

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE
SISTEMAS**

THIAGO MARTIGNONI

**LEITOR DIGITAL DESENVOLVIDO EM ANDROID PARA USUÁRIOS COM VISÃO
LIMITADA**

TRABALHO DE CONCLUSÃO DE CURSO

PATO BRANCO

2014

THIAGO MARTIGNONI

**LEITOR DIGITAL DESENVOLVIDO EM ANDROID PARA USUARIOS COM VISÃO
LIMITADA**

Monografia apresentada como requisito parcial para obtenção do título de Tecnólogo no Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas da Universidade Tecnológica Federal do Paraná, Campus Pato Branco.
Orientador: Prof. Robison Cris Brito

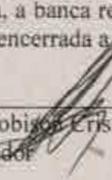
PATO BRANCO-PR

2014

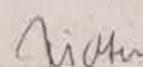
ATA Nº: 251

DEFESA PÚBLICA DO TRABALHO DE DIPLOMAÇÃO DO ALUNO THIAGO MARTIGNONI.

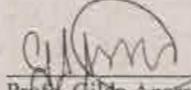
Às 16:30 hrs do dia 17 de dezembro de 2014, Bloco V da UTFPR, Câmpus Pato Branco, reuniu-se a banca avaliadora composta pelos professores Robison Cris Brito (Orientador), Richardson Ribeiro (Convidado) e Gilda Aparecida de Assis (Convidada), para avaliar o Trabalho de Diplomação do aluno Thiago Martignoni, matrícula 1066943, sob o título Leitor Digital Desenvolvido em Android para Usuários com Visão Limitada; como requisito final para a conclusão da disciplina Trabalho de Diplomação do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, COADS. Após a apresentação o candidato foi entrevistado pela banca examinadora, e a palavra foi aberta ao público. Em seguida, a banca reuniu-se para deliberar considerando o trabalho APROVADO. Às 17:05 hrs foi encerrada a sessão.



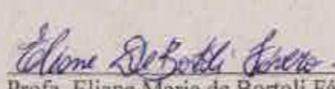
Prof. Robison Cris Brito, M.Sc.
Orientador



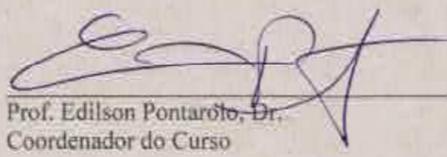
Prof. Richardson Ribeiro, Dr.
Convidado



Prof. Gilda Aparecida de Assis, Dr.
Convidada



Prof. Eliane Maria de Bortoli Fávero, M.Sc
Coordenadora
do Trabalho de Diplomação



Prof. Edilson Pontaroto, Dr.
Coordenador do Curso

RESUMO

MARTIGNONI, Thiago. LEITOR DIGITAL DESENVOLVIDO EM ANDROID PARA USUARIOS COM VISÃO LIMITADA. 2014. 47f. Monografia de Trabalho de Conclusão de Curso - Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, Universidade Tecnológica Federal do Paraná. Pato Branco, 2014.

Hoje, as pessoas que possuem deficiência visual, seja ela total ou parcial, possuem varias dificuldades no dia-a-dia, muitas vezes isso acontece por serem dependentes de outras pessoas, seja para ler, para se locomover ou fazer qualquer outra atividade rotineira. Visando melhorar a qualidade de vida e possibilitar que essas pessoas se tornem mais autônomas, o presente trabalho propõe um aplicativo para a plataforma Android que auxilia pessoas com baixa visão na atividade de leitura, sendo processando o texto capturado pela câmera do device Android para aumenta-lo no display ou através da sintetização deste para o usuário. Desta forma, o próprio usuário pode escolher qual a melhor opção do sistema de acordo com o seu grau de deficiência ou com o texto que deve ser interpretado.

Palavras-chave: Aplicativo Móvel, Android, Acessibilidade, Leitor Digital.

ABSTRACT

MARTIGNONI, Thiago. Digital player developed for Android users with visual impairments. 2014. 47 f. Monografia de Trabalho de Conclusão de Curso - Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, Universidade Tecnológica Federal do Paraná. Pato Branco, 2014.

Today, people who are blind, whether total or partial, have various difficulties in day-to-day, often this happens because they are dependent on others, either to read, to get around or make any other routine activity. To improve the quality of life and enable these people to become more autonomous, this paper proposes an application for the Android platform that helps people with low vision reading activity, and processing the text captured by the camera to Android device increasing it on display or through the synthesis of this to the user. Thus, the user himself can choose the best system option according to their disability or with the text that should be interpreted.

KeyWords: Mobile Application, Android, Accessibility, Digital Reader.

LISTA DE FIGURAS

Figura 1 - Percentual de Deficiência por região	12
Figura 2 - Leitores Digitais	18
Figura 3 - Imagem do Aplicativo Google Play Livros.....	20
Figura 4 - Lupa Eletrônica Pebble HD.....	21
Figura 5 - Diagrama de Caso de Uso.....	28
Figura 6 - Diagrama de Atividade.....	29
Figura 7 - Estrutura Código Fonte	30
Figura 8 - Distribuição Plataforma Android.....	31
Figura 9 - Dispositivo Samsung Galaxy Note 10.1	31
Figura 10 - Plugin Layout ADT	32
Figura 11 - Tela Inicial do Aplicativo.....	32
Figura 12 - Opção de Lupa do Aplicativo	33
Figura 13 - Câmera do Dispositivo Móvel	34
Figura 14 - Captura de Imagem para OCR	34
Figura 15 - Imagem Transformada em texto	35

LISTA DE QUADROS

Quadro 1 - Ferramentas utilizadas no desenvolvimento do aplicativo	24
Quadro 2 - Requisito Funcional - Aplicar Zoom.....	27
Quadro 3 - Requisito Funcional - Transformar Imagem em Texto	27
Quadro 4 - Requisito Funcional - Sintetizar texto	28
Quadro 5 - Requisito Funcional - Ampliar ou Reduzir Texto	28

LISTA DE TABELAS

Tabela 1 - Visão do mercado de sistemas operacionais de smartphones	22
---	----

LISTA DE LISTAGENS

Listagem 1 – Correção de Imagem	36
Listagem 2 – Funcionalidade Lupa.....	37
Listagem 3 – Convertendo imagem.....	38
Listagem 4 – Utilizando a API tesseract.....	39
Listagem 5 – Sintetizando o texto	39

LISTA DE SIGLAS

ADT	Android Developer Tools
API	Application Programming Interface
CBO	Conselho Brasileiro de Oftalmologia
IBM	International Business Machines
IDE	Integrated Development Environment
OCR	Optical Character Recognition
PDF	Portable Document Format
SDK	Source Development Kit
UML	Unified Modeling Language
XML	Extensible Markup Language

SUMÁRIO

1. INTRODUÇÃO	11
1.1. CONSIDERAÇÕES INICIAIS	11
1.2. OBJETIVOS	14
1.2.1. Objetivos Gerais	14
1.2.2. Objetivos Específicos.....	15
1.3. JUSTIFICATIVA	15
1.4. ESTRUTURA DO TRABALHO	16
2. REFERENCIAL TEÓRICO	18
2.1. LEITORES DIGITAIS	18
2.2. LEITORES DIGITAIS PARA DISPOSITIVOS MÓVEIS	19
2.3. LUPAS ELETRÔNICAS	21
2.4. SMARTPHONES ANDROID	22
3. MATERIAIS E MÉTODOS	24
3.1. MATERIAIS	24
3.2. MÉTODOS	25
4. RESULTADOS	27
4.1. MODELAGEM DO SOFTWARE	27
4.2. DESENVOLVIMENTO DO SISTEMA MÓVEL	29
4.2.1 Aplicativo Desenvolvido.....	32
4.2.3 Codificação de funcionalidades.....	36
5. CONCLUSÃO	41
5.1. PROBLEMAS ENCONTRADOS	42
5.2. TRABALHOS FUTUROS	42
6. REFERÊNCIAS	44

1. INTRODUÇÃO

Este capítulo contém as considerações iniciais, assim como os objetivos geral e específico buscados com esse trabalho. A justificativa apresenta os motivos que levaram a escolha deste tema. Ao final do capítulo, é encontrada a estrutura utilizada para escrever o trabalho.

1.1. CONSIDERAÇÕES INICIAIS

O tema acessibilidade, nos dias de hoje, vem ganhando espaço nas mais diversas áreas sejam nas universidades, escolas, clubes ou seminários. Isso se deve a preocupação da maioria dos cidadãos em atender as dificuldades das pessoas com necessidades especiais, para que estas possam interagir naturalmente com a sociedade.

