitar a leitura Braille.
Pré-condição	É preciso que o usuário esteja na tela principal do aplicativo.
FLUXO PRINCIPAL	
Ações do Ator	Ações do Sistema
1. O usuário seleciona a opção “Exercícios”.	
	2. É apresentada a tela de “Exercícios”, onde é exibido um texto descritivo sobre os exercícios de Leitura e Escrita Braille e duas botões de opções, de leitura e de escrita.
3. O usuário seleciona a opção de “Leitura”	
	4. O sistema busca no banco de dados 10 palavras aleatórias para apresentar no exercício.
	5. É apresentada a tela de “Exercício - Leitura”, onde é exibido um texto em braile, um campo de entrada de texto (em português) e botões de avançar e retroceder.
6. O usuário tenta adivinhar qual a palavra escrita em braile, das 10 (dez) palavras do exercício, escrevendo seu correspondente em português.	
7. O usuário pressiona o botão “avançar” para ir para a	

próxima palavra.	
	8. No final do exercício o sistema verifica quais palavras o usuário acertou e errou e as exibem, palavras “originais” e as escritas pelo usuário.

Quadro 10: Exercitar a escrita Braille

Nome do Caso de Uso	Exercitar Escrita Braille.
Ator Principal	Usuário.
Resumo	Descreve as etapas para exercitar a escrita Braille.
Pré-condição	É preciso que o usuário esteja na tela principal do aplicativo.
FLUXO PRINCIPAL	
Ações do Ator	Ações do Sistema
1. O usuário seleciona a opção “Exercícios”.	
	2. É apresentada a tela de “Exercícios”, onde é exibido um texto descritivo sobre os exercícios de Leitura e Escrita Braille e duas botões de opções, de leitura e de escrita.
3. O usuário seleciona a opção de “Escrita”	
	4. O sistema busca no banco de dados 10 palavras aleatórias para apresentar no exercício.
	5. É apresentada a tela de “Exercício - Escrita”, onde é exibido um texto em português, um campo de texto (em braille), uma célula Braille para entrada de dados, e botões de avançar e retroceder.
6. O usuário tenta transcrever para Braille as 10 (dez)	

palavras do exercício, escritas em português, utilizando como entrada de dados a célula braile.	
7. O usuário pressiona o botão “avançar” para ir para próxima palavra.	
	8. No final do exercício o sistema verifica quais palavras o usuário acertou e errou e as exibem, palavras “originais” e as escritas pelo usuário.

Quadro 11: Avaliar aplicativo.

Nome do Caso de Uso	Avaliar aplicativo.
Ator Principal	Usuário.
Resumo	Descreve as etapas para avaliar o aplicativo.
Pré-condição	É preciso que o usuário esteja na tela principal do aplicativo.
FLUXO PRINCIPAL	
Ações do Ator	Ações do Sistema
1. O usuário seleciona a opção “Sobre” na barra de ação.	
	2. É apresentada a tela de “Sobre”, onde é exibido um texto descritivo sobre o aplicativo BrailleApp e os botões de “Avaliar” e “Compartilhar” o aplicativo.
3. O usuário seleciona a opção de “Avaliar”	
	4. O sistema operacional chama um aplicativo do dispositivo que permita acesso ao <i>Google Play</i> e o redirecionada para o mesmo para realizar a avaliação

	do aplicativo.
--	----------------

Quadro 12: Compartilhar aplicativo

Nome do Caso de Uso	Compartilhar aplicativo.
Ator Principal	Usuário.
Resumo	Descreve as etapas para compartilhar o aplicativo.
Pré-condição	É preciso que o usuário esteja na tela principal do aplicativo.
FLUXO PRINCIPAL	
Ações do Ator	Ações do Sistema
1. O usuário seleciona a opção “Sobre” na barra de ação.	
	2. É apresentada a tela de “Sobre”, onde é exibido um texto descritivo sobre o aplicativo BrailleApp e os botões de “Avaliar” e “Compartilhar” o aplicativo.
3. O usuário seleciona a opção de “Compartilhar”	
	4. O sistema operacional chama um aplicativo do dispositivo que permita o compartilhamento de texto e passa como mensagem o endereço para <i>download</i> do BrailleApp.

Quadro 13: Configura aplicativo (tamanho da fonte, vibração e volume).

