

UNIVERSIDADE DO EXTREMO SUL CATARINENSE - UNESC

CURSO DE CIÊNCIA DA COMPUTAÇÃO

CLAITON DE MELO MARCILIO

**PROTÓTIPO DE APLICATIVO ANDROID PARA EXTRAÇÃO DE TEXTO EM
IMAGENS E CONVERSÃO EM VOZ, ORIENTADO AO APOIO A LEITURA PARA
DEFICIENTES VISUAIS**

CRICIÚMA

2013

CLAITON DE MELO MARCILIO

**PROTÓTIPO DE APLICATIVO ANDROID PARA EXTRAÇÃO DE TEXTO EM
IMAGENS E CONVERSÃO EM VOZ, ORIENTADO AO APOIO A LEITURA PARA
DEFICIENTES VISUAIS**

Trabalho de Conclusão de Curso, apresentado para obtenção de grau de Bacharel, no curso de Ciência da Computação da Universidade do Extremo Sul Catarinense, UNESC.

Orientador: Prof. Esp. Fabrício Giordani
Coorientador: Prof. M.Eng. Evânio Ramos
Nicoleit

CRICIÚMA

2013

CLAITON DE MELO MARCÍLIO

**PROTÓTIPO DE APLICATIVO ANDROID PARA EXTRAÇÃO DE TEXTO EM
IMAGENS E CONVERSÃO EM VOZ, ORIENTADO AO APOIO A LEITURA PARA
DEFICIENTES VISUAIS**

Trabalho de Conclusão de Curso aprovado pela Banca Examinadora para obtenção do Grau de Bacharel, no Curso de Ciência da Computação da Universidade do Extremo Sul Catarinense, UNESC, com Linha de Pesquisa em Acessibilidade e Desenvolvimento para Dispositivos Móveis.

Criciúma, 24 de junho de 2013.

BANCA EXAMINADORA



Prof. Fabricio Giordani - Especialista - (UNESC) - Orientador



Prof. Evânio Ramos Nicoleit - Mestre - (UNESC) – Co-orientador



Prof. Gustavo Bisognin - Mestre - (UNESC)



Prof. Gilberto Vieira da Silva - Especialista - (UNESC)

**Dedico este trabalho a minha mulher Bruna,
meu filho Arthur e a toda minha família, pela
inspiração e motivação.**

AGRADECIMENTOS

Agradeço primeiramente a Deus, por iluminar toda a minha trajetória até aqui, guiando meus passos e dando a sabedoria necessária para concluir mais esta etapa de minha vida.

Aos meus pais Marizete e José Carlos e a minha irmã Carla, um agradecimento especial, por nunca medirem esforços para que tudo isto fosse possível, apoiando em cada momento de dificuldade e sempre me guiando pelo caminho da ética, da moral e do respeito mútuo. Essa conquista deve-se em muito a todo estímulo pela busca da sabedoria e valorização do conhecimento por eles fomentado. Dedico a eles esse trabalho, como forma de agradecimento por todo amor e carinho recebido desde a criação, em toda a vida escolar e principalmente agora, no difícil período da graduação. A eles todo meu amor e gratidão.

Um agradecimento todo especial a minha mulher Bruna, a qual é a minha fonte de inspiração. Ela que, com todo seu amor, soube tornar muito mais suave e agradável cada momento de dificuldade vivido durante este trabalho. A conclusão do mesmo, só foi possível graças a todo seu apoio, carinho, compreensão e atenção, dados diária e incondicionalmente. Esta mulher que me acompanha a tanto e que é o melhor presente que Deus me deu, vem agora trazer minha maior alegria, nosso tão amado filho Arthur, o qual aguardamos ansiosamente por sua chegada. Obrigado por tudo meu amor!

A todos meus amigos do curso, que ajudaram durante todo o período em que estivemos na universidade, seja com momentos de descontração ou na execução das tarefas, mas participaram na composição de todo conhecimento adquirido nesse período.

A meu grande amigo Luiz Ricardo, que não mediu esforços para ajudar a todos, especialmente a mim, na elaboração deste trabalho. Um homem de caráter, honestidade, humildade e de um talento singular, que mereceu todas suas conquistas e as que estão por vir. Este que provou que o tempo não é um empecilho quando se tem vontade e dedicação. Parabéns e obrigado Gerente.

Ao Raul, este grande parceiro que esteve presente em todos momentos, companheiro de café e de noites de desenvolvimento de trabalho. Quero agradecer a todo apoio e incentivo na vida acadêmica e pessoal, e desejar muita sorte em seu futuro, o qual, tenho certeza, será brilhante.

Ao meu grande irmão Anderson, este que é um amigo de longa data e que sempre esteve disposto a auxiliar em tudo, seja na parceria nos trabalhos em dupla, ou para tirar dúvidas em matérias e trabalhos, enfim, uma pessoa sem igual. Quero agradecer por todos fins de semana desprendidos para que pudéssemos desempenhar as tarefas do curso e pela parceria de sempre. Devo muito do meu conhecimento a este grande homem, que possui um talento e um carisma único. Valeu Adirso!

Um agradecimento a todos os professores do curso, em especial aos professores Fabricio e Evânio, os quais puderam me guiar por todo esse período de desenvolvimento do trabalho de conclusão de curso.

A Associação dos Deficientes Visuais do Sul (Advsul), por ter me proporcionado uma verdadeira lição de vida, através do apelo social a qual eles se disponibilizam a fazer diariamente. A todos novos amigos que fiz na associação, quero agradecer pelo apoio e pela experiência que com certeza jamais esquecerei.

Por fim, a todos que de forma direta ou indireta influenciaram para que este trabalho e esta formação fosse possível. Muito obrigado a todos.

“Me dá um grande e profundo conforto que as coisas vistas são temporais e coisas não vistas são eternas.”

Helen Keller

RESUMO

A plataforma dos dispositivos móveis é uma das que mais tem evoluído e se popularizado atualmente. Aparelhos tais como celulares, *smartphones* e *tablets* são cada vez mais comuns e com maior capacidade de processamento. A plataforma Android é atualmente uma das líderes nesse segmento de tecnologia e disponibiliza uma série de recursos para que desenvolvedores criem cada vez mais novas funcionalidades para estes dispositivos. Um dos conceitos que tem se beneficiado com essa evolução é a acessibilidade, originando as chamadas Tecnologias Assistivas. Essas tecnologias buscam trazer melhor qualidade de vida e mais acesso a pessoas portadoras de deficiência. Baseado nesse conceito, este trabalho apresenta o desenvolvimento de um protótipo de aplicativo Android para extrair o texto presente em imagens capturadas a partir da câmera de um dispositivo móvel e convertê-lo em voz, com intuito de apoiar deficientes visuais no acesso a informação impressa. Para isso, foi realizada uma fundamentação teórica referente as técnicas de Reconhecimento Óptico de Caracteres (OCR), responsável por reconhecer caracteres presentes em imagens, bibliotecas de conversão de texto para voz, metodologias de acessibilidade em dispositivos moveis, dentre outros. Com os conhecimentos obtidos a partir da pesquisa, foi possível elaborar um protótipo de aplicativo, utilizando a ferramenta Tesseract como mecanismo OCR, e a biblioteca *TextToSpeech*, nativa do Android, que faz a conversão de texto em voz. Em testes realizados com o protótipo, os resultados apresentados demonstraram que o mesmo é adequado para usuários portadores de baixa visão e usuários com visão normal, permitindo que estes convertam o texto capturado pelo dispositivo para saída de voz, fornecendo assim acesso fácil a informações impressas.

Palavras-chave: Acessibilidade. Dispositivos móveis. Deficiência Visual. Plataforma Android. Reconhecimento Óptico de Caracteres.

ABSTRACT

The platform of mobile devices is one of the most evolved and has become popular nowadays. Devices such as mobile phones, smartphones and tablets are becoming more common and more powerful processing capability. The Android platform is currently one of the leaders in this technology segment and offers a number of resources to developers to create more and more new features for these devices. One of the concepts that have benefited from this development is accessibility, originating calls Assistive Technologies. These technologies seek to bring better quality of life and more access to people with disabilities. Based on this concept, this paper presents the development of a prototype Android application to extract the text present in the captured images from the camera of a mobile device and convert it to voice, in order to assist the visually impaired access to printed information. For this, we performed a theoretical referent techniques Optical Character Recognition (OCR), responsible for recognizing characters present in images, text conversion libraries for voice, methodologies accessibility in mobile devices, among others. With the knowledge obtained from the research, it was possible to develop a prototype application using the tool as Tesseract OCR engine, and the library TextToSpeech, Native Android, which makes converting text into voice. In tests conducted with the prototype, the results demonstrate that it is suitable for users with low vision and users with normal vision, allowing them to convert the text captured by the voice output device, thereby providing easy access to printed information.

Keywords: accessibility. Mobile devices. Visual Impairment. Android platform. Optical Character Recognition.

LISTA DE ILUSTRAÇÕES

Figura 1 – Tela principal do programa MAGIc.....	21
Figura 2 – Ferramenta Lupa do Windows em modo ancorado.....	22
Figura 3 – Tela principal do programa ZoomText.....	23
Figura 4 – Tela principal do programa JAWS.....	24
Figura 5 – Tela do Painel de Controle do Virtual Vision	24
Figura 6 – Tela de Preferências do Orca	25
Figura 7 - Método de entrada de texto NavTap.....	27
Figura 8 – Arquitetura do Android	31
Figura 9 – Compilação nas máquinas virtuais Java e Dalvik	35
Figura 10 – Ciclo de Vida de uma Activity	37
Figura 11 – Construção da Biblioteca com NDK	50
Figura 12 – Trecho do código fonte para utilização da Tesseract.....	52
Figura 13 – Método responsável pela troca de cor conforme alinhamento do dispositivo.....	56
Figura 14 – Câmera desalinhada	57
Figura 15 – Câmera alinhada	57
Figura 16 – Exemplo de imagem capturada nos cenários A, B e C, respectivamente	59

LISTA DE TABELAS

Tabela 1 – Número de smartphones embarcados no mundo por trimestre (milhões)	26
Tabela 2 – Os Quatro Possíveis Componentes de um Aplicativo.....	36
Tabela 3 – Resultados dos testes conforme cada cenário	60

LISTA DE ABREVIATURAS E SIGLAS

API	<i>Application Program Interfaces</i>
APK	<i>Application Packages</i>
HP	Hewlett-Packard
IBGE	Instituto Brasileiro de Geografia e Estatística
MPO	<i>Mobile Phone Organizer</i>
NDT	<i>Native Development Toolkit</i>
OCR	Reconhecimento Óptico de Caracteres
OHA	<i>Open Handset Alliance</i>
SDK	<i>Software Development Kit</i>
SOAP	<i>Simple Object Access Protocol</i>
TA	Tecnologia Assitiva
TTS	<i>TextToSpeech</i>
UDDI	<i>Universal Description Discovery and Integration</i>
W3C	<i>World Wide Web Consortium</i>
WSDL	<i>Web Services Description Language</i>
XML	<i>Extensible Markup Language</i>

SUMÁRIO

1 INTRODUÇÃO	14
1.1 OBJETIVO GERAL	15
1.2 OBJETIVOS ESPECÍFICOS	15
1.3 JUSTIFICATIVA	15
1.4 ESTRUTURA DO TRABALHO.....	16
2 DEFICIÊNCIA NO CONTEXTO SOCIAL.....	18
2.1 A SOCIEDADE E AS PESSOAS COM DEFICIÊNCIA.....	18
2.2 ACESSIBILIDADE	19
2.2.1 Tecnologia Assistiva Para Deficientes Visuais	20
2.2.2 Acessibilidade em Dispositivos Móveis.....	25
3 TECNOLOGIA	30
3.1 ANDROID.....	30
3.1.1 Arquitetura Android	31
3.1.1.1 Aplicações	32
3.1.1.2 Framework Application	32
3.1.1.3 Bibliotecas	34
3.1.1.4 Android Runtime.....	35
3.1.1.5 Kernel Linux	35
3.1.2 Componentes das Aplicações	36
3.1.2.1 <i>Activities</i>	36
3.1.2.2 <i>Services</i>	38
3.1.2.3 <i>Content Providers</i>	38
3.1.2.4 <i>Broadcast Receivers</i>	39
3.2 RECONHECIMENTO ÓPTICO DE CARACTERES.....	39
3.2.1 Etapas do Reconhecimento Óptico De Caracteres	40
3.2.2 Pré-processamento de Imagens	41
3.2.3 Ferramentas de Reconhecimento Óptico de Caracteres	42
4 TRABALHOS CORRELATOS.....	44
4.1 OCRDROID.....	44
4.2 PRÉ-PROCESSAMENTO DE IMAGENS PARA O RECONHECIMENTO ÓPTICO DE CARACTERES.....	45

4.3 ESTUDO DE INTERFACES ACESSÍVEIS PARA USUÁRIOS COM DEFICIÊNCIA VISUAL EM DISPOSITIVOS MÓVEIS TOUCH SCREEN.....	45
4.4 SISTEMA INTEGRADO PARA BIBLIOTECAS E ARQUIVOS DIGITAIS.....	47
5 O APLICATIVO ANDROID PARA EXTRAÇÃO DE TEXTO EM IMAGENS E CONVERSÃO EM VOZ	48
5.1 METODOLOGIA.....	48
5.1.1 A Ferramenta OCR Tesseract.....	48
5.1.2 Reconhecimento das Bibliotecas Android.....	50
5.1.2.1 Captura da Imagem.....	50
5.1.2.2 Utilizando o Tesseract.....	51
5.1.2.3 Conversão Texto-Voz.....	52
5.1.3 Desenvolvimento do Protótipo	53
5.1.3.1 Câmera Personalizada	53
5.1.3.2 Suporte à Orientação do Dispositivo	55
5.1.3.3 Saída por Meio de Voz	57
5.1.4 Testes de Validação	58
5.1.4.1 Teste de Eficiência	58
5.1.4.2 Teste de Usabilidade.....	61
5.2 RESULTADOS OBTIDOS.....	62
6 CONCLUSÃO	64

1 INTRODUÇÃO

A leitura tem um papel muito importante no desenvolvimento social e intelectual humano. A partir do ingresso na vida escolar, ela se torna imprescindível para o acesso ao conhecimento e a informação. Porém, muitas vezes o acesso a esse meio se torna difícil, principalmente para deficientes visuais, seja por falta de recursos tecnológicos, como livros digitalizados e arquivos de áudio, ou por falta de exemplares especializados, como livros em Braille¹, por exemplo.