O conceito de acessibilidade começou a surgir em meados da década de 50, com a prática da reintegração de adultos reabilitados nas próprias famílias, no mercado de trabalho ou na comunidade em geral. Profissionais de reabilitação constatavam que essa prática era dificultada e até impedida pela existência de barreiras arquitetônicas nos espaços urbanos, nos edifícios, residências e nos meios de transporte coletivo, surgia assim a fase da integração.

Apesar de estas preocupações serem relativamente antigas, o tema começou a ser amplamente discutido no Brasil após a promulgação da Constituição de 1988, onde o tema foi citado na Carta Magna em seu artigo 5, que garante o direito de ir e vir, e no artigo 227 que diz “A lei disporá sobre normas de construção dos logradouros e dos edifícios de uso público e de fabricação de veículos de transporte coletivo, a fim de garantir acesso adequado as pessoas portadoras de deficiência” (COSTA; MAIOR; LIMA, 2005).

Em 2000 estas preocupações foram transformadas em Leis. A primeira é a lei 10048 que foi elaborada pelo poder Legislativo e trata da prioridade e em acessibilidade das pessoas com deficiência, já a Lei 10098 foi feita pelo poder Executivo onde o assunto é dividido em acessibilidade ao meio físico, meios de transporte, comunicação, informação e ajudas técnicas (COSTA; MAIOR; LIMA, 2005).

Atualmente buscam-se ações que combatam a exclusão aos benefícios da vida em sociedade, que é provocada pela falta de classe social, pela origem geográfica, idade ou existência de necessidades especiais. Esta fase de inclusão objetiva oferecer aos mais necessitados as oportunidades de acesso a bens e serviços, dentro de um sistema que beneficie a todos.

Apesar dos inúmeros incentivos nesta área, hoje os investimentos na inclusão de pessoas com necessidades especiais ainda são pequenos, existindo pessoas e empresas, em especial, que visam otimizar seus lucros, que criticam a igualdade de direitos e não querem cooperar com aqueles que fogem dos padrões de normalidade estabelecidos por um grupo que é a maioria. (SIMS, 2014)

Dentre as dificuldades encontradas no dia-a-dia pelas pessoas que possuem alguma necessidade especial está o preconceito e a falta de interesse da população para que essas pessoas sejam incluídas na sociedade. Neste contexto, pode-se citar as pessoas com deficiências visuais, que hoje ainda encontram muita dificuldade para se manterem informadas através de documentos impressos, como jornais, revistas e materiais de divulgação.

A Figura 1 apresenta o percentual de pessoas com deficiência visual no Brasil, fragmentando estes dados também por região. Observa-se que os portadores de deficiência visual são maioria no Brasil. Denomina-se deficiência visual pessoas com algum problema na visão que não podem ser corrigidas com o uso de lente.



Figura 1 - Percentual de Deficiência por região

Fonte: (Nauar, 2014)

Entre as principais limitações encontradas pelas pessoas com deficiência visual está a falta de sinalização sonora nas vias públicas, a dificuldade encontrada para o uso do transporte coletivo, a falta de informação adaptada, além da falta de oportunidade de empregos.

Dados do Conselho Brasileiro de Oftalmologia (CBO) revelam que 30% das crianças em idade escolar e mais de 90% dos adultos com mais de 40 anos apresentam problemas oculares que interferem em seu desempenho diário (MINISTÉRIO DA SAÚDE, 2007).

Nos últimos anos, pesquisas apontam que a média de idade dos brasileiros tem aumentado o que reflete diretamente na quantidade de pessoas que possuem problemas de visão. Estima-se que a visão humana tenha um “tempo de vida” de 40 anos, assim, após esta idade é comum que as pessoas usem lentes corretivas. (IBGE, 2008)

Com o advento das novas tecnologias para o processamento de informação (*hardware* ou *software*) é possível melhorar a acessibilidade das pessoas que apresentem qualquer nível de deficiência visual, podendo ser utilizado, além dos computadores, os *smartphones* e também os *tablets*, permitindo a melhora na qualidade de vida das pessoas com deficiência visual, seja na obtenção da informação, como na comunicação.

Sabe-se que a evolução e a popularização dos dispositivos móveis vêm crescendo nos últimos anos. Esta evolução começou em 1993 com um protótipo de *smartphone* chamado “Simon” , este desenvolvido pela IBM (International Business Machines). A ideia era unir o celular e o computador para permitir que os usuários conseguissem trocar mensagens com computadores através do dispositivo. Na época, o protótipo contava com algumas funcionalidades, como calendário, agenda, hora mundial e permitia o acesso aos e-mails. (CANALTECH, 2012).

A grande revolução dos dispositivos móveis veio 14 anos depois, com a empresa Apple, que apresentou a primeira geração do iPhone. Um ano depois e com a popularização destes *smartphones*, a empresa Google lançou seu próprio sistema operacional para dispositivos móveis de baixo custo, auxiliando na popularização dos *smartphones*.

Juntamente com a popularização do dispositivo móvel, a concorrência entre os fabricantes fez surgir dispositivos cada vez menores, mais potentes e mais

baratos. Alguns deles possuem o poder de processamento equiparado a computadores de hoje em dia. Isso faz com que cada vez mais as pessoas se sintam atraídas por esses aparelhos e muitas vezes a carreguem-no consigo durante o dia inteiro, visto que algumas funcionalidades dos dispositivos móveis podem auxiliar no dia-a-dia de todas as pessoas.

Estes dispositivos podem ser amplamente explorados para a inclusão de pessoas com necessidades especiais, principalmente deficientes visuais, já que estes dispositivos móveis possuem um grande poder de processamento, acesso a rede mundial de computadores e possui também meios de interação com o meio, tal como microfone, autofalante e câmera.

O presente trabalho apresenta uma solução para algumas das dificuldades que são encontradas por pessoas que possuem deficiência visual, em especial, pessoas com baixa visão. O uso da câmera do celular permite a apresentação dos dados configurando o nível de *zoom* via software, apresentando a informação para o usuário por meio do display do dispositivo. Outra funcionalidade é o processamento da imagem, utilizando uma técnica de OCR (Optical Character Recognition), para recuperar o texto da imagem, passando este por uma técnica de sintetização de voz para que, o usuário com visão limitada não precise ler o texto, e sim, escutá-lo com o auxílio do dispositivo móvel.

1.2. OBJETIVOS

A seguir são apresentados o objetivo geral e os objetivos específicos do trabalho.

1.2.1. Objetivos Gerais

Desenvolvimento de um leitor digital para plataforma Android para pessoas que possuam baixa visão.

1.2.2. Objetivos Específicos

Dentre os objetivos específicos, destacam-se:

- Desenvolver um estudo fundamental sobre leitores digitais e dispositivos móveis;
- Realizar a análise de um sistema de leitor digital para dispositivos móveis;
- Implementar um sistema de leitor digital para dispositivos móveis, com funcionalidades de zoom e sintetização;
- Realizar os testes em um *device* Android real.

1.3. JUSTIFICATIVA

Atualmente, com as tecnologias computacionais existentes, é possível ver alguns dispositivos que desempenhem a função de leitor digital, tais como o Kindle, o SonyPRS, e o próprio IPad. Entretanto, estes permitem a leitura de documentos no formato digital, como e-books e arquivos pré-definidos para a leitura. Alguns possuem recurso de acessibilidade para a leitura dos documentos, porém, este não é o foco da plataforma.

Outro item que deve ser considerado é o valor a ser desembolsado para que o usuário possua tal tecnologia. O principal leitor digital fabricado pela Amazon, chamado Kindle, custa no mercado nacional em torno de R\$300,00 reais, o que é um valor considerável visando as limitações do dispositivo, que auxilia apenas na leitura de livros na forma digital.

Outra tecnologia existente que pode auxiliar na leitura, esta voltada para o público com baixa visão é a lupa eletrônica. Atualmente existem dois tipos deste dispositivo: as “câmera-mouse”, que são portáteis; e as denominadas “bandeja móvel”, que se movimentam através de um suporte. O aumento de tamanho de alguns desses dispositivos podem chegar a até a dezenas de vezes o tamanho da imagem real, mas quando manipuladas pela mão do usuário pode apresentar instabilidade devido a firmeza que é necessária.

Comparando com outros dispositivos móveis, o preço da lupa eletrônica é muito elevado, podendo chegar mais de R\$2.000,00 aproximadamente sete vezes o valor que seria gasto com um leitor digital comum ou com smartphones e tablets.

Sabe-se que atualmente o mercado de celulares e dispositivos móveis está aumentando muito, algumas pesquisas apontam que no Brasil atualmente tem-se 1,3 celular por habitante, sendo que este número desconsidera as modalidades de tablets e leitores digitais (VEJA, 2012).