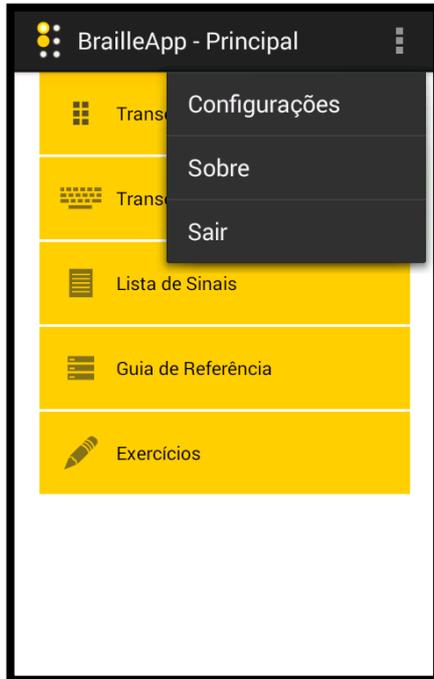
Nome do Caso de Uso	Configurar aplicativo.
Ator Principal	Usuário.
Resumo	Descreve as etapas para configurar o aplicativo.
Pré-condição	É preciso que o usuário esteja na tela principal do aplicativo.

FLUXO PRINCIPAL	
Ações do Ator	Ações do Sistema
1. O usuário seleciona a opção “Configuração” na barra de ação.	
	2. É apresentada a tela de “Configuração”, onde é exibido as opções de “Tamanho da fonte”, “Vibrar” e “Volume”.
3. O usuário seleciona a opção de “Tamanho da fonte”	
	4. O sistema exibe um <i>pop-up</i> com as opções de tamanho de fonte e o botão cancelar.
5. O usuário seleciona o tamanho de fonte.	
	6. O sistema altera o tamanho da fonte do aplicativo, se for diferente da atual, e fecha o <i>pop-up</i> .
FLUXO ALTERNATIVO I	
1. O usuário seleciona a opção “Configuração” na barra de ação.	
	2. É apresentada a tela de “Configuração”, onde são exibidas as opções de “Tamanho da fonte”, “Vibrar” e “Volume”.
3. O usuário marca/desmarca a opção “Vibrar”.	
	4. O sistema configura a opção de vibrar para sim ou não dependendo da opção do usuário.
FLUXO ALTERNATIVO II	

1. O usuário seleciona a opção “Configuração” na barra de ação.	
	2. É apresentada a tela de “Configuração”, onde são exibidas as opções de “Tamanho da fonte”, “Vibrar” e “Volume”.
3. O usuário seleciona a opção “Volume”.	
	4. O sistema exibe um <i>pop-up</i> com as opções de volume e o botão cancelar.
5. O usuário seleciona o volume.	
	6. O sistema altera o volume do aplicativo.
7. O usuário fecha o <i>pop-up</i> , pressionando o botão “cancelar”.	

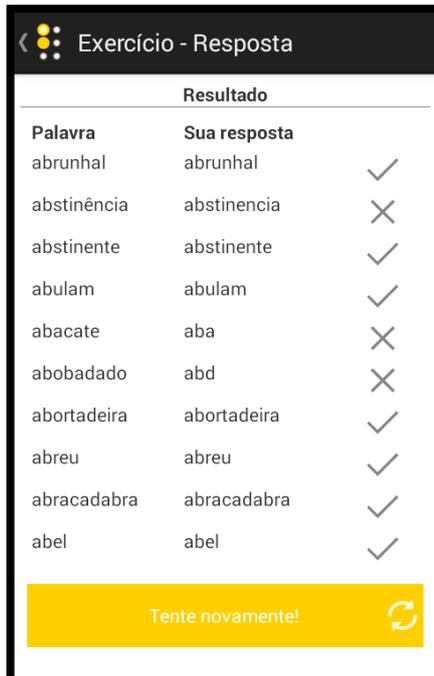
APÊNDICE B – TELAS DO APLICATIVO BRAILLEAPP

Figura 26: Tela principal (2)



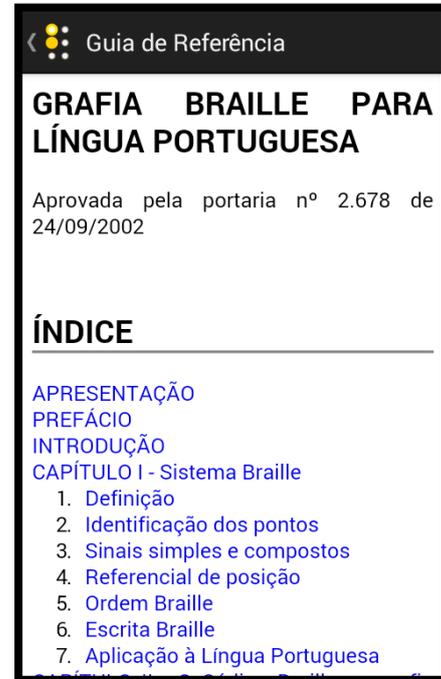
Fonte: Elaborada pelo Autor (2014)