Atualmente, há uma tendência de que os novos livros sejam digitalizados ou disponibilizados na internet. Porém, existe uma quantidade muito grande de exemplares importantes que poderão continuar disponíveis apenas em meio impresso. Existem ainda documentos e informações não formais, como textos, artigos, trabalhos, relatórios, entre outros, que também podem não estar disponíveis em meio digital.

Com a evolução das tecnologias, diversos recursos surgem buscando tornam atividades cotidianas mais fáceis, tal como promover alternativas para a resolução de problemas rotineiros. A aplicação desses recursos na promoção da acessibilidade a pessoas portadoras de deficiência recebe o nome de Tecnologia Assistiva. O objetivo dessas tecnologias é ampliar a comunicação, trazer mobilidade, possibilitar e facilitar a execução tarefas, promover o desenvolvimento de habilidades e de conhecimento (BERSCH, 2008).

A popularização e a evolução dos dispositivos móveis torna possível a elaboração de novas soluções para auxiliar a execução de tarefas, inclusive na aplicação como Tecnologia Assistiva. Recursos como Reconhecimento Óptico de Caracteres, bibliotecas de conversão de texto em áudio, captura de imagens em alta definição, entre outros, tornam possível a elaboração de inúmeros tipos diferentes de aplicações, para as mais variadas finalidades.

Este trabalho teve como objetivo utilizar técnicas de Reconhecimento Óptico de Caracteres e conversão de texto em áudio para a elaboração de um protótipo de aplicativo para Android, voltado para o apoio à leitura de documentos impressos por parte de usuários portadores de deficiências visuais.

¹ Sistema universal de leitura e escrita desenvolvido na França por Louis Braille, um jovem deficiente visual (LEMOS, 2005).

1.1 OBJETIVO GERAL

O objetivo desse trabalho foi desenvolver um protótipo de aplicativo na plataforma Android para que, de forma acessível, usuários de dispositivos móveis possam realizar a leitura de impressos. Para isso, foram utilizadas técnicas de reconhecimento de palavras em imagens e conversão de texto em voz.

1.2 OBJETIVOS ESPECÍFICOS

Os objetivos específicos aos quais se pretendeu atingir com a elaboração do trabalho são:

- a) estudar metodologias de acessibilidade de deficientes visuais em dispositivos móveis;
- b) utilizar a técnica de reconhecimento óptico de caracteres com a ferramenta Tesseract;
- c) estudar conversão de texto em voz com a biblioteca *TextToSpeech*;
- d) pesquisar formas de interação com o usuário do aplicativo;
- e) desenvolver um protótipo da aplicação;
- f) realizar testes com usuários em potencial;
- g) avaliar a eficiência do protótipo.

1.3 JUSTIFICATIVA

A sociedade atualmente possui diversos problemas na inclusão de pessoas portadoras de deficiências. Seja na falta de conscientização pública, na escassez de infraestrutura adequada ou na indisponibilidade de recursos, a dificuldade de acesso igual a todos é facilmente identificada. Ao analisar os acervos disponíveis em diferentes bibliotecas, por exemplo, podem-se constatar diversos problemas:

- a) há escassez de exemplares transcritos em Braille ou em meio digital. O número de livros disponíveis nesses formatos é muito inferior aos disponíveis na disposição original;
- b) o espaço de armazenagem necessário para volumes em Braille é muito grande, o que se torna um problema para o usuário e para o

- estabelecimento que o fornece;
- c) existe atraso entre a publicação original e a transcrição em Braille ou digitalização do exemplar, prejudicando o leitor;
 - d) a portabilidade de livros em Braille é comprometida, pois os livros são pesados e tem um grande volume, dificultando o transporte dos mesmos;
 - e) textos, artigos, documentos, trabalhos e outros impressos acabam não convertidos para uma forma acessível ao deficiente.

Em outras situações também é possível identificar problemas no acesso a informação. Em cartazes explicativos, folders, rótulos de embalagens, manuais de instrução, cardápios, listas telefônicas, correspondências, dentre outros, que dificilmente são disponibilizados em meio acessível.

Em 2010, dados do Instituto Brasileiro de Geografia e Estatística (IBGE) demonstram que o número de pessoas no Brasil que apresentam alguma deficiência visual ultrapassa os 6,5 milhões, sendo que desses mais de 528 mil declararam que não conseguem de modo algum enxergar (IBGE, 2010). Dessa forma, é possível perceber que o número de pessoas que sofrem com os problemas de falta de acessibilidade é muito grande, o que exige a elaboração de diferentes alternativas para a solução destes problemas.

Com o desenvolvimento de uma aplicação para digitalização instantânea de textos e conversão em arquivo de áudio, problemas como espaço de armazenagem, portabilidade, acesso a impressos e atraso na publicação de versões acessíveis podem ser resolvidos. Com a aplicação das técnicas de reconhecimento óptico de caracteres e conversão de texto em voz, torna-se possível desenvolver uma aplicação que amplia a acessibilidade de deficientes visuais, tornando possível a leitura de livros e muitos outros meios informativos ainda no formato impresso.

1.4 ESTRUTURA DO TRABALHO

Este trabalho está dividido em uma estrutura com seis capítulos. Sendo que o primeiro capítulo traz uma breve descrição sobre o tema do trabalho, contendo a introdução, o objetivo geral e os específicos e ainda a justificativa do trabalho proposto.

O segundo capítulo trata dos assuntos relacionados a deficiências no

contexto social, englobando a sociedade e a acessibilidade, deficiência visual, tecnologias assistivas para deficientes visuais e acessibilidade em dispositivos móveis.

O terceiro capítulo aborda o levantamento teórico referente a todas as tecnologias aplicadas no desenvolvimento do trabalho. Dois grandes grupos se destacam nesse capítulo: Android, com sua estrutura e funcionamento, e Reconhecimento Óptico de Caracteres, tratando de suas etapas e ferramentas existentes.

No quarto capítulo são apresentados trabalhos correlatos ao estudo desenvolvido.

O capítulo quinto traz a estrutura do trabalho desenvolvido, onde aborda-se a metodologia aplicada para o desenvolvimento do protótipo e onde são avaliados os resultados obtidos com a elaboração do mesmo.

O último capítulo, de número seis, traz as conclusões referente ao trabalho desenvolvido, expondo os resultados obtidos em análise aos objetivos da pesquisa, as dificuldades decorrentes da elaboração e as soluções encontradas, e ainda sugestões para trabalhos futuros.

2 DEFICIÊNCIA NO CONTEXTO SOCIAL

Para que se possa fazer uma análise do contexto social ao qual as pessoas com deficiência vivem atualmente, se faz necessário compreender que não somente fatos atuais o determinam, mas também condições históricas que imputam a essas pessoas. Tais condições são imprescindíveis para a compreensão da trajetória desse segmento de pessoas (CARVALHO; TUREK, 2006).

2.1 A SOCIEDADE E AS PESSOAS COM DEFICIÊNCIA

Nos tempos primitivos, no período histórico conhecido como sociedade comunal, os humanos viviam de forma brutal. Nessa época, podia-se considerar uma pessoa com deficiência se ela não conseguisse garantir a própria sobrevivência, ou tivesse uma aparência muito fora do comum. Porém é importante ressaltar que muitas das deficiências que se conhece hoje poderiam passar despercebidas em sociedades de tempos primórdios. Em tempos onde não haviam dialetos desenvolvidos, por exemplo, uma pessoa surda-muda podia viver sem que sua deficiência fosse detectada, pois a comunicação era baseada em gestos e sons. Podem-se ressaltar também pessoas com problemas de postura, curvatura, problemas de costela ou em membros inferiores, que poderiam passar despercebidos por ainda existirem povos com tendência a andar sobre quatro patas (CARVALHO; TUREK, 2006).

Em muitos povos primitivos, considerava-se que o extermínio era a solução para pessoas que tinham algum tipo de deficiência, dentre estas estavam crianças, mulheres, idosos ou quaisquer que apresentassem doenças ou deficiências mentais e físicas (SILVA, 1987). No Brasil, por exemplo, no período pré-colonial, onde as civilizações eram na maior parte formada por indígenas, pessoas com deficiência eram exterminadas ao nascer ou sacrificadas em rituais de purificação por acreditar-se que estas trariam maldição para a tribo (FIGUEIRA, 2008).

Um pouco mais a frente, no período colonial onde o Brasil foi marcado pela escravidão, havia um grande número de pessoas que adquiriam deficiências por motivo de maus tratos, castigos e por meio de doenças que assolavam a população da época. Destes motivos destacam-se os açoitamentos a escravos que

muitas vezes os tornavam cegos ou dilacerados de diversas formas, e também doenças como a paralisia, que afetavam diversas pessoas nesse período. Muitas dessas pessoas eram abandonadas a própria sorte, tendo que mendigar por recursos para sobreviver. Tal condição de marginalização social se mantém por muito tempo, principalmente nas classes menos favorecidas (FIGUEIRA, 2008).

As reivindicações por melhores condições sociais a portadores de deficiência passaram a ser mais presentes após a segunda guerra mundial, a qual foi responsável por muitos soldados e civis passarem a condição de deficientes na sociedade em todo o mundo. Essa realidade resultou na realização de diversas Conferências Mundiais e ainda no Ano Internacional da Pessoa com Deficiência em 1981. Tal período no Brasil coincide com as décadas onde a classe trabalhadora lutou intensamente por melhores condições de vida, que culminou na Constituição Federal nos anos 80, luta essa na qual as pessoas com deficiência se fizeram presentes, batalhando por suas especificidades (CARVALHO; TUREK, 2006).

Desse modo, atingindo âmbito mundial, passou a existir uma maior mobilização por parte dos deficientes e da sociedade em geral por melhores condições de vida, reabilitação e integração social e profissional, reestruturação de ambientes e espaços físicos públicos e privados, garantia de direitos legais, entre outros aspectos que possam promover acessibilidade a todo e qualquer tipo de pessoa. Hoje, o mundo tem uma nova visão sobre as diferenças, mais voltada para a igualdade de direitos e respeito às especificidades de pessoas que possuem alguma necessidade especial, buscando valorizar o potencial de cada um como sujeito. Dessa forma se faz necessário garantir acesso e condições favoráveis a todos, visando o rompimento das barreiras sociais e funcionais geradas pela deficiência, com o intuito de tornar a sociedade mais acessível, justa e igualitária (BRASIL, 2005).

2.2 ACESSIBILIDADE

Na busca por melhores condições de vida para todas as pessoas portadoras de deficiências, surgiu um termo muito utilizado e difundido em todo o mundo: acessibilidade. Segundo a legislação brasileira (BRASIL, 2004), considera-se:

acessibilidade: condição para utilização, com segurança e autonomia, total

ou assistida, dos espaços, mobiliários e equipamentos urbanos, das edificações, dos serviços de transporte e dos dispositivos, sistemas e meios de comunicação e informação, por pessoa portadora de deficiência ou com mobilidade reduzida;

Dessa forma considera-se que acessibilidade é um conjunto de fatores que viabilizam o acesso e a utilização de qualquer ambiente, real ou virtual, por parte das pessoas com deficiência ou mobilidade reduzida, da mesma forma que qualquer pessoa da sociedade.

O principal objetivo da acessibilidade é promover a inclusão social. O processo de inclusão é aquele que se estabelece na sociedade com intenção de melhorar a qualidade de vida, promover a autonomia de renda, o desenvolvimento humano e a igualdade de direitos e oportunidades para pessoas ou segmentos de pessoas que possuem algum tipo de desvantagem em relação aos demais indivíduos da sociedade (PASSERINO; MONTARDO, 2007).

Uma grande aliada nesse processo é a tecnologia. A evolução tecnológica vem promovendo diversos recursos que tornam atividades cotidianas muito mais fáceis. No que diz respeito aos equipamentos, recursos e serviços tecnológicos que auxiliam pessoas portadoras de deficiências, o termo Tecnologia Assitiva (TA) é muito aplicado. Dessa forma a TA tem como objetivo ampliar a comunicação, trazer mobilidade, possibilitar e facilitar a execução tarefas, promover o desenvolvimento de habilidades e de conhecimento através de recursos tecnológicos (BERSCH, 2008).

Dentre os diversos tipos de pessoas portadoras de deficiências que são beneficiadas pelo uso da TA, pode-se destacar o deficiente visual. Diversas tecnologias foram e vem sendo desenvolvidas visando à inclusão deste grupo de pessoas no mundo digital e a promoção de atividades do cotidiano das mesmas.

2.2.1 Tecnologia Assistiva Para Deficientes Visuais

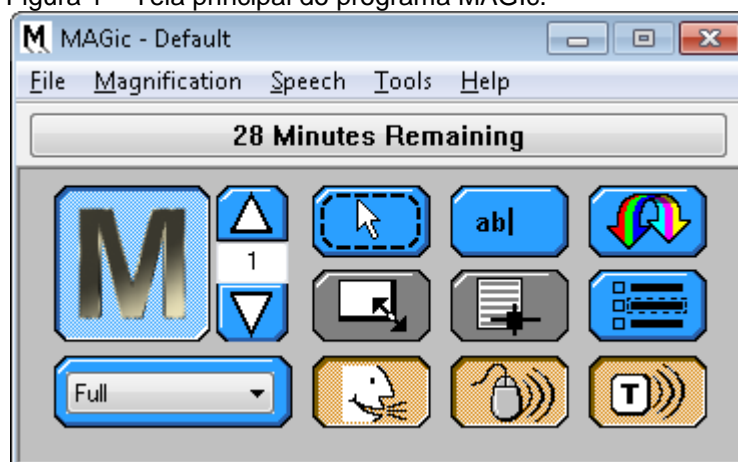
Para melhor compreensão das TAs utilizadas por deficientes visuais, se faz necessário entender alguns dos conceitos aplicados a esse tipo de deficiência. Segundo o Decreto 5.296/04, são considerados dois níveis de deficiência visual: baixa visão e cegueira. A baixa visão compreende uma acuidade visual no intervalo de 0,3 e 0,05 no melhor olho com a melhor correção óptica. A cegueira esta caracterizada pela acuidade de visão menor ou igual a 0,05 no melhor olho, também

contando com a melhor correção óptica (BRASIL, 2004).