Assim, pode-se considerar que o brasileiro tem fácil acesso a tecnologia dos dispositivos móveis. Nestes dispositivos podem ser instalados com facilidade softwares para leitura de e-books e materiais digitais, porém, o grande diferencial está na câmera digital presente nestes. Hoje em dia é muito difícil que um dispositivo móvel saia de fábrica sem uma câmera nativa, podemos utilizar essa funcionalidade dos aparelhos para fazer um leitor digital, utilizando a própria câmera do dispositivo.

O aplicativo proposto pretende unir a utilização do dispositivo móvel que utiliza o sistema operacional Android, com seus recursos de hardware e software, para auxiliar pessoas com baixa visão no desenvolvimento de tarefas diárias, como leitura de um jornal, de uma placa informativa ou mesmo de dados presentes na tela de um computador. Para isso pode-se utilizar a câmera nativa do dispositivo móvel para capturar uma imagem com texto, processa-la, dando a opção de apresentar no display do dispositivo em um tamanho maior personalizável, assim como permitir a sintetização deste.

1.4. ESTRUTURA DO TRABALHO

O trabalho está dividido em cinco capítulos. Destes, o primeiro apresenta a introdução, objetivos e justificativa para a realização deste trabalho.

O Capítulo 2 apresenta a fundamentação teórica sobre leitores digitais e *smartphones*.

NO Capítulo 3 serão apresentado os materiais (*softwares*) utilizados no desenvolvimento do presente trabalho, assim como a metodologia utilizada para chegar ao resultado final deste trabalho.

O Capítulo 4 apresenta o resultado obtido, sendo este um software para leitura e processamento de texto, sendo apresentada a análise, as telas e os principais códigos.

Finalizando, o Capítulo 5 apresenta as conclusões, as dificuldades encontradas e as sugestões de trabalhos futuros.

2. REFERENCIAL TEÓRICO

Este capítulo apresenta os conceitos de leitores digitais e de smartphones.

2.1. LEITORES DIGITAIS

Atualmente, a tecnologia da informação existente, em especial nos dispositivos móveis, permite que usuários utilizem seus *devices* como leitores digitais, trazendo assim mais praticidade na leitura. Um leitor digital nada mais é que um dispositivo que tem como função mostrar em sua tela um texto para leitura, seja ele diretamente de um livro digital (*e-book*) ou de outras mídias digitais.

A grande vantagem dessa tecnologia é permitir que o usuário carregue consigo inúmeras obras e livros, permitindo a leitura em qualquer lugar.

A Figura 2 apresenta uma das principais características dos leitores digitais, que são a tela grande e a alta resolução, o qual permite a leitura de textos em lugares com pouca luz ou excessivamente iluminados (por exemplo, ao ar livre, sobre a luz do sol).

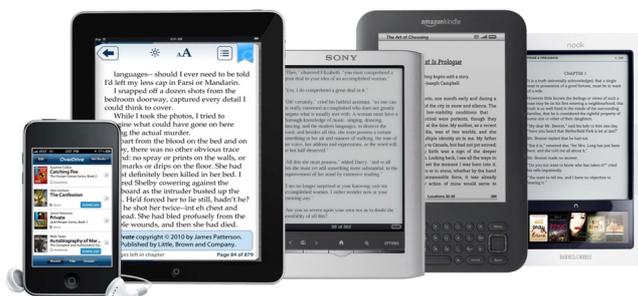


Figura 2 - Leitores Digitais
Fonte: TI ESPECIALISTAS, 2014

O principal leitor digital existente hoje é o Kindle, da Amazon, que está no mercado há alguns anos, possuindo funcionalidades como rotação, alternando entre o modo retrato e paisagem, dependendo da posição em que é segurado. Seu preço no mercado nacional é R\$ 300,00.

Outro dispositivo que também se destaca é o Sony PRS, que apesar de possuir funcionalidades semelhantes ao Kindle, possui um valor menor, variando em torno de R\$ 250,00.

A grande vantagem do uso destes dispositivos na leitura de textos é a eliminação completa de papeis, em contrapartida, um dos principais pontos fracos da plataforma é a autonomia das baterias, que duram em torno de 30 horas. Outra desvantagem é o elevado custo destes dispositivos no Brasil, uma vez que possui uma função muito específica, se comparado aos *smartphones* Android, onde alguns possuem o mesmo valor e possui muitas outras funcionalidades, acredita-se que este seja o principal motivo para sua não popularização aqui no Brasil.

2.2. LEITORES DIGITAIS PARA DISPOSITIVOS MÓVEIS

É muito comum nos dias de hoje a utilização de dispositivos móveis para trabalhar ou estudar. Estes dispositivos permitem aos seus usuários, nos espaços de tempo livre, a realização de muitas tarefas que até então só era possível realizar na frente de um computador.

Vários desses dispositivos permitem a instalação aplicativos para visualização e leituras de documento digital nos mais diversos formatos, em especial, em PDF (*Portable Document Format*).

Os aplicativos que apresentam as características de leitores digitais aparecem em todos os principais sistemas operacionais para *smartphones* (Android, Windows ou IOS). Suas funcionalidades são as mais diversas possíveis, permitindo desde armazenar um documento para ler mais tarde, até o gerenciamento do documento eletrônico, como uso de marcadores, exportação para outros formatos, etc.

Para o sistema operacional IOS existe o aplicativo nativo do sistema chamado iBook, este pode ser obtido gratuitamente e serve apenas para visualização dos documentos pré-definidos, como livros e documentos em PDF.

Para o sistema Windows pode-se citar o aplicativo BookViser Reader, também utilizado apenas para visualização de livros e documentos.

Para a plataforma Android a variação é maior. O principal aplicativo foi desenvolvido pelo próprio Google e se chama Google Play Livros, como pode ser observado na Figura 3, porém é apenas para leitura de livros comprados diretamente da loja da Google.

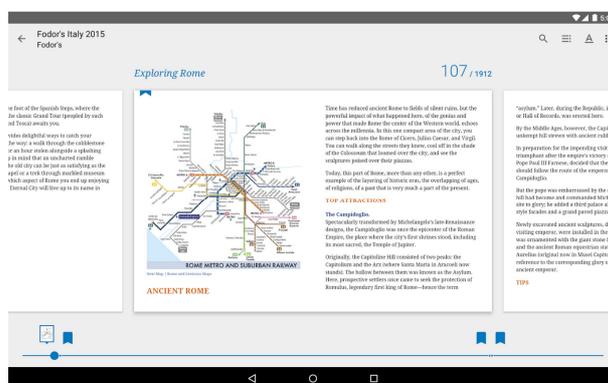


Figura 3 - Imagem do Aplicativo Google Play Livros

Também é possível encontrar aplicativos para dispositivos móveis que possuem a funcionalidade de transformar a imagem em texto, ou seja, que utiliza algoritmos de OCR (*Optical Character Recognition*). Entre eles, destacam-se o OCRNotes, Google Goggles e o MobileOCR, Estes transformam imagens tiradas com a câmera do dispositivos em textos. Os dois primeiros são gratuitos, já o terceiro tem o custo de R\$ 4,99.

Até a conclusão do presente trabalho, não foi encontrado aplicativos que faça a conversão de imagens tiradas com a câmera do dispositivo móvel em áudio.

2.3. LUPAS ELETRÔNICAS

As lupas eletrônicas são dispositivos que auxiliam as pessoas que possuem baixa visão. Para isso é necessário utilizar recursos das tecnologias atuais que possibilite a leitura e visualização de documentos ou sinalização que sem esses dispositivos não seria possível.

O funcionamento desse dispositivo irá depender muito do modelo que o usuário possui, pois existem os modelos portáteis e os modelos para bancada ou mesa. O primeiro modelo possui um design parecido com um *smartphone* e também é portátil, podendo ser carregado com o usuário durante o tempo todo. O funcionamento é através de uma câmera que transmite as imagens aumentadas em tempo real para uma tela acoplada no dispositivo, é o caso do aparelho Pebble HD (Figura 4).

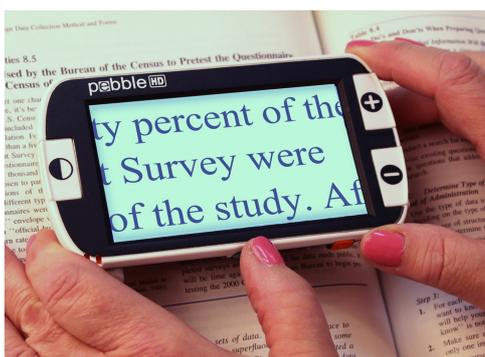


Figura 4 - Lupa Eletrônica Pebble HD

Fonte: (Pebble, 2014)

O segundo modelo para bancada ou mesa tem um funcionamento um pouco diferenciado. Alguns deles são menores, entretanto, precisam ser conectados a um aparelho de TV, para que as imagens sejam mostradas aumentadas, isso se dá por esses dispositivos não possuírem uma tela acoplada.