Figura 28: Resultado dos exercícios



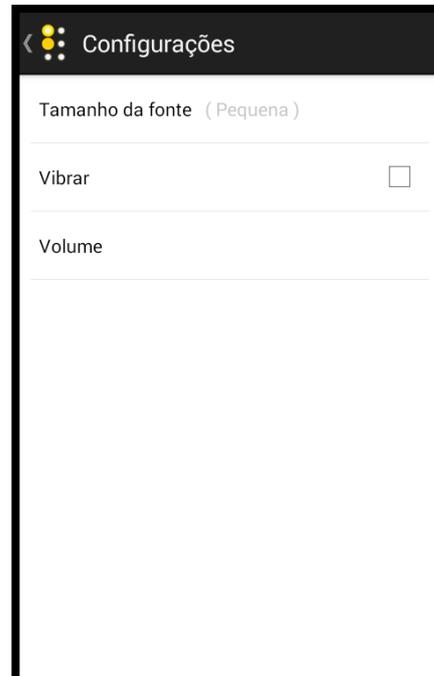
Fonte: Elaborada pelo Autor (2014)

Figura 27: Guia de Referência

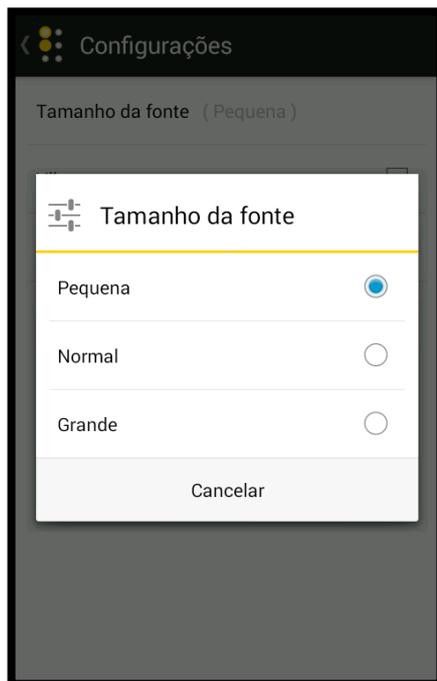


Fonte: Elaborada pelo Autor (2014)

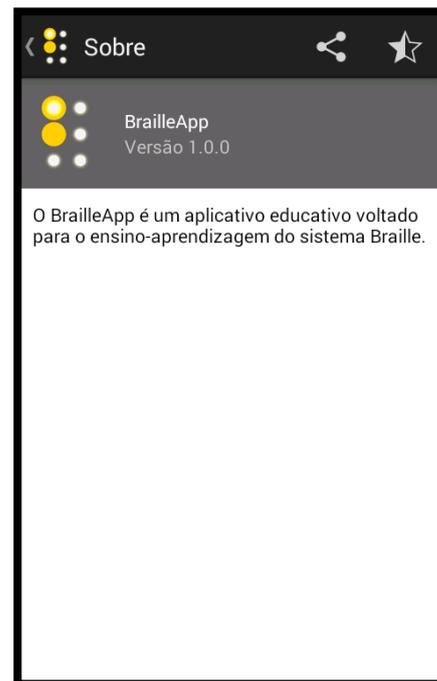
Figura 29: Configurações



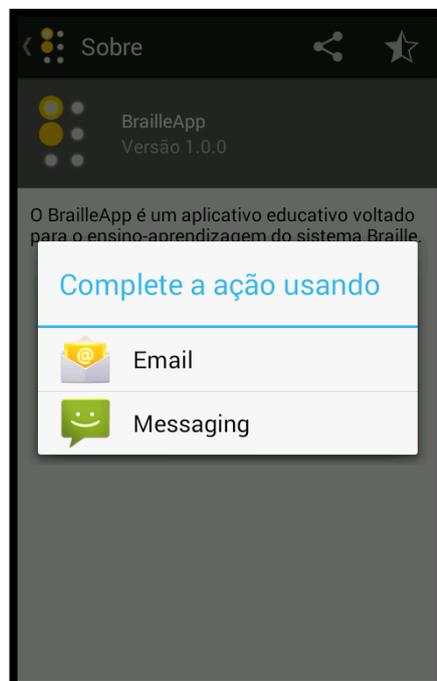
Fonte: Elaborada pelo Autor (2014)

Figura 30: Tamanho da fonte

Fonte: Elaborada pelo Autor (2014)

Figura 31: Sobre

Fonte: Elaborada pelo Autor (2014)

Figura 32: Compartilhar aplicativo

Fonte: Elaborada pelo Autor (2014)