Para deficientes visuais com baixa visão, os principais programas para computadores utilizados são os ampliadores de telas, também conhecidos como programas de lupa, zoom em telas, lente de aumento e outros (MELO, 2011). Os programas de ampliação de telas têm como principal função ampliar uma determinada área, variando entre ampliação parcial e total da tela, de forma a permitir a leitura e compreensão dos conteúdos exibidos pelo computador, podendo também ser complementado com sintetizadores de voz. Dentre os programas ampliadores de tela existentes, pode-se destacar:

- a) MAGIc: desenvolvido pela empresa Freedom Scientific, que possui uma linha com diversos produtos para pessoas com deficiência visual e é proprietária do software de leitura de tela JAWS. O ampliador de telas MAGIc chega a ampliar 36 vezes o tamanho da tela e possui também um leitor de telas integrado que ajuda na compreensão do conteúdo da tela (FREEDOM SCIENTIFIC, 2012). Possui diversas opções de personalização, como cor de fundo da tela, tipos de cursores, destaque de foco, definição da área de ampliação, ativação e desativação do sintetizador de voz, dentre outras funções, dispostas em botões conforme a figura 1. O software tem versão para testes com sessões de 40 minutos. A licença de utilização pode ser adquirida através de revendedores ou no site do fabricante;

Figura 1 – Tela principal do programa MAGIc.

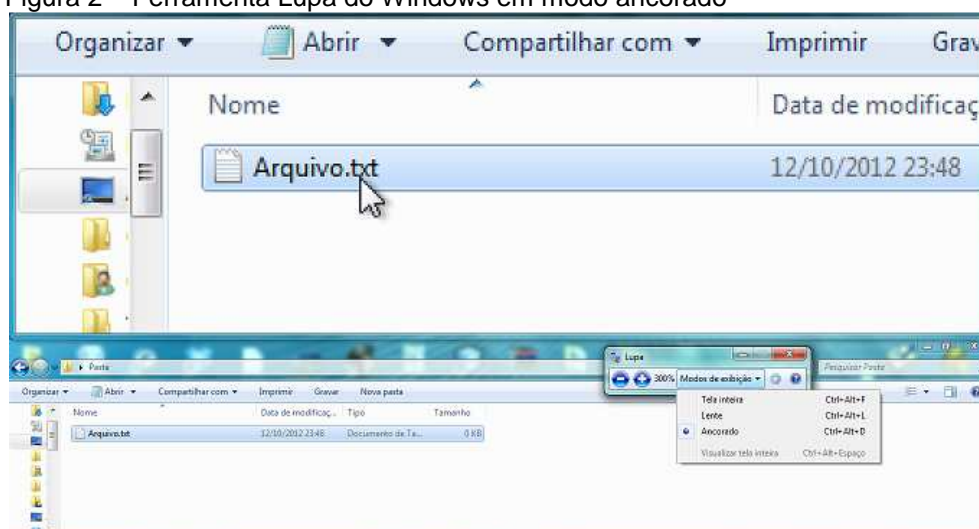


Fonte: Freedom Scientific (2012).

- b) Lente de aumento do Windows ou Lupa: por ser um recurso que já

vem disponível com o sistema operacional Windows, pode ser considerado o programa de ampliação mais acessível para usuários Windows (MELO, 2011). Possui três modos de ampliação: Tela inteira, onde toda a tela assume o percentual de zoom aplicado; Lente, onde o zoom se aplica somente em uma área retangular em torno do ponteiro do mouse, acompanhando a navegação do usuário; e Ancorado, ilustrado na figura 2, onde parte da tela fica fixada com a ampliação aplicada e o restante da tela em tamanho original;

Figura 2 – Ferramenta Lupa do Windows em modo ancorado



Fonte: Microsoft (2009).

c) ZoomText: uma ferramenta com diversos recursos específicos para usuários com baixa visão. Permite diversas configurações para otimização, como melhorias nas cores da tela, ajustes de foco, personalização de ponteiros e cursores, dentre outros, como demonstra a figura 3. Possui sintetizador de voz em várias línguas com possibilidade de gravação dos conteúdos processados pelo mesmo, um modo específico de pesquisas para auxiliar na busca em textos ou de conteúdo na *internet* e ainda um recurso de ampliação de impressos via webcam (AI SQUARED, 2012). O programa esta disponível em uma versão para testes com duração de 60 dias, mas também pode ser adquirido *online*.

Figura 3 – Tela principal do programa ZoomText



Fonte: Ai Squared (2011).

Deficientes visuais com cegueira têm como principal programa para utilização de computadores os Leitores de Telas. São softwares que consistem basicamente na leitura de elementos textuais presentes na tela do computador via sintetizadores de voz.

Dentre os diversos programas para leitura de telas, alguns se destacam pela popularidade:

- a) JAWS for Windows: outro produto da empresa Freedom Scientific, é o software de leitura de telas mais popular do mundo, demonstrado na figura 4. É compatível com a maioria das ferramentas do sistema operacional Windows e diversas aplicações populares, realizando a leitura de praticamente todas as informações que estão na tela. Tem como possibilidade de saída de dados não somente sintetizadores de voz, mas também é compatível com dispositivos de saída em Braille. Possui tecnologia de Reconhecimento Óptico de Caracteres (OCR) que reconhece a presença de informações textuais em imagens na tela, como arquivos em PDF, por exemplo (FREEDOM SCIENTIFIC, 2012). Disponível para *download* nas versões para teste e passível de compra da licença *online*;

Figura 4 – Tela principal do programa JAWS



Fonte: Freedom Scientific (2012).

- b) Virtual Vision: um software de leitura de telas brasileiro, desenvolvido pela empresa MicroPower, que teve a sua primeira versão lançada em 1998 e hoje já esta na versão 7 conforme figura 5, lançada em 2011. Possui suporte aos sistemas operacionais Windows mais atuais e suas ferramentas, incluindo versões 64 bits. Suporta também a aplicações do pacote Office, Live Messenger e Skype, dentre outras aplicações populares, sendo compatível com diversos navegadores. Possui diversas teclas de atalho para facilitar a navegação e permitir a utilização dos mais variados recursos (VIRTUAL VISION, 2012). O produto pode ser testado realizando-se o *download* no site do fabricante, mas possui uma limitação no tempo de utilização, sendo possível adquirir a licença *online*;

Figura 5 – Tela do Painel de Controle do Virtual Vision

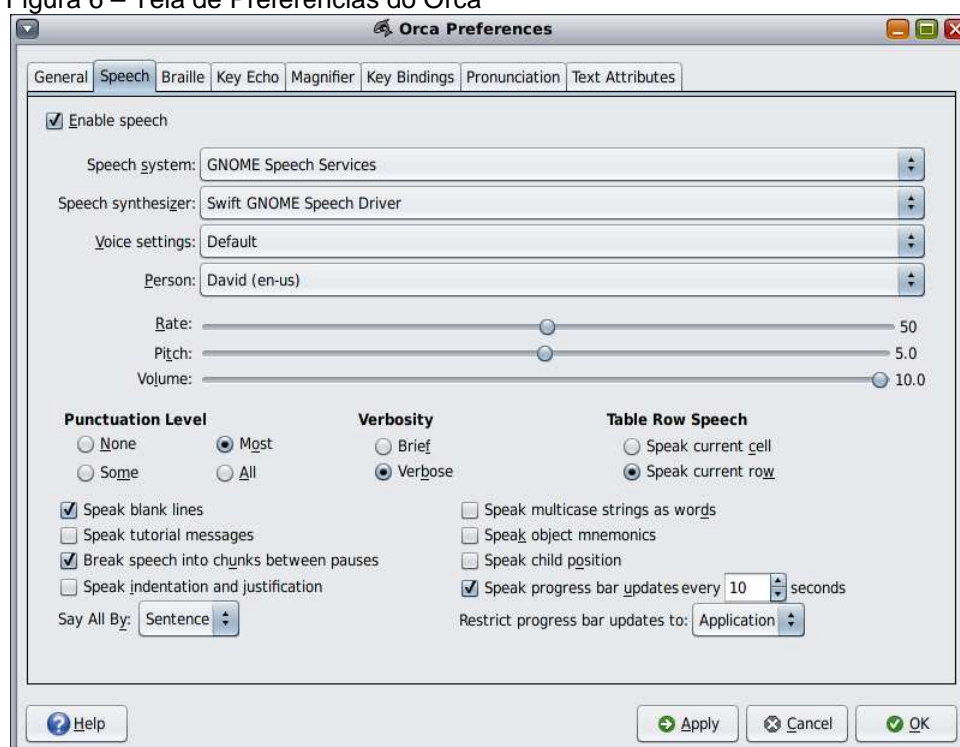


Fonte: MicroPower (2012).

- c) Orca: leitor de telas de código fonte aberto, tem seu desenvolvimento guiado pelo Accessibility Program Office of Sun Microsystems, Inc.

(Oracle) e faz parte da plataforma GNOME², sendo assim utilizado principalmente em ambiente Linux. É projetado para dar assistência na utilização de aplicações como OpenOffice, Firefox, plataforma Java, dentre outros, possuindo recursos de sintetizador de voz, Braille e também um ampliador de telas (ORACLE, 2012). Pode ser facilmente configurado, conforme a figura 6.

Figura 6 – Tela de Preferências do Orca



Fonte: Oracle (2011).

Além dos computadores, novos dispositivos tecnológicos populares vem buscando trazer os conceitos de acessibilidade para viabilizar a sua utilização por parte das pessoas portadoras de deficiência visual. Dentre estes se destacam principalmente os dispositivos móveis, como celulares, *smartfones*, *tablets*, entre outros.

2.2.2 Acessibilidade em Dispositivos Móveis

Uma das tecnologias que mais tem se popularizado é a dos dispositivos

² Projeto internacional de software livre que faz parte do Projeto GNU, com o intuito de oferecer ao usuário final um ambiente simples, atraente, intuitivo e acessível; e para o desenvolvedor um framework para construção de aplicações que se integrem com o desktop (GNOME BRASIL, 2011).

móveis, principalmente *smartphones* e *tablets*. A taxa de crescimento nas vendas destes dispositivos é muito grande. A quantidade de *smartphones* embarcados até o 3º trimestre de 2012 foi de mais de 478 milhões de dispositivos contra 327,6 milhões no mesmo período em 2011, conforme tabela 1. Comparativamente, em todo o ano de 2009 o número de embarques foi de 175,8 milhões (TELECO, 2012).

Tabela 1 – Número de smartphones embarcados no mundo por trimestre (milhões)

Empresa	1T11	2T11	3T11	4T11	1T12	2T12	3T12
Samsung	11,5	18,4	28,1	36	42,2	50,2	56,3
Apple	18,5	20,4	17,1	37	35,1	26	26,9
RIM	13,8	13,8	11,8	10	9,7	7,8	7,7
ZTE	-	2	4,1	3	-	8	7,5
HTC	9	11,6	12,7	10,2	6,9	8,8	7,3
Nokia	24,2	16,7	16,8	19,6	11,9	10,2	6,3
Motorola	4,1	4,4	4,8	5,3	5,1	ND	ND
Outros*	24,5	26	33,1	42	39,1	42,9	67,7
Total	101,6	108,9	123,7	157,8	144,9	153,9	179,7

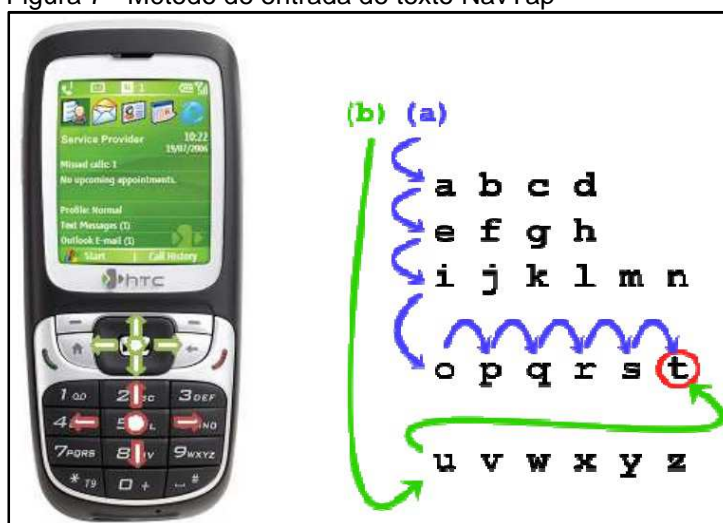
Fonte: Adaptado de TELECO Inteligência em Comunicações (2012).

Junto ao crescimento da utilização, veio à necessidade de criar um ambiente acessível a qualquer tipo de usuário, a fim de proporcionar a execução de diversas tarefas que facilitem o dia-a-dia de seus usuários. Dessa forma são usadas diversas técnicas e estratégias para estabelecer uma melhor interação entre o utilizador com suas capacidades e o dispositivo.

Para usuários com deficiência visual, dispositivos que possuem botões para interação são uma das formas mais utilizadas. Normalmente as teclas são utilizadas para navegação, enquanto o dispositivo responde de forma audível e tátil ao comando inserido, retornando as opções disponíveis (BENEDITO, 2010).

Um exemplo desse tipo de aplicação é o Navtap, um método de introdução de texto voltado para deficientes visuais. Esse método separa o alfabeto em linhas conforme figura 7, sendo que cada uma delas inicia-se com uma vogal e em seguida as consoantes em sequência. Dessa forma o usuário usa as teclas direcionais para navegar entre as letras, sendo que a cada seleção uma resposta sonora indica a letra selecionada e aguarda a seleção (GUERREIRO et al, 2009?).

Figura 7 - Método de entrada de texto NavTap



Fonte: Guerreiro et al (2009?).