Para uma pessoa com baixa visão, um aparelho desses em mãos ou em casa pode facilitar sua inclusão aos meios de comunicação impressos. A principal desvantagem é o seu preço, que fica em torno de R\$1.000,00 podendo chegar em alguns casos a R\$2.500,00 para aparelhos mais sofisticados.

2.4. SMARTPHONES ANDROID

O sistema operacional Android é um conjunto de sistema operacional com plataforma de desenvolvimento para dispositivos móveis. Sua grande vantagem é o código aberto que proporciona liberdade de customização para os fabricantes dos aparelhos, assim como para os desenvolvedores de aplicativos. Este é o motivo do grande número de dispositivos que suportam esta tecnologia.

A trajetória do Android começou em 2005, quando o Google adquiriu o controle de uma pequena empresa de Palo Alto, na Califórnia, ainda em fase embrionária, que se chamava Android Inc. Na época, o pouco que se sabia sobre ela era que produzia softwares para celulares. Em meio a muitas especulações do mercado, logo começava o desenvolvimento do novo Android, baseado no Linux, sistema operacional gratuito para computadores (UOL, 2014).

Em 2008, após 3 anos de desenvolvimento, a empresa Google lançou o seu sistema operacional no mercado com o dispositivo HTC Dream, lançado em 22 de outubro do mesmo ano (PERBONI, 2014). Após, a plataforma Android se popularizou, chegando atualmente ao sistema operacional mais utilizado pelos usuários, como pode ser visto na Tabela 1.

Tabela 1 - Visão do mercado de sistemas operacionais de smartphones

Sistemas Operacionais	2013 Domínio de mercado (%)	2012 Domínio de mercado (%)
Android	78,6	69,0
iOS	15,2	18,7
Windows Phone	3,3	2,4
BlackBerry	1,9	4,5
Outros	1,0	5,4
Total	100	100

Fonte: Dias, 2014.

O sistema Android possui diversas vantagens, entre elas possuir código aberto, o que permite que os mais diferentes tipos de dispositivos, em especial, alguns muito baratos, sejam popularizados. A forte integração entre o ambiente de desenvolvimento e o sistema operacional (ambos Android) também é considerada uma grande vantagem.

3. MATERIAIS E MÉTODOS

Este capítulo apresenta os materiais (*Softwares*) utilizados para desenvolvimento do aplicativo foco deste trabalho, assim como os métodos utilizados para chegar ao resultado final.

3.1. MATERIAIS

Os materiais e ferramentas utilizados para desenvolver o aplicativo, resultado do presente trabalho, são apresentados no Quadro 1:

Quadro 1 - Ferramentas utilizadas no desenvolvimento do aplicativo

Nome	URL	Valor	Utilização
Astah	http://astah.net/download	Sem Custo	Software para modelagem dos diagramas
Eclipse IDE	www.eclipse.org/	Sem Custo	Utilizado como ambiente de programação
Android SDK	https://developer.android.com/sdk	Sem Custo	Pacote necessário para desenvolver aplicativos para plataforma Android
Plugin ADT	http://developer.android.com/tools/sdk/eclipse-adt.html	Sem Custo	Plugin utilizado para desenvolvimento Android no Eclipse IDE
Tesseract OCR	https://code.google.com/p/tesseract-ocr/	Sem Custo	API que faz a conversão da imagem para texto

As características e definições de cada ferramenta utilizada para este trabalho são apresentados na sequência:

a) **Astah:** Antigamente denominado JUDE, é uma ferramenta para criar os diagramas necessários para a modelagem do sistema, é desenvolvido em

tecnologia java e pode ser utilizado por qualquer computador que possui uma maquina virtual Java instalado.

b) **A IDE Eclipse:** O Eclipse foi criado pela International Business Machines (IBM) e é um ambiente de desenvolvimento para aplicações Java, porém suporta outras linguagens como C/C++, PHP, Python e Android. Atualmente é a mais utilizada e recomendada para desenvolvimento Android (OLIVEIRA, 2013).

c) **Android SDK:** É uma ferramenta que possui diversas bibliotecas e ferramentas para criar, testar e debugar os aplicativos criados para a plataforma Android. Esta ferramenta não possui um ambiente de desenvolvimento (IDE), sendo necessário integrá-la com uma das IDEs no mercado, como por exemplo o Eclipse.

d) **Plugin ADT:** É um plugin que amplia os recursos do IDE Eclipse para permitir que o usuário possa desenvolver aplicativos em Android. Este plugin fornece um ambiente visual de desenvolvimento, assim como ferramentas que automatizam a execução e os testes das aplicações.

e) **Tesseract OCR:** API de código aberto para transformar uma variedade de imagens para o formato de texto em mais de 60 linguagens diferentes. Desde 2006, esta ferramenta é mantida pela Google e é considerada uma das mais assertivas quando o assunto é OCR.

3.2. MÉTODOS

O desenvolvimento do aplicativo de leitor digital para pessoas com visão limitada foi dividido em algumas etapas, seguindo o modelo sequencial linear de Pressman (2006). Essas etapas são:

a) **Requisitos:** A definição dos requisitos foi realizada verificando a fácil usabilidade do aplicativo para usuário com visão limitada.

b) **Análise:** Foi analisado através da criação de diagramas de caso de uso e de atividade as funcionalidades do aplicativo. Tratando-se de um aplicativo com baixa interatividade com o usuário, dado a limitação destes, com apenas estes

dois diagramas foi possível registrar as funcionalidades do aplicativo. Estes diagramas foram desenvolvidos na ferramenta *Astah Community*.

c) **Desenvolvimento:** O Desenvolvimento foi realizado com base nos materiais apresentados na seção 3.1. Foi optado por ambientes de desenvolvimentos gratuitos.

d) **Teste:** Os testes foram realizados apenas para o funcionamento correto do sistema, foi utilizado um tablet Samsung Note 10.1, já que os emuladores não permitem a simulação da câmera do celular.

4. RESULTADOS

Este capítulo apresenta uma visão geral da análise realizada para o desenvolvimento do aplicativo, assim como alguns recursos específicos do sistema operacional Android, em especial, as funções da câmera e sintetização das palavras.

Após, será apresentado o aplicativo desenvolvido, com foco nas telas em que o usuário irá interagir.

4.1. MODELAGEM DO SOFTWARE

Esta seção apresenta a análise do aplicativo, abordando a análise de requisitos, os diagramas utilizados, assim com as principais funcionalidades. Dentre esses diagramas encontra-se o diagrama de caso de uso, que mostra as funcionalidades que o usuário terá com o aplicativo, e o diagrama de atividade, que mostra o fluxo do processo dentro do aplicativo, em especial, para apresentar a navegação entre telas.

A primeira etapa é realizar o levantamento dos requisitos funcionais, estes são apresentados nos quadros 2 a 5:

Quadro 2 - Requisito Funcional - Aplicar Zoom

F1 Zoom em Imagens	Oculto ()
Descrição: O aplicativo deverá mostrar em uma tela a imagem capturada pela câmera nativa do celular em tempo real e possibilitar que o usuário decida qual a quantidade de zoom que será aplicada na imagem.	
Não há Requisitos Não-Funcionais	

Quadro 3 - Requisito Funcional - Transformar Imagem em Texto

F2 Transformar Imagem em texto	Oculto (X)
Descrição: O aplicativo deverá converter o texto contido na imagem tirada através da camera nativa do dispositivo móvel para uma String.	
Não há Requisitos Não-Funcionais	

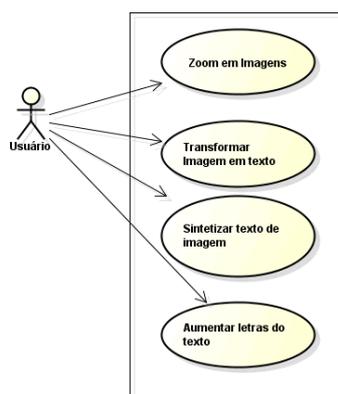
Quadro 4 - Requisito Funcional - Sintetizar texto

F3 Sintetizar texto	Oculto ()
Descrição: Possibilitar ao usuário sintetizar o texto contido na string que foi convertido de uma imagem.	
Não há Requisitos Não-Funcionais	

Quadro 5 - Requisito Funcional - Ampliar ou Reduzir Texto

F4 Ampliar ou Reduzir Texto	Oculto ()
Descrição: Possibilitar ao usuário o aumento ou redução do tamanho da letra no texto convertido através de uma imagem.	
Não há Requisitos Não-Funcionais	

A partir dos requisitos levantados para o sistema, foi possível criar o diagrama de caso de uso, conforme apresentado na Figura 5. Este diagrama mostra quais são as funcionalidades que o usuário com visão limitada terá a sua disposição.

**Figura 5 - Diagrama de Caso de Uso**

Basicamente, o usuário poderá utilizar o aplicativo como uma lupa eletrônica aplicando o zoom em imagens com o auxílio da câmera do dispositivo em tempo real, tirar fotos utilizando o mesmo recurso e transformar essas imagens em texto para síntese ou para um aumento de tamanho para melhor visualização.

Após a definição dos requisitos e o desenvolvimento do caso de uso, o passo seguinte foi o desenvolvimento do diagrama de atividade, que visa auxiliar o desenvolvimento de fluxo entre telas, assim como a usabilidade do aplicativo para o usuário com visão limitada.

O objetivo é deixar esta etapa o mais simples possível, já que a interação deste tipo de usuário com os aplicativos são muito limitadas.

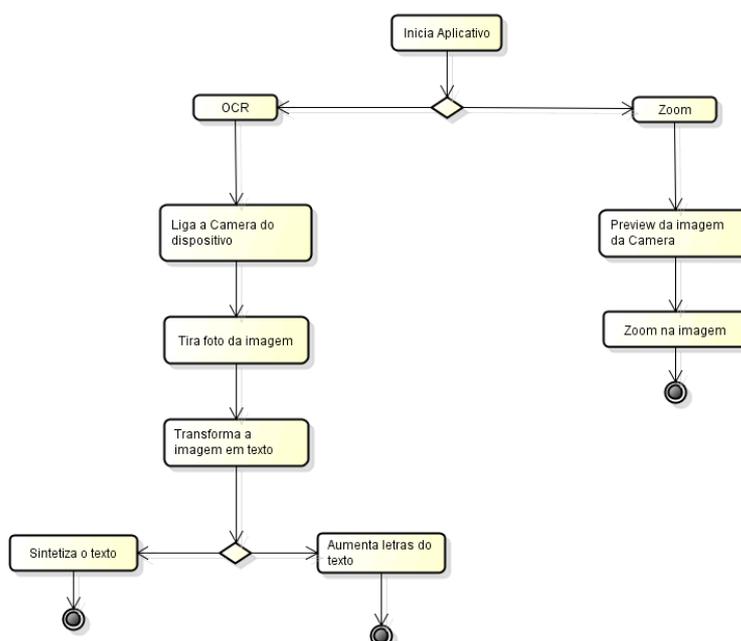


Figura 6 - Diagrama de Atividade

Após iniciar o aplicativo, o usuário tem duas opções: Zoom e OCR, onde a primeira irá abrir a interface para o usuário aplicar a função lupa, ou seja, dar zoom na imagem em tempo real. A segunda função irá abrir a câmera do dispositivo para que o usuário tire uma foto do texto desejado e a transforme em texto, assim ele terá duas possibilidades ou aumentar o tamanho da letra para leitura ou sintetizar o texto.

4.2. DESENVOLVIMENTO DO SISTEMA MÓVEL

Com a análise concluída, o passo seguinte foi a estruturação do projeto, identificando o número de classes. Com o pequeno número de telas, a quantidade de classes também ficou limitada.

Para o desenvolvimento, foi utilizado dois projetos: ProjetoTCC, o qual possui os códigos fontes do aplicativo que interage com o usuário com baixa visão, e o tess-two, biblioteca de código fonte com as funcionalidades para ler os dados da

câmera e transformar em OCR. A estrutura de ambos os projetos são apresentados na Figura 7.

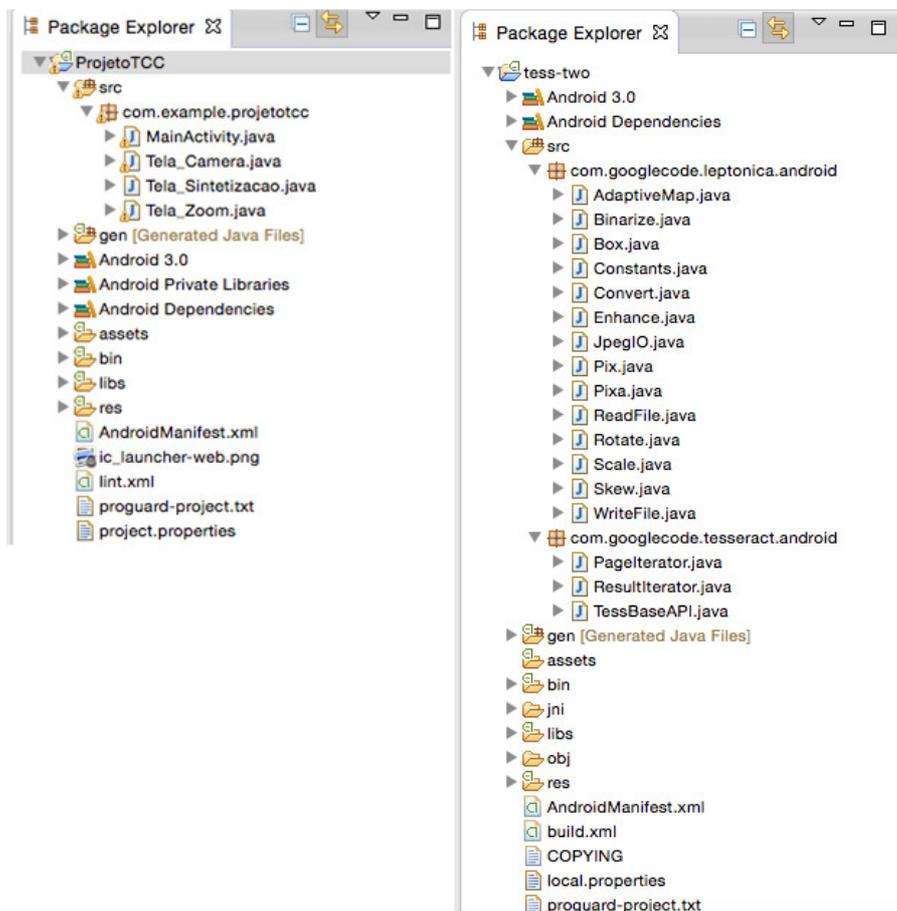


Figura 7 - Estrutura Código Fonte

Para desenvolvimento do aplicativo, foi utilizada a versão 4.0 do Android, sendo esta chamada de “Ice Cream Sandwich”. A escolha desta versão foi feita por o aplicativo possuir algumas funcionalidades que funcionam apenas nessa versão do Android, e também esta presente na maioria dos *devices* que utilizam o sistema Android hoje, aproximadamente 80% fazem parte dessa versão ou superior (Figura 8), ou seja não teriam problema com o funcionamento do aplicativo.

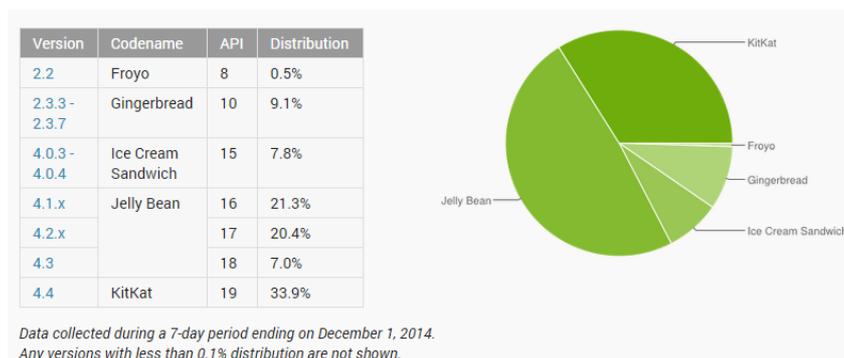


Figura 8 - Distribuição Plataforma Android

Fonte: Android Developer, 2014

Para o perfeito funcionamento do sistema móvel para pessoas com visão limitada, é necessário que o usuário já possua o *smartphone* ou *tablet* adaptado com as funcionalidades de acessibilidade. O dispositivo utilizado para testes foi o Samsung Galaxy Note 10.1, este é apresentado na Figura 09. Este dispositivo permite a função de acessibilidade, onde todos os menus e informações da tela são sintetizados, permitindo que deficientes visuais (e usuários com baixa visão) possam usar o *device*, assim como seus aplicativos.



Figura 9 - Dispositivo Samsung Galaxy Note 10.1

Fonte: Samsung, 2014

Para criar as telas de interação com o usuário foi utilizado o próprio editor visual de telas disponível no plugin ADT, o qual permite utilizar a tecnologia de arrastar e soltar para criar layout das telas. Fundamentalmente, estas telas são criadas via código xml, o que este editor faz é transformar um layout visual em código xml, como apresentado na Figura 10.