Alguns dispositivos são desenhados exclusivamente para deficientes visuais, como é o caso do *ALVA Mobile Phone Organizer (MPO)* e do *PAC Mate Omni*, por exemplo. Estes dispositivos são específicos para atender a usuários com deficiência visual e possuem saída de dados em Braille (BENEDITO, 2010).

O *ALFA MPO* é um assistente pessoal e telefone que permite fazer e receber chamadas e mensagens de texto, realizar anotações e gerenciar contatos. Possui um sintetizador de voz e uma tela Braille para retorno das informações. Pode ser conectado ao computador através de portas USB. Possui diversos botões bem dispostos em sua estrutura, facilitando bastante a sua utilização (AFB ACCESSWORLD, 2004, tradução nossa).

O *PAC Mate Omni* da Freedom Scientific permite acesso a diversas aplicações presentes nos dispositivos tradicionais. Por ser equipado com Windows Mobile, possui suporte aos arquivos do Office, emails, contatos, internet, entre diversos outros recursos. Possui um sintetizador de voz e tela Braille para interação com o usuário (FREEDOM SCIENTIFIC, 2012).

Dispositivos como estes permitem que seja possível que pessoas com deficiência visual tenham acesso aos mesmos recursos que os dispositivos tradicionais oferecem. O problema é o custo para aquisição de dispositivos como estes e a exigência do conhecimento da linguagem Braille, que são fatores que restringem determinados usuários (BENEDITO, 2010).

Com o surgimento da tecnologia de telas sensíveis ao toque, os dispositivos passaram a utilizar técnicas de reconhecimento de gestos como forma

de interação para usuários com deficiência visual. Dessa forma diversas aplicações utilizam diferentes formas de reconhecer os gestos na tela do dispositivo (BENEDITO, 2010).

Um método de introdução de texto que utiliza a abordagem de gestos na tela é o NavTouch, que funciona de forma semelhante ao NavTap citado anteriormente. O diferencial é que a navegação entre as letras se dá por meio de movimentos direcionais desenhados na tela, indicando o sentido desejado. Possui ainda ações especiais localizadas nos cantos da tela, como apagar caracteres, seleção de outros caracteres e inserção de espaço. A organização das letras e o retorno sonoro acontece da mesma forma que o NavTap. O NavTouch possui a grande vantagem de poder realizar os gestos em qualquer área da tela ao invés de procurar os botões como ocorre com o NavTap (GUERREIRO, 2008, tradução nossa).

Seguindo essa mesma idéia de navegação orientada por gestos, grandes fabricantes de dispositivos móveis populares proporcionaram o uso de seus produtos de forma acessível. Dois dos maiores sistemas operacionais de dispositivos moveis do mundo utilizam gestos e retorno sonoro para gerenciar o conteúdo de seus dispositivos. São estes o iOS e o Android.

Os dispositivos Apple contam com um recurso denominado VoiceOver, que se trata de um leitor de telas para sistema operacional iOS. Este leitor também é utilizado em computadores Mac e foi atribuído como recurso padrão aos dispositivos móveis a partir do iPhone 3GS. O VoiceOver funciona a partir da interação direta com a tela do dispositivo, sendo que ele faz a leitura de cada elemento contido na tela conforme o usuário toca em um ponto específico (APPLE, 2012).

O leitor VoiceOver funciona baseado nos movimentos aplicados na tela do dispositivo. Quando algum botão presente na tela é tocado, ele fará a leitura da sua descrição e com um toque duplo o botão é selecionado. Cada ícone tocado assume um cursor de navegação que indica o foco nesse ícone e a partir deste cursor é possível realizar movimentos de deslize para a direita e a esquerda, navegando assim entre os itens presentes na tela, enquanto o leitor descreve cada um deles (APPLE, 2012).

A plataforma Android também possui um mecanismo de acessibilidade padrão integrado, o TalkBack. Trata-se de um leitor de telas também baseado na interação com a tela do dispositivo, retornando uma resposta falada de cada item

tocado na tela, além de eventos e notificações. Permite ainda que outros parâmetros de resposta sejam ativados, como efeitos sonoros, vibrações, modos de exploração, entre outros (SUPPORT GOOGLE, 2012, tradução nossa).

Por ser uma plataforma de código fonte aberto, o Android possui recursos para o desenvolvimento de aplicações sendo que, dentre estes, alguns são utilizados na área da acessibilidade. Um exemplo desses recursos é a biblioteca *TextToSpeech* (TTS), que possui métodos para gerenciar o sintetizador de voz do dispositivo, afim de tornar uma determinada aplicação acessível (DEVELOPER ANDROID, 2012, tradução nossa).

Esses recursos apresentados demonstram a evolução das TAs em diferentes tipos de dispositivos e plataformas, comprovando que com o uso de técnicas existentes e a exploração de novas tecnologias sempre é possível inovar. Fatores como o alto poder de processamento e a popularidade dos dispositivos móveis atuais, tornam possível o desenvolvimento de soluções mais praticas e eficientes, principalmente na área da acessibilidade que tem tanto a evoluir.

3 TECNOLOGIA

As Tecnologias Assistivas surgem tanto da criação quanto da utilização e transformação de recursos tecnológicos já existentes. Muitas delas são criadas a partir da combinação de diferentes técnicas que, quando projetadas em um mesmo sistema, são capazes de atingir novos propósitos ou melhorar os processos já existentes.

Esse capítulo traz uma breve descrição dos recursos tecnológicos utilizados para a elaboração do trabalho proposto, como a plataforma Android, reconhecimento óptico de caracteres e sintetizadores de voz.

3.1 ANDROID

O Android é uma plataforma para dispositivos móveis completa, constituída por um sistema operacional, *middleware*³ e as aplicações que fazem interface com o usuário, além de possibilitar o desenvolvimento de aplicativos por meio de um *Software Development Kit* (SDK). Tal plataforma surgiu de uma parceria realizada entre a Google e a *Open Handset Alliance* (OHA), que consiste na aliança entre empresas que fazem parte das estruturas envolvidas na tecnologia de telefonia móvel. A OHA é composta por mais de 40 empresas e dentre elas estão (OPEN HANDSET ALLIANCE, 2012, tradução nossa):

- a) operadoras de celular;
- b) fabricantes de celulares;
- c) empresas de semicondutores;
- d) empresas de softwares;
- e) empresas de comercialização.

Em 2005, a Google comprou a empresa Android Inc., que desenvolvia um sistema operacional para dispositivos móveis baseado no Kernel do Linux. É em cima desse sistema que é desenvolvido o projeto do Android pela OHA, sendo que este tem seu código fonte aberto e é passível de alterações por desenvolvedores de softwares. Dessa forma o Android tem suas funcionalidades ampliadas

³ Também conhecido como mediador, é o software que faz a comunicação entre o sistema operacional e as aplicações, com intuito de facilitar o desenvolvimento de aplicações e a integração de sistemas (MIDDLEWARE OBJECTWEB, 2007).

constantemente e esta sempre incorporando tecnologia de ponta (GARGENTA, 2011, tradução nossa; OPEN HANDSET ALLIANCE, 2012, tradução nossa).

Com base na versão 2.6 do Kernel do Linux, o Android possui uma arquitetura bem estruturada e baseada em camadas, de forma a tratar isoladamente as funções de aplicações, execuções paralelas, serviços centrais, entre outros (PEREIRA; SILVA, 2009).

3.1.1 Arquitetura Android

Para desenvolver aplicações em Android, como em qualquer outro sistema operacional, é importante que se conheça sua arquitetura e o modo como estão dispostas suas camadas, para compreender de que forma ocorrem as interações dos processos dentro dessa plataforma. Dessa forma é imprescindível uma breve percepção de sua disposição.

Conforme demonstra a figura 6, a arquitetura do Android esta dividida em quatro camadas: Aplicações, *Framework*, Bibliotecas/Android Runtime e Kernel Linux.

Figura 8 – Arquitetura do Android



Fonte: Pereira e Silva (2009).

3.1.1.1 Aplicações

A camada mais alta, ou seja, que se encontra acima das outras é a camada de aplicação. Nela se encontram as aplicações padrões do sistema, que já vem com o próprio dispositivo no momento da compra, como discador do telefone, gerenciador de emails, agenda de contatos, navegador para Internet, calendários, dentre outros. Nessa camada também ficam as demais aplicações pessoais do usuário, que podem ser adquiridas na própria loja virtual Google Play, ou desenvolvidas pelos próprios usuários por meio do SDK (BURNETTE, 2010, tradução nossa).

As aplicações Android utilizam a linguagem de programação Java. São usadas as ferramentas do SDK para compilar o código, juntamente com os dados e arquivos de recursos, em um pacote Android, que é um arquivo com a extensão *.apk*. Todo o código que está dentro desse pacote *.apk* é considerado uma aplicação e através desse arquivo ocorre a instalação no dispositivo (DEVELOPER ANDROID, 2012).

Baseando-se no Kernel do Linux, o sistema operacional Android utiliza um sistema multiusuário, onde cada aplicação é um usuário diferente. A cada usuário é atribuída uma ID, através da qual o sistema faz as definições de permissão de acesso aos arquivos desse aplicativo, filtrando assim os usuários (aplicações) que terão acesso a esses dados. Quando executada, cada aplicação também possui um próprio processo Linux, iniciando e finalizando os componentes conforme a necessidade. Desse modo, o sistema Android garante, por padrão, que cada aplicativo terá acesso somente aos componentes necessários para seu funcionamento, criando um ambiente seguro e controlado onde os aplicativos não acessam partes do sistema a qual não tem permissão (DEVELOPER.ANDROID, 2012, tradução nossa).

3.1.1.2 Framework Application

A camada de *Framework* é onde estão localizadas as *Application Program Interfaces* (API) e os recursos em alto nível disponíveis aos desenvolvedores para a criação de suas aplicações. Ela proporciona diversas bibliotecas Java especialmente desenvolvidas para Android, além de diversos serviços e gerenciadores voltados

para utilização dos recursos do dispositivo (GARGENTA, 2011).

Alguns dos principais elementos dessa camada são (BURNETTE, 2010, tradução nossa; GARGENTA, 2011; PEREIRA; SILVA, 2009):

- a) *Activity Manager*: é responsável pela gerência do ciclo de vida e controle do deslocamento entre as aplicações e suas atividades;
- b) *Package Manager*: gerencia a leitura das informações contidas nos *Application Packages* (APKs), que são os pacotes de aplicativos Android. Faz a comunicação com todo o sistema, informando quais pacotes estão sendo utilizados no dispositivo e suas respectivas capacidades;
- c) *Window Manager*: faz o controle e gerenciamento das janelas das aplicações;
- d) *Content Provides*: proporciona o compartilhamento e publicação de dados, tal como compartilhamento de conteúdo, gerenciamento de pacotes e gestão de conteúdo. Dessa forma uma aplicação pode acessar informações de outra ou compartilhar das mesmas informações, conforme suas permissões;
- e) *View System*: permite o tratamento gráfico da aplicação, incluindo botões, frames e layouts.
- f) *Resource Manager*: gerencia os recursos provenientes de arquivos adicionais e conteúdo estático que o código do aplicativo utiliza, como bitmaps, instruções de animação, definições de layout e outros.
- g) *Location Manager*: fornece acesso a serviços de localização do sistema, como informações geográficas do dispositivo, muito utilizada em aplicações que utilizam GPS, por exemplo.
- h) *Notification Manager*: possibilita a emissão de mensagens para o usuário do sistema, alertando sobre eventos ou notificando situações que ocorrem com o dispositivo, até mesmo em aplicações que estão em *background* (segundo plano).

Nessa camada ainda estão disponíveis outros recursos para gerenciar elementos como Bluetooth, Wi-Fi, USB, sensores e outros (PEREIRA; SILVA, 2009). Os elementos do *framework* citados já vêm pré-instalados com o Android, mas é possível realizar extensões conforme a necessidade dos desenvolvedores (BURNETTE, 2010, tradução nossa).

3.1.1.3 Bibliotecas

Na camada de bibliotecas se encontram as bibliotecas nativas do Android escritas em C/C++ que são utilizadas por recursos do sistema, sendo compiladas especificamente para a arquitetura de dispositivos móveis. Elas podem ser acessadas através dos *frameworks* disponibilizados para desenvolvedores. Dentre as principais bibliotecas existentes, tomam destaque as seguintes (BURNETTE, 2010, tradução nossa; PEREIRA; SILVA, 2009):

- a) Surface Manager: proporciona acesso ao sistema de exibição. Permite que o sistema utilize efeitos como transparência e transições sofisticadas, além de realizar o tratamento e gerenciamento das exibições gráficas;
- b) 2D e 3D Libraries: através de algumas bibliotecas gráficas, é possível elaborar interfaces combinando elementos bidimensionais e tridimensionais. Com bibliotecas 3D baseadas em OpenGL, é possível otimizar os gráficos através de aceleração do hardware 3D, se disponível no dispositivo, ou então renderização desses gráficos via software;
- c) Media Libraries: essas bibliotecas dão suporte a diversos formatos de áudio e vídeo, tal como plugins de codec para hardware e software;
- d) SQLite: biblioteca desenvolvida em C que disponibiliza mecanismos de um banco de dados relacional, leve e eficiente, ideal para dispositivos móveis.
- e) Webkit: tem como base o código do renderizador de páginas de navegadores *Browser Webkit*, suportando tecnologias web como *javascript*, *DOOM*, *CSS* e *AJAX*.