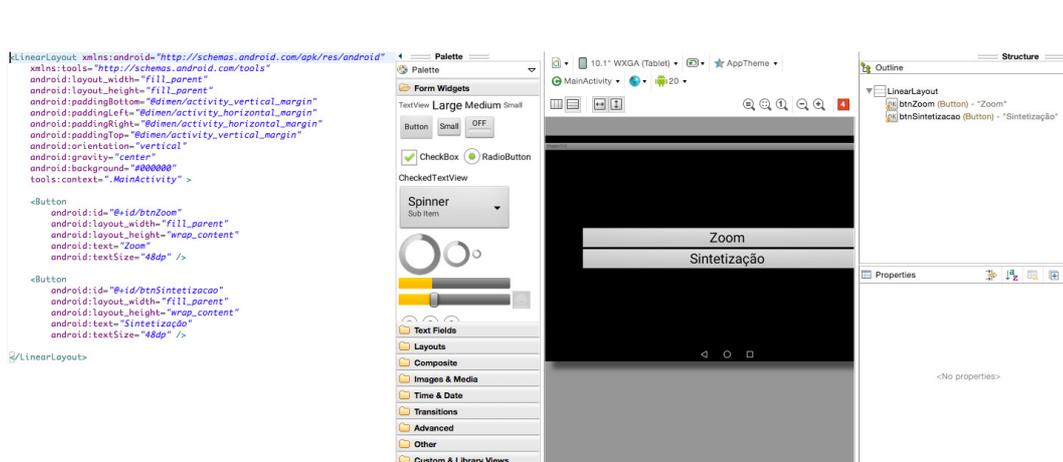


Figura 10 - Plugin Layout ADT

4.2.1 Aplicativo Desenvolvido

Ao iniciar o aplicativo, a tela principal é apresentada, nela o usuário poderá escolher entre as duas opções do menu principal: “Zoom” que irá mostrar a funcionalidade de lupa, e “Sintetização”, que irá solicitar para o usuário tirar uma foto do texto que deverá ser convertido para áudio.. A Figura 11 apresenta este menu Principal.



Figura 11 - Tela Inicial do Aplicativo

Ao acessar a funcionalidade do botão “Zoom”, é apresentado um *preview* da câmera e um botão chamado Tela Inicial, que retorna para o menu principal (Figura 13). Nessa tela o usuário tem como opção utilizar o zoom da câmera para aproximação do texto que deseja ler, para isso ele deve clicar a tela do dispositivo com dois dedos simultaneamente.

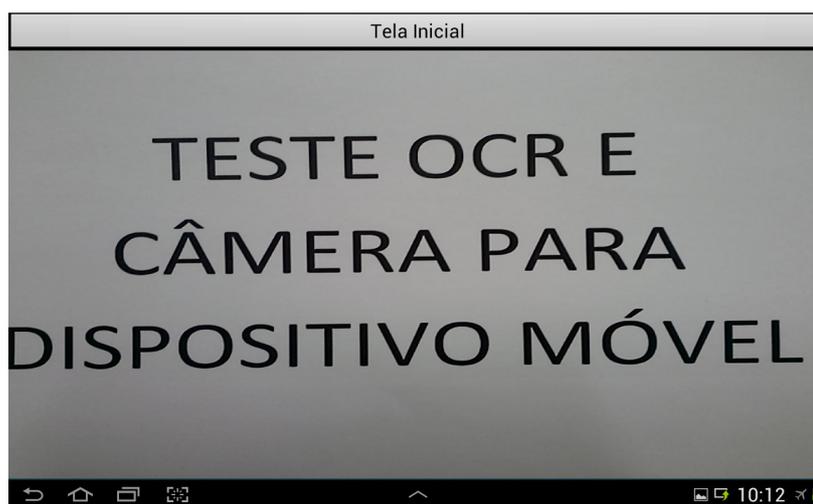


Figura 12 - Opção de Lupa do Aplicativo

A opção de zoom é limitada de acordo com o dispositivo móvel, alguns deles possuem uma câmera com mais megapixels que outras podendo ter uma aproximação maior sem perder qualidade. Para voltar ao estado normal (visualização padrão), o usuário deverá ir até o limite alcançado de zoom para a câmera, no próximo clique duplo na tela ela voltará ao normal.

Na tela inicial, escolhendo a opção “Sintetização”, o dispositivo abrirá a câmera do dispositivo diretamente para que seja capturada uma imagem do texto, como pode ser visto na Figura 13.

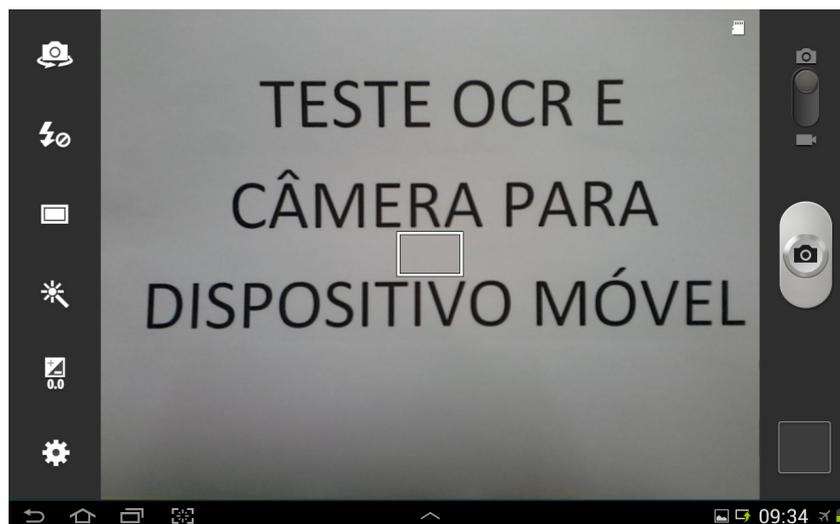


Figura 13 - Câmera do Dispositivo Móvel

Após, o usuário com visão limitada pode personalizar a características da câmera utilizando as opções do lado esquerdo da tela, que são as opções nativas da própria câmera do celular, ou pressionar o botão de câmera, do lado direito, para a captura da imagem pelo dispositivo móvel, será apresentada uma tela com o preview da foto e dois botões como pode ser visto na Figura 14, permitindo cancelar a foto ou confirma-la.

Escolhendo a opção Confirmar, o aplicativo processará o OCR e sintetizará este texto.



Figura 14 - Captura de Imagem para OCR

Após o usuário salvar a imagem no dispositivo e a API tesseract fazer o reconhecimento de caracteres, o texto contido será mostrado em uma tela nova, onde será possível o usuário realizar três funções, como pode ser visto na Figura 15. A primeira delas é a opção de sintetizar o texto de acordo com o idioma que o usuário possui no dispositivo móvel, dessa forma o usuário pode escutar o texto contido na imagem. Outra opção é aumentar a letra para facilitar a leitura clicando no botão “A+” e também poderá diminuir a letras clicando no botão “A-“, além de o botão voltar que retorna para a tela anterior.



Figura 15 - Imagem Transformada em texto

Para o reconhecimento dos caracteres e a sintetização do texto contido nela, não é necessário o usuário ter conexão com a rede da operadora ou acesso a rede mundial de computadores o que traz uma enorme vantagem de mobilidade para utilização dessa funcionalidade.

A seguir serão apresentados os códigos utilizados para alcançar as funcionalidades pretendidas para o trabalho.

4.2.3 Codificação de funcionalidades

Para utilização da funcionalidade de Lupa, foi necessário utilizar a câmera em tempo real dentro de uma área onde o usuário pudesse voltar ao aplicativo normalmente. Para isso foi necessário utilizar os objetos das classes *SurfaceView* e *SurfaceHolder*, para criar um quadro dentro da tela, o qual permite mostrar o *preview* com a imagem que esta sendo capturada na câmera.

Ao iniciar a utilização dessas classes, juntamente com a classe do tipo *Camera*, percebeu-se um problema na orientação no momento em que o dispositivo mudava sua rotação (de retrato para paisagem), para corrigir esse problema foi adicionado um código diretamente na classe *SurfaceHolder*, que é responsável pelas ações geradas pela classe *SurfaceView*, que é encarregada de receber a instancia direta da interface visual. Na Listagem 1 é mostrado o código para correção do problema de orientação.