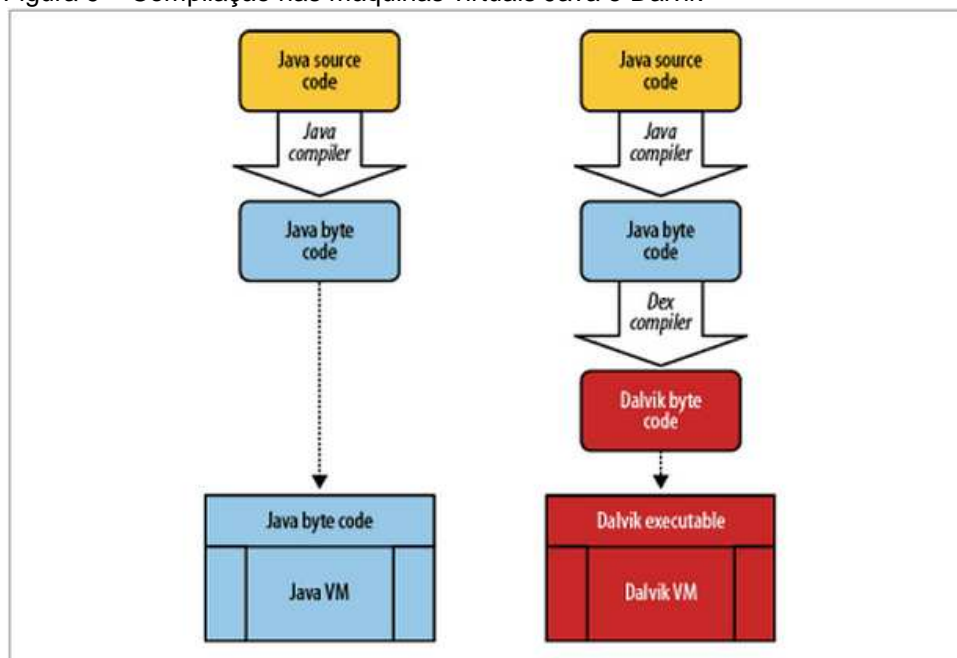
É possível realizar o desenvolvimento de outras bibliotecas por meio do *Native Development Toolkit* (NDK), disponível a partir do Android 1.5. O NDK é um conjunto de ferramentas que permitem a implementação de aplicativos utilizando código nativo, como C e C++, possibilitando a utilização de bibliotecas já escritas nessas linguagens de forma integrada ao aplicativo (DEVELOPER ANDROID, 2013, tradução nossa).

3.1.1.4 Android Runtime

A camada do Android Runtime é o ambiente onde ficam as instâncias da Máquina Virtual Dalvik, criada na execução das aplicações. A Máquina Virtual Dalvik foi desenvolvida para obter melhor desempenho em dispositivos móveis, funcionando em sistemas com pouca memória RAM, baixa frequência de CPU e sistemas operacionais sem recursos para *Swap*, visando baixo consumo de bateria, memória e processamento. Foi projetada para rodar diversas instancias paralelamente (PEREIRA; SILVA, 2009).

A Máquina Virtual Dalvik se difere da Java tradicional, pois executa arquivos *.dex* (nome proveniente de *Dalvik Executable*), que são convertidos no momento compilação a partir dos arquivos *.jar* e *.class*, conforme demonstra a figura 9. Os arquivos *.dex* são mais compactos e eficientes do que arquivos *.class*, o que é muito importante para conservação de espaço em memória e economia de bateria em dispositivos móveis (BURNETTE, 2010, tradução nossa). Os arquivos *.dex* são gerados através da ferramenta DX que esta contida no SDK do Android (PEREIRA; SILVA, 2009).

Figura 9 – Compilação nas máquinas virtuais Java e Dalvik



Fonte: Gargenta (2011).

3.1.1.5 Kernel Linux

O Android foi construído sobre uma base muito sólida: o Kernel Linux.

Situado na camada mais profunda da arquitetura, o Linux fornece uma abstração entre o hardware as demais camadas, o que permite uma ampla portabilidade entre diferentes plataformas (BURNETTE, 2010, tradução nossa).

O Android utiliza o Kernel do Linux para seus serviços fundamentais, como gerenciamento de memória e processos, segurança, protocolos de rede e outros. Utiliza ainda um recurso de gerenciamento de energia, que checa quais dispositivos não estão sendo utilizados por aplicações e os finaliza (PEREIRA; SILVA, 2009).

3.1.2 Componentes das Aplicações

Uma aplicação Android pode ter varias funcionalidades diferentes, como editar notas, reproduzir musicas, tocar um alarme ou abrir as informações de um contato telefônico. Tais funções podem ser de quatro tipos diferentes: *Activity*, *Service*, *BroadcastReceiver* e *ContentProvider*. Cada um desses tipos de funções são denominados como componentes da aplicação (STEELE; TO, 2010, tradução nossa). A tabela 2 exemplifica cada componente de aplicação.

Tabela 2 – Os Quatro Possíveis Componentes de um Aplicativo

Funcionalidade	Classe Base Java	Exemplo
Algo focado que o usuário possa fazer	<i>Activity</i>	Editar uma nota, jogar um jogo
Processo em <i>background</i>	<i>Service</i>	Tocar uma musica, atualizar ícone de clima
Receber mensagens	BroadcastReceiver	Disparar alarme referente a um evento
Gravar e recuperar dados	ContentProvider	Abrir um contato telefônico

Fonte: Steele e To (2013, tradução nossa).

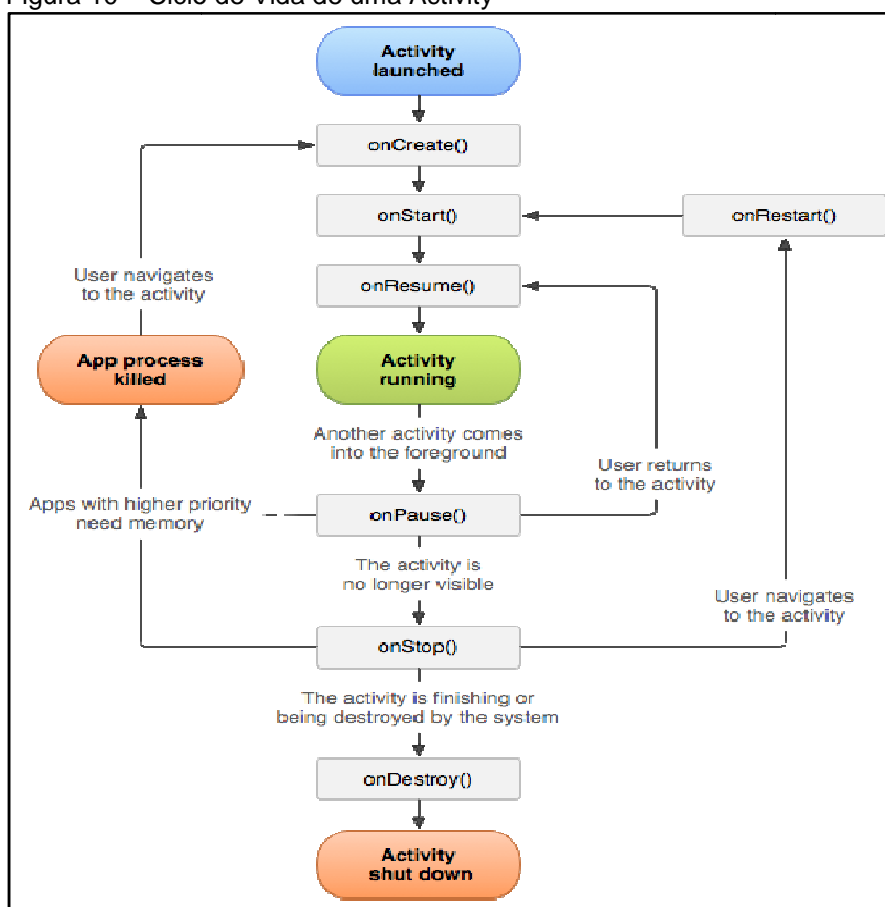
3.1.2.1 Activities

Uma atividade (*activity*) representa uma única tela da aplicação, a qual interage com o usuário (KOMATINENI; MACLEAN, 2012, tradução nossa). Para que se possa entender melhor, em um aplicativo gerenciador de emails por exemplo, pode haver uma *activity* para criar um e-mail, uma outra para listar os e-mails da

caixa de entrada e uma para ler um email. Mesmo que essas *activities* trabalhem juntas, cada uma é independente. Sendo assim, uma aplicação de câmera pode chamar uma *activity* de um aplicativo de e-mails, para enviar a foto a alguém por exemplo (DEVELOPER ANDROID, 2013, tradução nossa).

A *activity* é uma especialização da classe *Activity* e tem uma série de estados diferentes, que compõem o ciclo de vida da *activity* (PEREIRA; SILVA, 2009). O ciclo de vida da *activity* é composto pelos estados demonstrados conforme a figura 10.

Figura 10 – Ciclo de Vida de uma Activity



Fonte: Android Developers (2013).

Normalmente, uma aplicação é composta por diversas *activities*, que são chamadas a partir de uma *activity* principal. Cada vez que uma nova *activity* é iniciada a anterior é interrompida, mas o sistema mantém essas *activities* anteriores em uma pilha, denominada *back stack*. Essa pilha funciona no modo “último que entra, primeiro que sai”, sendo assim quando a *activity* atual é finalizada, ela é destruída e retirada da pilha, e a *activity* anterior que à invocou é resumida

(DEVELOPER ANDROID, 2013, tradução nossa).

3.1.2.2 *Services*

Os *services* são tarefas designadas a manter-se funcionando, independente de qualquer *activity* (MURPHY, 2009, tradução nossa). Um *service* roda em *background*, ou seja, em segundo plano, e não possui interface para comunicação com o usuário. Podem ser utilizados para efetuar qualquer ação, como uma *activity* faria, porém sem interface. Dessa forma, são ideais para tarefas que precisam ser executadas ao mesmo tempo em que outras *activities* estão em execução, como checar por atualizações ou executar um tocador de músicas, por exemplo (GARGENTA, 2011, tradução nossa).

Um *service* pode assumir duas formas diferentes (DEVELOPER ANDROID, 2013, tradução nossa):

- a) *Started Service*: quando algum componente da aplicação, como uma *activity* por exemplo, iniciá-lo a partir do método *startService()*. A partir de então, ele pode rodar em *background* por tempo indeterminado, ainda que o componente que o iniciou venha a ser destruído. Esse tipo de serviço normalmente é utilizado para operações que não precisam retornar resultados ao componente que o chamou, como para efetuar um *download* de arquivo, por exemplo. Ao finalizar sua tarefa, o *service* deve parar a si próprio.
- b) *Bound Service*: é um *service* que possui uma ligação com o componente que o invocou. Ele oferece uma interface cliente-servidor que permite interações, como enviar requisições, receber resultados e até mesmo realizar comunicação entre processos. Esse tipo de *service* fica ativo enquanto o componente que o chamou estiver em funcionamento. Pode estar ligado a vários componentes, porém assim que todos componentes se desvincularem, o serviço é finalizado.

3.1.2.3 *Content Providers*

Content Providers são componentes que proporcionam um nível de abstração para os dados armazenados no dispositivo que podem ser acessados por

diversos aplicativos. O modelo de desenvolvimento Android permite que o desenvolvedor gerencie os dados de sua aplicação e a forma como eles serão acessados por outras aplicações. A construção de um *Content Provider* permite total controle sobre como esses dados são acessados (MURPHY, 2011, tradução nossa).

Segundo Burnette (2010, tradução nossa), a melhor forma de compartilhar dados globais entre diferentes aplicações é através de *content providers*. Um exemplo de utilização é o *content provider* de contatos, disponibilizado pela Google, onde ficam os dados de qualquer contato, como nome, endereço, telefone. Esses dados podem ser acessados por qualquer aplicativo que precisarem dessas informações.

3.1.2.4 *Broadcast Receivers*

Broadcast receivers são componentes responsáveis pela resposta aos anúncios emitidos pelo sistema em *broadcast*, ou seja, de forma difundida a todas as aplicações. Muitos desses anúncios tem origem do próprio sistema, como anúncios de que a tela está desligada, a bateria está fraca, ou foi capturada uma imagem, por exemplo. Sendo assim, o *broadcast receiver* pode executar alguma tarefa, ou notificar o usuário, com base na informação emitida pelo sistema (PEREIRA; SILVA, 2009).

Para criar um *broadcast receiver*, basta implementar a interface *BroadcastReceiver*. Esta interface possui apenas um método, *onReceive()*, onde é implementado tudo que é necessário ser feito, baseado na *intent* (informação) recebida (MURPHY, 2011, tradução nossa).

Um *broadcast receiver* é válido somente durante a execução do método *onReceive()*. Quando este é finalizado, o sistema o considera como objeto a ser concluído e não ativo. Dessa forma o sistema está livre para encerrar o seu processo (DEVELOPER ANDROID, 2013, tradução nossa).

3.2 RECONHECIMENTO ÓPTICO DE CARACTERES

Os primeiros softwares de reconhecimento de caracteres surgiram por volta dos anos 60. O principal nome dessa tecnologia é David Hammond Shepard, fundador da empresa *Intelligent Machines*. Inicialmente ele construiu juntamente

com seu amigo Harvey Cook Jr., uma máquina chamada Gismo no sótão de sua casa. Tal máquina inicialmente era capaz de reconhecer 23 letras do alfabeto, mas depois de alguns anos conseguiram desenvolvê-la para reconhecer os 26 caracteres. Tal tecnologia foi nomeada como *Optical Character Recognition* (OCR), no português, reconhecimento óptico de caracteres (NEW YORK TIMES, 2007).

A tecnologia de OCR é um sistema de processamento de imagens que consiste na extração de texto a partir de imagens digitais. Estas imagens podem ser obtidas a partir de páginas impressas como livros, revistas, jornais, documentos, dentre outros, que são digitalizadas por aparelhos scanners ou máquinas digitais, tal como imagens já armazenadas em meio digital (ALVES, 2003).

O processo de OCR consiste em separar os caracteres das outras informações que estão contidas na imagem para obter somente o texto desta (SOUZA, 2011). Normalmente essas informações são compostas por figuras (gráficos, diagramas, fotos, gravuras, e outros) e texto. É importante identificar e classificar essas informações para que o OCR melhore seu desempenho, pois assim não perdesse tempo tentando interpretar áreas sem informação textual (ALVES, 2003).

3.2.1 Etapas do Reconhecimento Óptico De Caracteres

O processo de reconhecimento de caracteres, tal como outros sistemas de processamento de imagens, é composto por etapas distintas. Suas principais etapas são (SOUZA, 2011):

- a) aquisição de dados: é realizada a digitalização da imagem, que pode acontecer por meio de scanner, câmera digital, dispositivos móveis, entre outros;
- b) processamento da imagem: nessa etapa são realizadas as correções na imagem para otimizar o processo de reconhecimento. São realizadas análises de rotação, reduções de ruídos, verificação de junção de caracteres, segmentação da imagem, dentre outros;
- c) análise de características: identificação de características como tamanho de fonte, diagramação e layout da imagem, entre outros aspectos que podem facilitar o processo de reconhecimento;
- d) geração do arquivo de saída: gera o arquivo em formato de texto

contendo os dados extraídos da imagem.

Cada etapa dentro do processo de OCR é fundamental e determinante para que o resultado seja obtido com êxito. Uma das etapas com maior grau de relevância, como citado anteriormente, é a do processamento da imagem. Através de métodos de pré-processamento pode-se reduzir problemas ocasionados nas etapas de aquisição de dados.

3.2.2 Pré-processamento de Imagens

A aplicação de técnicas de pré-processamento sobre as imagens tem como objetivo otimizar os resultados obtidos no processo de OCR. Tais técnicas são realizadas através de algoritmos eficientes desenvolvidos para cada finalidade (ALVES, 2003). Destacam-se algumas técnicas no processo de OCR:

- a) histograma: é o conjunto de números que indicam o percentual de pixels, em uma determinada imagem, que possuem um nível de cinza específico. A partir desses valores, pode-se criar uma representação gráfica do seu histograma e então avaliar a qualidade de itens como contraste e brilho médio da imagem (MARQUES FILHO; VIEIRA NETO, 1999).
- b) filtragem: busca minimizar os efeitos de degradação de uma imagem gerados por falta de foco, ruídos, turbulência, entre outros. O uso de filtros também pode ser aplicado para gerar efeitos em imagens. Diversos tipos de filtros podem ser aplicados para diferentes finalidades, como por exemplo, filtros de média e mediana para remoção de ruídos e filtros de realce para destacar detalhes finos na imagem (ALVES, 2003).
- c) limiarização: consiste na separação das regiões da imagem em duas classes, o fundo e o objeto. A forma mais simples de aplicar essa técnica é separando o histograma de uma imagem em duas partes a partir de um valor limiar, e então converter os pixels com tons de cinza menores ou iguais a esse valor em brancos e os demais em preto. Pelo resultado obtido nessa técnica, ela também pode ser chamada de binarização (MARQUES FILHO; VIEIRA NETO, 1999).
- d) rotação: no momento da aquisição da imagem, o documento pode

estar mal posicionado, podendo dessa forma ficar inclinado em seu eixo. Desse modo, a imagem pode gerar falhas no processo de reconhecimento. O sistema de detecção do grau de rotação pode ser feito através de cálculos matemáticos baseados em projeções de caracteres (ALVES, 2003).

Diversas outras técnicas também podem ser utilizadas nessa etapa a fim de proporcionar um resultado mais satisfatório no processo de reconhecimento. Tais procedimentos podem ocorrer dentro da própria aplicação de OCR, como também em aplicações externas especializadas no tratamento da imagem.

3.2.3 Ferramentas de Reconhecimento Óptico de Caracteres

Atualmente existem diversas aplicações de OCR disponíveis no mercado, tanto código fonte aberto quanto código proprietário. Algumas delas se destacam por serem conhecidas e utilizadas em diversos projetos, tal como por sua precisão no processamento de seus motores OCR.

A aplicação Tesseract é um mecanismo de OCR desenvolvido inicialmente pela Hewlett-Packard (HP) em 1985 e passou a ser mantido pela Google em 1995. Pode trabalhar em conjunto com a biblioteca de processamento de imagem Leptonica, conseguindo assim ler uma grande variedade de formatos de imagens e trabalhar com mais de 60 idiomas. Já foi eleito um dos três principais mecanismos de OCR em testes de precisão. Seu código fonte é aberto sob a licença Apache 2.0 e roda em diversas plataformas, como Linux, Windows, Mac OSX, entre outros (TESSERACT OCR, 2012, tradução nossa).

O GORC é um programa OCR desenvolvido sob a licença pública GNU. Foi desenvolvido inicialmente por Joerg Schulenburg, que ainda mantém a equipe de desenvolvimento (GOOCR, 2012).

O SnapVision é uma solução para captura de documentos em imagens desenvolvido inicialmente como TopOCR e voltado para dispositivos móveis, que atualmente combina uma câmera própria de 5 *megapixels* com seu software de OCR. É uma ferramenta de código proprietário e com valor de aquisição (TOPSOFT, 2012).

O OCRopus é um sistema de OCR escrito em NumPy, Python e Scipy, e foca-se na aprendizagem de máquina em grande escala, aplicada na problemática

da análise dos documentos. Também está sendo mantido junto aos projetos da Google e também está aberto sob a Licença Apache 2.0 (OCROPUS, 2012).

O GNU Ocrad é um programa OCR baseado em métodos de extração de características. Suporta imagens no formato *Bitmap* (BMP), *Greyscale* (PGM) e *Portable Pixel Map* (PPM). Separa o layout dos documentos analisados em blocos e colunas de textos encontrados na imagem. O Ocrad pode funcionar de forma independente ou dar suporte a outras aplicações (GNU OCRAD, 2012).

4 TRABALHOS CORRELATOS

Diversas pesquisas e desenvolvimentos são realizados a todo o momento no mundo inteiro na área de tecnologia. Algumas áreas como tecnologia móvel e recursos Web são muito exploradas, devido a grande quantidade de usuários a qual essas plataformas atingem e a variedade de recursos para o desenvolvimento que elas disponibilizam.

Nos tópicos subsequentes são abordadas algumas dessas pesquisas relacionadas ao presente trabalho.

4.1 OCRDROID

O OCRdroid é um *framework* para utilização de OCR em dispositivos moveis desenvolvido por três universitários da Universidade do Sul da Califórnia, das faculdades de Ciência da Computação e Engenharia Elétrica. Em 2009, Anand Joshi, Mi Zhang e Ritesh Kadmawala desenvolveram uma estrutura genérica para desenvolvimento de aplicações com recurso de OCR em dispositivos móveis.

O OCRdroid utiliza um leve pré processamento de imagem dentro do dispositivo, combinado com um servidor de *back-end* para execução do processamento OCR. Foi elaborado sobre a plataforma Android e utiliza diversos recursos do dispositivo, como a obtenção da imagem através da câmera, a utilização do acelerômetro para detecção do desalinhamento da imagem e processamento local para identificação do enquadramento do texto, por exemplo.

Em seu servidor de *back-end* são processadas as imagens recebidas do dispositivo, que então passam por processos de pré-processamento para depois serem submetidas ao processo de OCR. Para o OCR é utilizado o código-fonte aberto da ferramenta Tesseract, da empresa Google.

Um exemplo de aplicação utilizando esse *framework* é o PocketPal, desenvolvida pelos próprios acadêmicos para testar o *framework*. Nela, o usuário pode fotografar recibos de compras realizadas que são então enviadas ao servidor, para que se possa manter um histórico dessas compras. Após o processamento, o recibo estará disponível para edição e extração de informações, tal como para consultas posteriores.

4.2 PRÉ-PROCESSAMENTO DE IMAGENS PARA O RECONHECIMENTO ÓPTICO DE CARACTERES

Em 2011, o acadêmico Vanderval Borges de Souza, do curso de Ciência da Computação na Universidade do Extremo Sul Catarinense elaborou um estudo sobre técnicas de pré-processamento de imagens para reconhecimento óptico de caracteres. Com o conhecimento obtido, desenvolveu uma aplicação Java para aplicar as técnicas sobre as imagens (SOUZA, 2011).

A aplicação faz a leitura de uma imagem selecionada pelo usuário e carrega em uma área para visualização. Então se disponibiliza uma série de filtros a serem escolhidos pelo usuário, que pode definir a melhor sequencia de filtros a serem aplicados. Depois a imagem estará pronta para ser submetida ao programa de OCR (SOUZA, 2011).

Dentre as opções possíveis estão: limiarização, rotacionar, mediana e filtros. A limiarização é utilizada para determinar o nível de cinza da imagem, buscando torna-la mais nítida. A rotação pretende manter a imagem no ângulo correto para melhorar o desempenho do OCR. A aplicação dos filtros de mediana procura reduzir a quantidade de ruídos (SOUZA, 2011).

Para verificar a eficiência da utilização do pré-processamento, foi aplicada a métrica da Distância de Levenshtein, responsável por medir o índice de acerto e correções a serem realizadas no processo de OCR. Nas imagens submetidas ao pré-processamento a eficiência do processo foi até 12.84% maior, comprovando a relevância da aplicação dessa técnica no processo de OCR (SOUZA, 2011).

4.3 ESTUDO DE INTERFACES ACESSÍVEIS PARA USUÁRIOS COM DEFICIÊNCIA VISUAL EM DISPOSITIVOS MÓVEIS TOUCH SCREEN

Na Universidade Federal do Ceará, um grupo de pesquisadores realizou um estudo sobre o contexto de acessibilidade em dispositivos moveis com tecnologia *touch screen*. Agebson Rocha Façanha, Windson Viana e Mauro Cavalcante Pequeno fizeram, em 2011, uma comparação entre as principais tecnologias assistivas e avaliaram, junto a um grupo de portadores de deficiência visual, uma proposta de interface acessível na plataforma Android (FAÇANHA,

PEQUENO, VIANA, 2011).

Por meio de reuniões com grupos de desenvolvedores com conhecimento em acessibilidade e de deficientes visuais voluntários da Associação de Cegos do Estado do Ceará, foi elaborado um projeto de interação em forma de protótipo de baixa fidelidade. Através deste foi feita a simulação de execução, testes e a identificação de problemas (FAÇANHA, PEQUENO, VIANA, 2011)..

A partir de então, foram levantados os requisitos básicos para acessibilidade em interfaces *touch screen*. Alguns deles foram (FAÇANHA, PEQUENO, VIANA, 2011):

- a) sempre fornecer um retorno para todas as ações;
- b) preferência de usar gestos no lugar de botões;
- c) quando houver botões, incluir um modo de exploração;
- d) apresentar opções em formato de listas ou em colunas que tenham como referencia as laterais do dispositivo;
- e) avisos e *pop-ups* devem ocupar a tela toda;
- f) evitar uso de expiração de sessão (*time out*), para não desorientar os usuarios;
- g) se o *design* não for adaptado a rotação, fixar uma orientação do *layout*;
- h) incluir opção de contraste para usuários com baixa visão.

Após esses requisitos serem levantados, foi desenvolvida uma proposta de interface acessível para facilitar a inclusão de usuários com deficiência visual na rede social Twitter. O programa foi denominado LêBrailleTWT. Seu modo de utilização é baseado em uma série de gestos que podem ser aplicados sobre a tela (FAÇANHA, PEQUENO, VIANA, 2011).

Outra proposta é o LêBraille, um método de entrada de textos que procura simular os instrumentos de escrita em Sistema Braille. O aplicativo possui uma célula Braille onde podem ser selecionados os pontos para compor as letras, além de uma série de movimentos disponíveis para interação com o dispositivo (FAÇANHA, PEQUENO, VIANA, 2011).

Após o desenvolvimento dessas aplicações, foram realizados testes com os usuários a fim de validar a utilização da interface. Diversas sugestões de melhorias foram identificadas e alguns problemas de execução também, por ser um protótipo de software. Mesmo dessa forma, bons resultados forma obtidos, sendo

que 40% dos usuários que realizaram os testes conseguiram cumprir a tarefa com sucesso (FAÇANHA, PEQUENO, VIANA, 2011).

4.4 SISTEMA INTEGRADO PARA BIBLIOTECAS E ARQUIVOS DIGITAIS

O Sistema Integrado para Bibliotecas e Arquivos Digitais (SInBAD) é um projeto da Universidade de Aveiro, em Portugal, que tem como finalidade a construção de um sistema para armazenamento dos documentos produzidos na universidade, como teses, dissertações, fotografias, vídeos, entre outros. Suporta vários tipos de arquivos, mas tem em sua maioria documentos do tipo PDF e imagens (CAMPOS, 2009).

Em 2009, o então mestrando Américo José de Oliveira Campos realizou uma pesquisa para a implantação de um conjunto de serviços Web para apoio a conversão dos formatos de arquivo PDF e imagem em arquivo texto, dentre outros formatos. Como estes processos podem ser demorados, projetou-se ainda um sistema de computação em Grid, de forma paralela, para otimizar a execução desses processos (CAMPOS, 2009).

Foram desenvolvidos quatro *Web Services* para diferentes funções: extração de texto de ficheiros PDF, redimensionamento e conversão de imagens, reconhecimento de texto por mecanismo OCR e remoção das margens brancas dos documentos (CAMPOS, 2009).

Como resultado da aplicação dos *Web Services* em uma estrutura de execução paralela, foi possível perceber que ao aumentar o número de unidades de execução, o tempo de processamento caiu drasticamente, principalmente na mudança de uma para duas unidades. Um ficheiro de 7,25MB, por exemplo, caiu de mais de 10 minutos para aproximadamente 3 minutos de execução no processo de OCR, executando com uma e duas unidades de execução, respectivamente (CAMPOS, 2009).

Dessa forma foi possível perceber que esse tipo de tecnologia é muito importante para otimização do tempo de processamento, principalmente em aplicações de alta complexidade ou que demandam mais capacidade de processamento (CAMPOS, 2009).

5 O APLICATIVO ANDROID PARA EXTRAÇÃO DE TEXTO EM IMAGENS E CONVERSÃO EM VOZ

Aplicando os conhecimentos adquiridos durante o levantamento bibliográfico referente a acessibilidade para deficientes visuais, reconhecimento óptico de caracteres e conversão de texto em voz, foi possível o desenvolvimento do protótipo de um aplicativo Android para conversão de imagem em voz. Foram aplicadas ainda formas de interação com o usuário do aplicativo, também obtidas no processo de fundamentação teórica.

5.1 METODOLOGIA

As etapas metodológicas utilizadas durante todo o desenvolvimento foram: levantamento bibliográfico; reconhecimento da ferramenta de OCR Tesseract; pesquisa sobre desenvolvimento em Android e a biblioteca TextToSpeech para conversão de texto em voz; implementação do aplicativo e realização de testes; análise de desempenho e descrição dos resultados obtidos.