Listagem 1 – Correção de Imagem

```

1. public void surfaceChanged(SurfaceHolder holder, int format, int w, int h) {
2.     Log.e(TAG, "surfaceChanging");
3.
4.     if(mCamera != null){
5.
6.         // XXX stopPreview() will crash if preview is not running
7.         if (mPreviewRunning) {
8.             mCamera.stopPreview();
9.         }
10.
11.         Log.d(TAG, "");
12.
13.         Camera.Parameters p = mCamera.getParameters();
14.         p.setPreviewSize(w, h);
15.
16.         if (getResources().getConfiguration().orientation ==
Configuration.ORIENTATION_PORTRAIT) {
17.             p.set("orientation", "portrait");
18.             // parametros.set("rotation", 90);
19.         } else if (getResources().getConfiguration().orientation ==
Configuration.ORIENTATION_LANDSCAPE) {
20.             p.set("orientation", "landscape");
21.             // parametros.set("rotation", 90);
22.         }
23.
24.         int rotation = getWindowManager().getDefaultDisplay().getRotation();
25.         int degrees = 0;
26.
27.         switch (rotation) {
28.             case Surface.ROTATION_0: degrees = 0; break;
29.             case Surface.ROTATION_90: degrees = 270; break;
30.             case Surface.ROTATION_180: degrees = 0; break;
31.             case Surface.ROTATION_270: degrees = 90; break;
32.         }
33.
34.         mCamera.setDisplayOrientation(degrees);

```

```

35.         p.setRotation(degrees);
36.
37.         try {
38.             mCamera.setPreviewDisplay(holder);
39.
40.         } catch (IOException e) {
41.             mCamera.release();
42.             mCamera = null;
43.             e.printStackTrace();
44.         }
45.
46.         mCamera.setParameters(p);
47.         try {
48.             mCamera.setPreviewDisplay(mSurfaceHolder);
49.         } catch (IOException e) {
50.             // TODO Auto-generated catch block
51.             // e.printStackTrace();
52.             Log.d("Erro:", e.toString());
53.         }
54.         mCamera.startPreview();
55.         mPreviewRunning = true;
56.         Log.e(TAG, "surfaceChanged");
57.     }else{
58.         Log.d(TAG, "mCamera = null!!");
59.     }
60. }

```

Após a correção da imagem por parte das classes *SurfaceViewer* e *SurfaceHolder*, pode-se utilizar a classe *Camera* para outras funcionalidades. Na funcionalidade da lupa não é necessário à função de capturar imagem, portanto será utilizado somente a função de zoom da câmera, que é tratado pela função `params.setZoom()`. Para tratar o método que será efetuado o zoom na câmera, foi utilizado a função `actionToString()` que irá verificar o posicionamento dos toques na tela e retornará uma *String* identificando se o zoom será aplicado ou não, como registrado na Listagem 2.

Listagem 2 – Funcionalidade Lupa

```

1.  @Override
2.  public boolean onTouchEvent( MotionEvent e ) {
3.      Camera.Parameters params = mCamera.getParameters();
4.
5.      int action = MotionEventCompat.getActionMasked( e );
6.
7.      if(mCamera.getParameters().isZoomSupported()){
8.          String strAction="";
9.          maxZoomLevel = params.getMaxZoom();
10.         System.out.println(currentZoomLevel);
11.
12.         if(e.getPointerCount()<2){
13.             AutoFocusCallback cb = null;
14.             mCamera.autoFocus(cb);
15.
16.         }
17.
18.         if(e.getPointerCount()>1){
19.             strAction = actionToString(action);
20.             if(strAction.equals("zoom")){
21.
22.                 if(currentZoomLevel == maxZoomLevel){
23.                     currentZoomLevel =0;
24.                 }else{

```

```

25.         currentZoomLevel=currentZoomLevel+5;
26.     }
27.
28.     params.setZoom(currentZoomLevel);
29.     mCamera.setParameters(params);
30.     }
31.     }
32.
33. }
34. return true;
35.
36. }
37.
38. public String actionToString( int action ) {
39.
40.     switch( action ) {
41.     case MotionEvent.ACTION_DOWN: return "down";
42.     case MotionEvent.ACTION_MOVE: return "move";
43.     case MotionEvent.ACTION_POINTER_DOWN: return "pointer down";
44.     case MotionEvent.ACTION_POINTER_UP: return "zoom";
45.     case MotionEvent.ACTION_UP: return "up";
46.     case MotionEvent.ACTION_OUTSIDE: return "outside";
47.     case MotionEvent.ACTION_CANCEL: return "cancel";
48.     }
49.
50.     return "";
51. }
52.

```

Para realizar a transformação da imagem para texto foi necessário utilizar as funções nativas da classe Camera. Porém para transformar a imagem em texto é necessário converter em um formato específico através da classe Bitmap, como apresentado na Listagem 3. Após essa mudança de formato a imagem está pronta para ser transformada.

Listagem 3 – Convertendo imagem

```

53.     BitmapFactory.Options options = new BitmapFactory.Options();
54.     options.inSampleSize = 4;
55.
56.     Bitmap bitmap = BitmapFactory.decodeFile(_path, options);
57.
58.     try {
59.         ExifInterface exif = new ExifInterface(_path);
60.         int exifOrientation = exif.getAttributeInt(
61.             ExifInterface.TAG_ORIENTATION,
62.             ExifInterface.ORIENTATION_NORMAL);
63.
64.         Log.v(TAG, "Orient: " + exifOrientation);
65.
66.         int rotate = 0;
67.
68.         switch (exifOrientation) {
69.         case ExifInterface.ORIENTATION_ROTATE_90:
70.             rotate = 90;
71.             break;
72.         case ExifInterface.ORIENTATION_ROTATE_180:
73.             rotate = 180;
74.             break;
75.         case ExifInterface.ORIENTATION_ROTATE_270:
76.             rotate = 270;
77.             break;
78.         }
79.
80.         Log.v(TAG, "Rotation: " + rotate);
81.

```

```

82.         if (rotate != 0) {
83.
84.             int w = bitmap.getWidth();
85.             int h = bitmap.getHeight();
86.
87.
88.             Matrix mtx = new Matrix();
89.             mtx.preRotate(rotate);
90.
91.
92.             bitmap = Bitmap.createBitmap(bitmap, 0, 0, w, h, mtx, false);
93.         }
94.
95.
96.
97.         bitmap = bitmap.copy(Bitmap.Config.ARGB_8888, true);
98.
99.     } catch (IOException e) {
100.         Log.e(TAG, "Couldn't correct orientation: " + e.toString());
101.     }

```

Para a transformação da imagem em texto deve-se utilizar a API tesseract. O código desta funcionalidade é apresentada na Listagem 4. Para conversação, deve-se passar por parâmetro a imagem (bitmap) e chamar o método getUTF8Text.

Listagem 4 – Utilizando a API tesseract

```

1.     TessBaseAPI baseApi = new TessBaseAPI();
2.     baseApi.setDebug(true);
3.     baseApi.init(DATA_PATH, lang);
4.     baseApi.setImage(bitmap);
5.
6.     String recognizedText = baseApi.getUTF8Text();
7.
8.     baseApi.end();
9.
10.    if ( lang.equalsIgnoreCase("eng") ) {
11.        recognizedText = recognizedText.replaceAll("[^a-zA-Z0-9]+", " ");
12.    }
13.
14.    recognizedText = recognizedText.trim();
15.

```

Com o texto formado, é possível sintetizar o texto através da função TextToSpeech(), esta nativa da plataforma Java, a partir da versão 1.6 conhecida como “Donut”, que é detalhada na Listagem 5.

Listagem 5 – Sintetizando o texto

```

1.     btIniciaSintetizacao.setOnClickListener(new View.OnClickListener() {
2.
3.         @Override
4.         public void onClick(View v) {
5.
6.             Intent checkTTSIntent = new Intent();
7.             checkTTSIntent.setAction(TextToSpeech.Engine.ACTION_CHECK_TTS_DATA);
8.             startActivityForResult(checkTTSIntent, MY_DATA_CHECK_CODE);
9.
10.            String texto = etTextoOCR.getText().toString();
11.            Sintetiza(texto);
12.
13.

```

```
14.         }
15.     });
16.
17.     public void onInit(int status) {
18.         // TODO Auto-generated method stub
19.         if (status == TextToSpeech.SUCCESS) {
20.             myTts.setLanguage(Locale.US);
21.
22.         }
23.         else if (status == TextToSpeech.ERROR) {
24.             Toast.makeText(this, "Desculpe! A Aplicação falhou...",
25.                 Toast.LENGTH_LONG).show();
26.         }
27.
28.
29.         private void Sintetiza(String texto){
30.             myTts.speak(texto, TextToSpeech.QUEUE_FLUSH, null);
31.         }
32.
33.         protected void onActivityResult(int requestCode, int resultCode, Intent data) {
34.             if (requestCode == MY_DATA_CHECK_CODE) {
35.                 if (resultCode == TextToSpeech.Engine.CHECK_VOICE_DATA_PASS) {
36.                     myTts = new TextToSpeech(this, this);
37.                 }
38.                 else {
39.
40.                     Intent installTTSIntent = new Intent();
41.                     installTTSIntent.setAction(TextToSpeech.Engine.ACTION_INSTALL_TTS_DATA);
42.                     startActivity(installTTSIntent);
43.                 }
44.             }
45.         }
```

5. CONCLUSÃO

O aplicativo desenvolvido tem a finalidade de ser um leitor digital para plataforma Android que pode ser utilizado por pessoas que possuem baixa visão, auxiliando no dia-a-dia destas para que se tornem menos dependentes em suas atividades diárias.