O levantamento bibliográfico teve como objetivo trazer o embasamento necessário referente aos temas que compõem a estrutura do projeto de pesquisa, como aspectos da acessibilidade e tecnologias assistivas, os sintetizadores de voz, o desenvolvimento em Android e o Reconhecimento Óptico de Caracteres.

5.1.1 A Ferramenta OCR Tesseract

A ferramenta escolhida para a realização do reconhecimento do texto foi a Tesseract, pois além de ser uma biblioteca *open-source* com um excelente histórico de desenvolvimento, tendo na sua lista de desenvolvedores empresas como *Hewlett-Packard* e *Google*, ainda apresentou uma pré-disposição para a utilização em dispositivos móveis com sistema operacional Android. Na sua página oficial, as plataformas Android e iPhone são citadas dentre as suportadas pela ferramenta (TESSERACT OCR, 2013, tradução nossa).

A ferramenta Tesseract tem como princípio básico de funcionamento receber uma imagem, submetê-la ao processo de reconhecimento dos caracteres e então retornar o texto reconhecido. Ela pode ser usada como uma instalação direta

no ambiente *desktop* em diversos sistemas operacionais, funcionando com chamadas através de linhas de comando, passando por parâmetros o caminho da imagem e do arquivo texto de saída. Pode ser usada também como biblioteca no desenvolvimento de outras aplicações, disponibilizando uma série de funcionalidades ao sistema que a utiliza.

Os primeiros testes com a ferramenta Tesseract foram realizados em ambiente *desktop*, utilizando plataforma Windows 8 e linguagem de programação Java, por meio do ambiente integrado para desenvolvimento de software *NetBeans* 7.3. Desenvolveu-se então uma aplicação a qual consistia em um simples teste de execução da ferramenta, selecionando uma imagem e submetendo-a ao processo de reconhecimento de caracteres. Tal processo trás como retorno um arquivo com o texto reconhecido. Já nas primeiras execuções o resultado foi satisfatório, tendo em vista que os textos foram devidamente reconhecidos pela ferramenta.

Esse primeiro reconhecimento da ferramenta demonstrou a possibilidade de utilizá-la de forma integrada com a linguagem Java, o que tornaria possível a criação de um *Web Service* para o reconhecimento do texto.

Após essa etapa, iniciaram-se os testes com a utilização da ferramenta como biblioteca de forma integrada, diretamente na aplicação Android. A API do Tesseract para Android pode ser encontrada diretamente no site *Google Code* no formato de projeto de biblioteca. Um projeto de biblioteca permite que outros projetos dependam dele e utilizem seus recursos. Sendo assim, um projeto de biblioteca não é instalado diretamente no dispositivo, mas sim utilizado por outras aplicações.

Esse projeto de biblioteca API faz uso dos recurso do NDK Android, conforme mencionado no subcapítulo 3.1.1.3, utilizando assim as bibliotecas em C e C++ da ferramenta Tesseract. A utilização do NDK Android requer que o código nativo seja construído através do comando *ndk-build*. É necessário também atualizar o projeto através do comando *android update project* para que este fique compatível com a versão do SDK da aplicação a qual fará utilização da biblioteca. Tal comando gera os arquivos e pastas que faltam ou que precisam ser atualizados, quando necessário, para a utilização da biblioteca. A figura 11 demonstra os comandos citados.

Figura 11 – Construção da Biblioteca com NDK

```
1 cd <caminho-do-projeto>
2 ndk-build
3 android update project --path .
4 ant release
```

Fonte: Do Autor.

Com a execução desses passos, a biblioteca já se torna disponível para ser importada e utilizada como biblioteca para reconhecimento de caracteres. A partir de então, iniciou-se o desenvolvimento de testes, tal como do protótipo do aplicativo.

5.1.2 Reconhecimento das Bibliotecas Android

Para o desenvolvimento da aplicação Android, o ambiente integrado para desenvolvimento de *software* utilizado foi o Eclipse, que conta com o *plugin Android Development Tools* (ADT), um *plugin* para desenvolvimento em Android. O dispositivo ao qual foram submetidos os testes foi o Samsung Galaxy SII, por estar disponível ao acadêmico e também por contar com uma câmera de alta definição, que ajuda a obter melhores resultados em processos de reconhecimento de caracteres.

O desenvolvimento da aplicação Android teve como etapa inicial colocar em prática a funcionalidade central da aplicação: o reconhecimento óptico de caracteres. Dessa forma o primeiro objetivo foi desenvolver uma aplicação simples que realizasse a captura da imagem e a submetesse ao OCR. Para isso foi criado um projeto de teste, contendo a biblioteca Tesseract.

5.1.2.1 Captura da Imagem

Em pesquisas realizadas sobre a captura de imagens o uso da câmera em Android, foram constatadas duas formas de realizar essa tarefa, utilizando uma captura de imagem rápida ou customizando uma câmera. Inicialmente, a caráter de teste, foi utilizada a captura de imagem rápida com uma *intent* já disponível na API Android: a *intent* do tipo *MediaStore.ACTION_IMAGE_CAPTURE*. O termo *intent* é

uma descrição abstrata que representa uma operação a ser realizada, ou seja, a própria API Android já disponibiliza uma operação pronta para a captura de imagens.

Ao utilizar a *intent* para captura de imagens, esta faz uma requisição para o sistema operacional, que utiliza uma aplicação de captura de imagem existente no dispositivo e retorna em seguida para a aplicação que a invocou, retornando a imagem capturada.

5.1.2.2 Utilizando o Tesseract

Com a *activity* de captura de imagem pronta, foi possível iniciar os trabalhos com a biblioteca Tesseract. A API do Tesseract possui uma classe *TessBaseAPI* que faz a interface com o motor OCR do Tesseract. Dessa forma, ao instanciar um objeto do tipo *TessBaseAPI*, muitos dos recursos originais do Tesseract já se tornam disponíveis.

Para realizar um reconhecimento de caracteres com a biblioteca Tesseract, se faz necessário seguir alguns passos básicos. Com a instância *TessBaseAPI* criada, é necessário chamar o método *TessBaseAPI.init()*, definindo por parâmetros o arquivo *tessdata* referente a linguagem desejada. É necessário também definir a imagem a ser reconhecida, admitindo variáveis do tipo *file* ou *bitmap*, através do método *setImage()*. A captura do texto pode ser feita através do método *getUTF8Text()*. O procedimento descrito pode ser observado no trecho de código-fonte conforme figura 12.

Figura 12 – Trecho do código fonte para utilização da Tesseract

```

//Cria um objeto Tesseract
TessBaseAPI baseApi = new TessBaseAPI();

//Faz a leitura de path ocr
baseApi.init(Environment.getExternalStorageDirectory()
    + "/external_sd/tesseract-ocr", "por");
//verifica rotação da imagem
try {
    checarOrientacao();
} catch (IOException e) {
    e.printStackTrace();
}
//Cria um novo arquivo com a imagem já rotacionada
File file = new File(_caminhoImagem);
//Define o arquivo a ser submetido ao Tesseract
baseApi.setImage(file);
//Captura o texto reconhecido
String textoReconhecido = baseApi.getUTF8Text();
//Finaliza o Tesseract
baseApi.end();

```

Fonte: Do autor.

Na figura 12 também é possível perceber a invocação do método *chacarOrientacao()*. Esse método verifica a rotação da imagem capturada e adequa a mesma para a rotação padrão do Tesseract: imagem em modo paisagem (rotação = 0).

5.1.2.3 Conversão Texto-Voz

Os dispositivos atuais que utilizam a plataforma Android possuem mecanismos nativos de saída de texto para fala. Estes mecanismos podem ser disponibilizados pela própria Google ou de outros fabricantes. As vozes desses mecanismos são muito utilizadas em aplicações de orientação, como aplicativos de *Global Positioning System* (GPS), por exemplo.

Porém, as linguagens disponibilizadas pelos mecanismos padrões do dispositivo normalmente são muito limitadas. Portanto buscou-se um mecanismo de saída de voz alternativo e o mecanismo aplicado foi o *software* proprietário SVOX *Classic TTS*, por dar suporte a língua portuguesa do Brasil.

Para conversão de texto em voz, a plataforma Android disponibiliza a

biblioteca *TextToSpeech*, que possui uma série de recursos para gerenciar os mecanismos de saída de voz do dispositivo.

Para que uma instância do objeto *TextToSpeech* possa sintetizar um texto, é necessário que este já tenha concluído sua inicialização. Dessa forma, uma classe que utiliza TTS deve implementar o método *onInit()* da interface *TextToSpeech.OnInitListener* para ser notificado quando a inicialização estiver completa. Alguns métodos também são muito importantes quanto ao gerenciamento do TTS, como o método *setLanguage()*, o qual define a linguagem a ser utilizada, o método *stop()*, que pára a leitura em execução, e o método *shutdown()*, que finaliza e libera os recursos utilizados pelo TTS.

Com o texto reconhecido pela biblioteca Tesseract, instanciou-se um objeto do tipo TTS, aplicando os métodos anteriormente descritos, o qual ficou responsável pela leitura do texto.

5.1.3 Desenvolvimento do Protótipo

Com base no aplicativo desenvolvido durante o reconhecimento das bibliotecas do Android, iniciou-se o desenvolvimento do protótipo. Baseado nos resultados obtidos nesse reconhecimento, foram elencados os passos para a criação do protótipo, iniciando uma nova etapa, onde foram melhoradas as implementações utilizadas na etapa de reconhecimento, agora aplicando conceitos de usabilidade e modo de interação com o usuário.

Alguns dos principais itens elencados foram: desenvolver uma câmera personalizada com foco automático, fornecer auxílio quanto a orientação do dispositivo e disponibilizar saídas por meio de voz.

5.1.3.1 Câmera Personalizada

Como relatado na Seção 5.1.2.1, além da utilização da *intent* de captura de imagem disponibilizada pelo Android, existe também a opção de criar uma câmera personalizada, a fim de atender necessidades específicas de uma aplicação. Isto se fez necessário para o caso deste protótipo. Para que seja possível que deficientes visuais possam fotografar um texto, é necessário que parâmetros como foco, iluminação, enquadramento, dentre outros, sejam o mais otimizados possível.

A plataforma Android disponibiliza bibliotecas de acesso a recursos dos *hardwares* disponíveis no dispositivo. No caso das câmeras, a biblioteca *android.hardware.Camera* disponibiliza tais recursos.

Para a criação de uma aplicação com câmera personalizada se faz necessário seguir alguns passos. Dentre os principais estão:

- a) detectar as câmeras disponíveis no dispositivo e então solicitar acesso a câmera desejada;
- b) criar uma classe *Preview*, a qual será responsável pela visualização ao vivo do que está sendo capturado pela câmera. Esta classe deverá estender *SurfaceView* e implementar a interface *SurfaceHolder*;
- c) criar um *layout Preview* referente a interface gráfica que fará a interação com o usuário, como a visualização e os controles disponíveis;
- d) configurar os *Listeners* aos componentes para captura das ações exercidas pelo usuário sobre a aplicação, como pressionar o botar de fotografar, por exemplo;
- e) programar os métodos para captura e armazenamento da imagem;
- f) liberar a câmera através do comando *release()*, o que irá permitir que outras aplicações façam a utilização da câmera.

A câmera personalizada do protótipo foi desenvolvida seguindo estas etapas. Todas as classes envolvidas com a activity da câmera estão organizadas no pacote *br.claiton.camera*, composto pelas classes *Autofocus*, *CameraActivity*, *CameraPreview* e *CameraTCC*.

Dentre as classes que compõem a estrutura da câmera, a classe *Autofocus* é a responsável por focar a câmera automaticamente. Nela foi implementada a interface *Camera.AutoFocusCallback* que possui o método *onAutoFocus()* que retorna quando a câmera está focada ou não. Dessa forma, de modo recursivo, com base na resposta (focado ou não focado) é definido o tempo de atraso e chamado o método *autoFocus()* novamente, tornando assim o processo de focagem da câmera automatizado. O tempo de atraso para o foco automático definido foi de três segundos, caso esteja focado, e um segundo, caso não tenha conseguido focar.

5.1.3.2 Suporte à Orientação do Dispositivo

Para que o processo de reconhecimento de caracteres tenha um bom resultado, é necessário que a imagem submetida esteja alinhada horizontal ou verticalmente, e que esteja na orientação correta. Dessa forma, visando apoiar os usuários deficientes visuais na execução dessa tarefa, buscou-se a utilização de um recurso disponível na maioria dos dispositivos móveis atualmente: o acelerômetro.

O acelerômetro é um tipo de sensor que mede a aceleração aplicada ao dispositivo, podendo assim informar a posição a qual este se encontra. Usando desse recurso, a activity da câmera pode saber quando o dispositivo se encontra alinhado ou não.

Para utilizar um sensor em Android, é necessário implementar a interface *SensorEventListener*, com os métodos *onSensorChanged()* e *onAccuracyChanged()*, responsáveis pelas ações executadas quando houverem mudanças na leitura e na precisão do sensor, respectivamente.

É necessário também obter uma instancia de objeto do tipo *SensorManager* solicitando o serviço do sistema operacional responsável pelos sensores, por meio do método *getSystemService(SENSOR_SERVICE)*. Chama-se então o método *SensorManager.registerListener()*, o qual registrará um *Listener* ao sensor para obter seus dados.

Com o gerenciador de sensores configurado, implementou-se a lógica para verificar se o dispositivo está devidamente alinhado, dentro do método *onSensorChaged()*. A partir de dois parâmetros pré-determinados, nomeados "TOLERANCIA_MINIMA" e "TOLERANCIA_MAXIMA", é verificado se aceleração exercida sobre os eixos ultrapassa um limite estabelecido de 0.5 m/s², para mais ou para menos. Esses valores são verificados tanto no eixo x quanto no eixo y, considerando que o dispositivo pode estar na orientação de retrato ou paisagem.

Com base na informação obtida pelos sensores, foram implantados dois recursos para auxiliar o usuário a perceber quando o dispositivo está alinhado, sendo um recurso visual e outro auditivo. Com tais recursos, se faz possível passar informações tanto a usuários com baixa visão, quanto a usuários cegos.