Durante o desenvolvimento do aplicativo foi realizado um estudo sobre os leitores digitais e os dispositivos móveis, este estudo mostra que apesar dos leitores digitais possuírem uma funcionalidade interessante para pessoas com problema de visão, ainda não é viável em nosso país pela pouca produtividade e alto custo se comparado com os dispositivos móveis como smartphones e tablets. Destaca-se entre estes dispositivos a plataforma Android por possuir código fonte aberto e permitir a manipulação de seus recursos, como câmera, sintetização de voz e a sua popularidade que vem aumentando a cada dia.

Com a possibilidade de manipular os recursos da plataforma Android, iniciou-se um estudo para criação de um leitor digital para pessoas com dificuldade de visão utilizando os dispositivos que utilizam essa tecnologia. Para isso foi pensado em algumas funcionalidades que iriam auxiliar essas pessoas tais como ampliar imagens em tempo real utilizando a câmera do celular fazendo a funcionalidade parecida com uma lupa eletrônica, e utilizar a câmera do dispositivo móvel para capturar uma imagem com texto e transforma-la em um texto para que fosse possível manipular e sintetizar.

As funcionalidades foram desenvolvidas e aplicadas utilizando basicamente os recursos nativos do próprio celular, com exceção da conversão da imagem para o texto, OCR, que foi utilizado a API Tesseract, esta escolhida pelo motivo de não ser necessário a utilização de rede de celular ou internet para que funcione corretamente, proporcionando uma maior mobilidade para o usuário.

Para realização de testes e verificação das funcionalidades do aplicativo, foi utilizado um tablet que utiliza a plataforma Android, assim sendo possível validar todos os objetivos que seria necessário alcançar com o trabalho.

5.1. TESTES

Os testes para as funcionalidades do aplicativo foram realizados em um dispositivo móvel real, o tablet Samsung Galaxy Note 10.1. Para isso foi utilizado a câmera do dispositivo para captar imagens em documentos impressos e em livros não levando em consideração a luminosidade do ambiente.

Foi identificado que apesar do alto percentual de reconhecimento nos documentos impressos e nos livros, o aplicativo não reconhece letras cortadas ou com letras em escritas emendadas. Também foi identificado que não é recomendável a utilização de imagens juntamente com o texto a ser reconhecido, visto que o aplicativo não faz distinção de figuras e texto.

5.2. PROBLEMAS ENCONTRADOS

Durante o desenvolvimento do aplicativo foi encontrado algumas dificuldades, entre elas pode-se citar a dificuldade em alocar as funcionalidades de câmera no aplicativo, primeiramente foi encontrado problemas para fazer o tratamento do zoom da função lupa, achando como alternativa a função de multitoque do dispositivo.

Também foi encontrada dificuldade para salvar a imagem no cartão de memória do dispositivo móvel o que acarretou no aumento do tempo para o desenvolvimento, conseqüentemente o tratamento da foto a API Tesseract também foi dificultado por ser necessário buscar a imagem do cartão de memória.

5.3. TRABALHOS FUTUROS

Para trabalho futuro pretende-se melhorar a usabilidade do aplicativo, para ele ser mais intuitivo e melhorar no desempenho para a transformação da imagem

para texto, para isso é necessário manter a API Tesseract atualizada, então terá de ter uma manutenção frequente.

No que diz respeito a função de lupa, ou seja, o aumento de zoom na imagem em tempo real também pode ser melhorado, para isso será necessário adaptar o *preview* da câmera para novas formas de aplicar o zoom na imagem, facilitando a usabilidade do usuário.

Também se pode pensar em melhorar a interface visual para o usuário, pois como este aplicativo foi feito apenas para o trabalho de conclusão de curso não foi analisada a possibilidade de ter uma interface mais elaborada, e sim focado apenas nas funcionalidades do aplicativo.

6. REFERÊNCIAS

Android – **Dashboards** | **Android Developers** – Disponível em: <<http://developer.android.com/about/dashboards/index.html>> <acesso em 02/12/2014>

CANALTECH - **Simon, o primeiro smartphone do mundo, surgia há 20 anos** - Disponível em: < <http://canaltech.com.br/noticia/smartphones/Simon-o-primeiro-smartphone-surgia-ha-20-anos/>> <Acesso em: 05/11/2014>

COSTA Gabriela R. V; MAIOR Izabel M. M. de L.; LIMA Niusarete M. - **ACESSIBILIDADE NO BRASIL: UMA VISÃO HISTÓRICA.** Disponível em: <http://www.prodiam.sp.gov.br/multimidia/midia/cd_atiid/conteudo/ATIID2005/MR1/01/AcessibilidadeNoBrasilHistorico.pdf> <Acesso em: 04/11/2014>

DIAS, Eduardo, Campos. **Dificuldade Zero.** Disponível em: <<http://dificuldadezero360.blogspot.com.br/2014/02/o-atual-mercado-dos-sistemas.html>> Acesso em 19 agosto 2014.

IBGE - **População Brasileira Envelhece Em Ritmo Acelerado** - Disponível em: <<http://saladeimprensa.ibge.gov.br/noticias?idnoticia=1272%3Eacesso&view=noticia>> <Acesso em: 05/11/2014>

Ministério da Saúde - **PROJETO OLHAR BRASIL** - Disponível em: <http://www.pmf.sc.gov.br/arquivos/arquivos/pdf/11_05_2011_15.20.17.f0faa995acac e3507158a755051265b1.pdf> <Acesso em: 05/11/2014>

NAUAR, Giowana - **Nordeste concentra maior percentual de população deficiente, mostra IBGE** - Disponível em: <<http://giowananauar.blogspot.com.br/2012/05/nordeste-concentra-maior-percentual-de.html>> <Acesso em: 05/11/2014>

OLIVEIRA – **Blog em Gestão de TI – Configurando o ambiente para desenvolvimento Android** – Disponível em: <<http://gestaoemti.com.br/index.php/configurando-o-ambiente-para-o-desenvolvimento-com-android/>> Acesso em: <22/10/2013>

PEBBLE – **Pebble HD 4.3 Inch Digital Handheld Magnifier** – Disponível em: <<https://www.enhancedvision.com/low-vision-product-line/pebble-hd-hand-held-portable-electronic-magnifier.html>> <Acesso em: 08/12/2014>

PERBONI, Marcos - **O que é Android – Sua história** - Disponível em: <<https://marcosvperboni.wordpress.com/em-breve-noticias-de-funcoes-do-android-para-leigos/>> <Acesso em: 05/11/2014>

PRESSMAN, Roger S. **Engenharia de Software. 6ª Edição**. São Paulo: McGraw-Hill do Brasil, 2006.

SAMSUNG – **Galaxy Note 10.1 | Samsung Brasil** - Disponível em: <<http://www.samsung.com/br/consumer/cellular-phone/tablet/galaxy-note/GT-N8000ZWAZTA>> <Acesso em: 05/11/2014>

SIMS - **O que é inclusão social** - Disponível em: <<http://www.sims.ap.gov.br/det.php?y=9>> <Acesso em: 04/11/2014>

TI ESPECIALISTAS – **eReader_Workshop_1_770x433** – Disponível em: <http://imagens.tiespecialistas.com.br/2013/09/eReader_Workshop_1-770x433.jpg> <Acesso em: 05/12/2014>

TUDOCELULAR - **Confira 5 vantagens e 5 desvantagens do sistema Android** - Disponível em: <<http://www.tudocelular.com/google/noticias/n36150/confira-5-vantagens-e-desvantagens-do-android.html>> <Acesso em: 05/11/2014>

UOL - **Sistema operacional do Google conquista espaço no mercado de smartphones** - Disponível em: <<http://android.uol.com.br/o-que-e-google-android.jhtm>> <Acesso em: 05/11/2014>

VEJA - **Brasil atinge marca de 1,3 celular por habitante.** - Disponível em: <<http://veja.abril.com.br/noticia/vida-digital/brasil-atinge-marca-de-1-3-celular-por-habitante>> <Acesso em: 05/11/2014>