Como recurso visual, foi implantada uma borda colorida na área de visualização da imagem, podendo esta estar nas cores vermelha, quando desalinhado ou desfocado, e verde, quando alinhado e em foco. Tal recurso foi

implementado no método *signalizar()* da classe *CameraActivity*, o qual é responsável por mudar as cores de acordo com a movimentação do dispositivo, como ilustrado na figura 13.

Figura 13 – Método responsável pela troca de cor conforme alinhamento do dispositivo

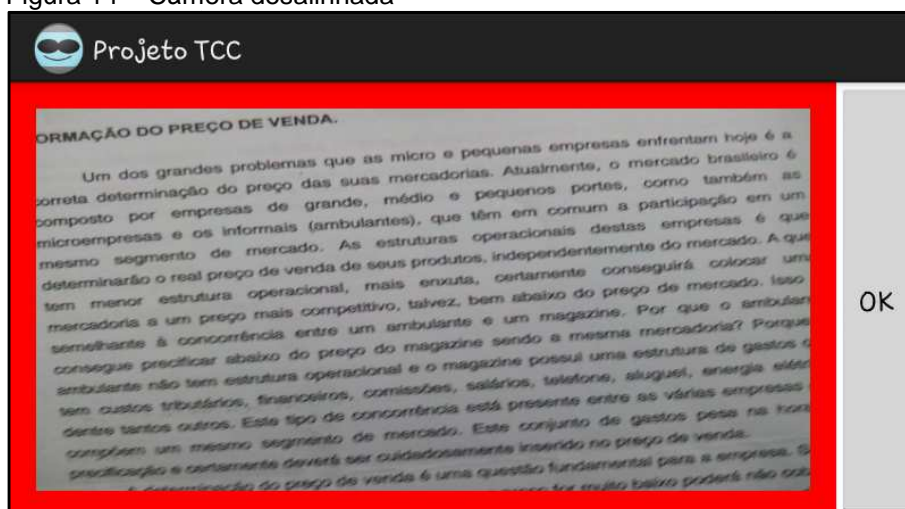
```
// Modifica as cores da borda da visualização da câmera
public void sinalizar() {
    if ((focado && retrato) || (focado && paisagem)) {
        preview.setBackgroundColor(Color.argb(255, 0, 255, 0));
    } else {
        preview.setBackgroundColor(Color.argb(255, 255, 0, 0));
    }
}
```

Fonte: Do autor.

Para recurso auditivo foi utilizado um sinal sonoro, que é emitido toda vez que o dispositivo se encontra alinhado, com o mesmo intuito de sinalizar a posição correta do dispositivo. A administração e reprodução de sons e recursos de áudio em aplicativos Android é realizada através da classe *SoundPool*. Criando um objeto do tipo *SoundPool* é possível carregar uma coleção de amostras de sons para a memória e então reproduzi-las.

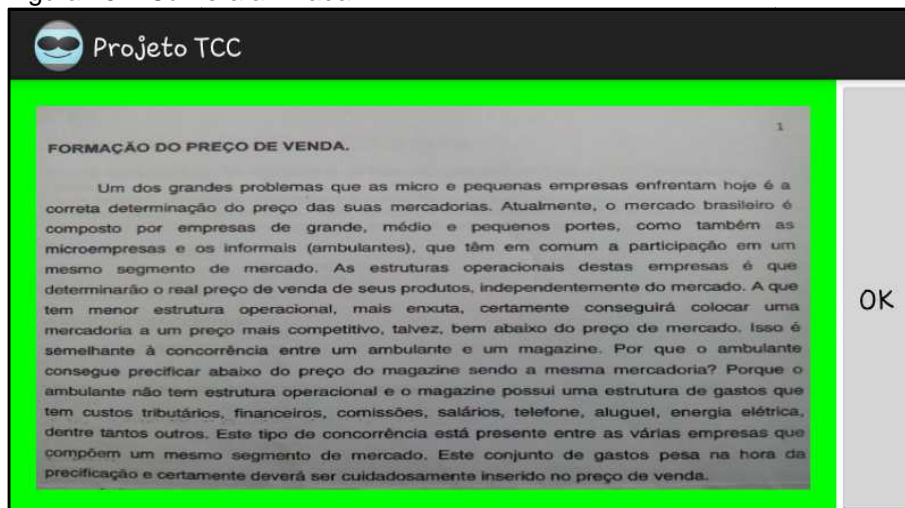
A implementação da lógica dos recursos de orientação, tanto visual quanto sonoro, se encontra no método *onSensorChanged()* da classe. Dessa forma cada vez que o dispositivo fica alinhado o aplicativo dispara as informações, conscientizando o usuário de que a imagem já pode ser capturada. A figura 14 e a figura 15 demonstram a orientação visual que ocorre nas situações onde o dispositivo se encontra desalinhado e alinhado, respectivamente.

Figura 14 – Câmera desalinhada



Fonte: Do autor.

Figura 15 – Câmera alinhada



Fonte: Do autor.

Reconhecer que o dispositivo está alinhado é uma informação muito importante, principalmente no processo do reconhecimento de caracteres, pois o resultado obtido é muito otimizado. Sendo assim, os recursos aplicados como suporte a orientação do dispositivo foram fundamentais no apoio ao processo de captura da imagem.

5.1.3.3 Saída por Meio de Voz

Fornecer instruções aos usuários deficientes visuais por meio de sintetizadores de voz é uma forma muito eficiente de tornar aplicações mais acessíveis. Dispositivos Android possuem um recurso chamado *TalkBack*, que

consiste em um leitor de telas que retorna por meio de um sintetizador de voz o texto tocado pelo usuário, ou seja, conforme o usuário toca na tela e/ou navega pelos menus, o dispositivo retorna por meio de voz a descrição do item tocado.

O *TalkBack* pode ser ativado nos dispositivos Android acessando as suas configurações no menu *acessibilidade*. Utilizando esse recurso, é possível ouvir as opções de um determinado aplicativo. Assim, esse recurso é muito eficiente sendo combinado ao aplicativo-protótipo neste trabalho elaborado.

No protótipo desenvolvido, a saída de voz ocorre por meio de objetos criados pela classe TTS. Essa classe foi elaborada baseada nos conhecimentos adquiridos no reconhecimento das bibliotecas, relatados no item 5.1.2. Nessa classe foram criados métodos otimizados para realizar a leitura de textos de forma mais prática e com diferentes finalidades, como por exemplo os métodos *falarAgora()*, que começa imediatamente uma leitura, interrompendo qualquer leitura atual, e o *aguardarEFalar()*, que fala o texto assim que terminar as leituras em execução.

5.1.4 Testes de Validação

Após a conclusão do desenvolvimento do protótipo do aplicativo, iniciaram-se os testes com intuito de avaliar o mesmo. Para isso, foram elencados dois métodos distintos de validação de dados, um de forma objetiva e outro de forma subjetiva.

Para avaliação objetiva, o protótipo foi submetido a uma bateria de testes, em diferentes cenários de utilização, a fim de medir a eficiência. Já a avaliação subjetiva foi realizada em campo, com usuários em potencial, visando avaliar a usabilidade do protótipo.

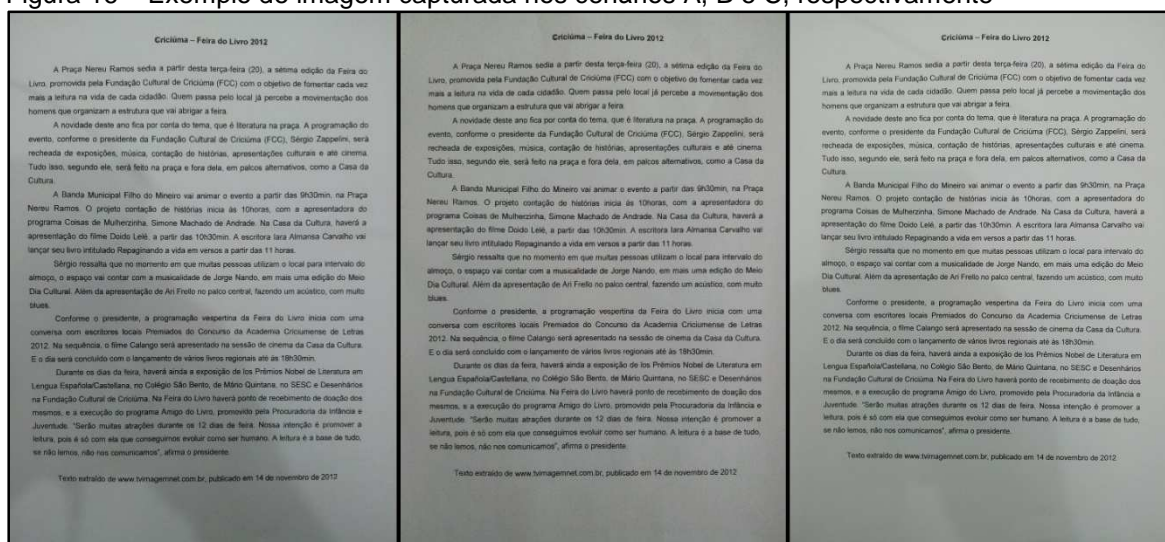
5.1.4.1 Teste de Eficiência

Durante o desenvolvimento do protótipo, em testes informais, foram percebidos fatores importantes que influenciavam diretamente no bom desempenho do reconhecimento de caracteres. Dentre os mais relevantes estão o foco e a luminosidade. Buscando amenizar os problemas gerados pela falta de foco, foi adicionada a funcionalidade de auto-foco no protótipo. Sendo assim, durante a

elaboração dos cenários utilizados para a bateria de testes, foi adicionado o fator luminosidade em diferentes condições, buscando medir sua influência sobre o processo de reconhecimento.

Os testes de eficiência se basearam em três diferentes cenários, sendo que em cada um deles havia uma iluminação ambiente diferente. No Cenário A, a fonte de luz se encontra acima do dispositivo, projetando uma sombra sobre o texto a ser escaneado. No Cenário B, a luz ambiente provém de uma fonte de posição transversal ao dispositivo, não ocasionando sombra alguma sobre o texto. E por último, no Cenário C, são utilizadas ambas as fontes de luz, projetando uma leve sombra sobre o texto, porém trazendo uma maior luminosidade ao ambiente. Um exemplo de imagem obtida em cada cenário pode ser observado na figura 16.

Figura 16 – Exemplo de imagem capturada nos cenários A, B e C, respectivamente



Fonte: Do autor.

Na execução dos testes, foram realizadas dez leituras para cada cenário e cada leitura teve seu tempo cronometrado e seus resultados gravados em imagem e texto. Cada texto reconhecido foi comparado ao texto original, para a apuração do percentual de erros ocasionados no processo. O impresso de referência utilizado durante os testes encontra-se no apêndice.

Com os resultados coletados da execução dos testes, elaborou-se a tabela 3, de forma a tabular as informações e facilitar a avaliação dos resultados.

Tabela 3 – Resultados dos testes conforme cada cenário

Teste	Cenário A		Cenário B		Cenário C	
	% Erro	Tempo Proc.	% Erro	Tempo Proc.	% Erro	Tempo Proc.
Teste 1	37,71%	00:00:38	0,73%	00:00:41	2,19%	00:00:32
Teste 2	44,04%	00:00:33	0,49%	00:00:32	21,41%	00:00:56
Teste 3	30,41%	00:00:32	2,43%	00:00:47	1,22%	00:00:27
Teste 4	37,71%	00:00:37	2,43%	00:02:45	0,97%	00:00:26
Teste 5	28,95%	00:00:34	7,30%	00:01:03	0,00%	00:00:26
Teste 6	31,87%	00:00:27	3,65%	00:00:49	1,22%	00:00:27
Teste 7	39,42%	00:00:28	2,68%	00:00:38	0,24%	00:00:27
Teste 8	31,87%	00:00:25	0,73%	00:00:30	0,00%	00:00:26
Teste 9	42,34%	00:00:40	12,17%	00:01:00	0,00%	00:00:26
Teste 10	27,74%	00:00:43	3,89%	00:00:54	0,49%	00:00:26
Média	35,21%	00:00:34	3,65%	00:00:58	2,77%	00:00:30

Fonte: Do autor.

Com base nos resultados obtidos, é possível perceber a direta influência da luminosidade para com o a eficiência do reconhecimento de caracteres. No Cenário A, avaliando cada arquivo de texto gerado, é possível perceber que os erros de reconhecimento são predominantes nas áreas onde há incidência da sombra sobre o texto, ocasionando assim uma média de 35,21% de erros contabilizados nesse cenário.

No Cenário B os resultados já são bem mais satisfatórios, apurando-se uma média de apenas 3,65% de erros. Porém o tempo de processamento da imagem foi maior em relação aos demais cenários, com tempo de duração de 58 segundos. Nos Cenários A e C, o tempo médio permaneceu na casa de 30 segundos. Sendo assim foi possível perceber que, no Cenário B, onde a iluminação é mais fraca que os demais, porém sem incidência de sombras sobre a imagem, o tempo para processamento foi maior, porém mantendo um baixo percentual de erros.

No Cenário C é possível observar um ganho considerável em termos de processamento e precisão do reconhecimento, apresentando uma média de 2,77% de erros e tempo médio de processamento de 30 segundos. Sendo assim, esse seria o cenário com a iluminação mais adequada, dentre os testados, para realizar a leitura.

5.1.4.2 Teste de Usabilidade

Para avaliar a usabilidade do protótipo, foram realizados testes com usuários em potencial, ou seja, que são ou que tem contato com deficientes visuais. Para isso, foram realizados testes junto a Associação dos Deficientes Visuais do Sul (Advisul), entidade destinada a assistência para deficientes visuais.

Inicialmente o protótipo foi apresentado aos deficientes visuais e colaboradores da entidade, com uma breve descrição dos objetivos e recursos disponibilizados. Em seguida, iniciaram-se os testes com os usuários, sendo compostos por cegos, portadores de baixa visão e com visão normal.

Os testes foram realizados de forma subjetiva, permitindo que os usuários se adaptassem ao dispositivo e aos recursos, de forma a obter gradualmente um melhor resultado com a utilização do protótipo. Como texto de referencia, foi utilizado o mesmo impresso dos testes de eficiência, relatados na Seção 5.1.4.1.

Com base na observação e em relatos dos usuários, algumas considerações foram feitas:

- a) a utilização do protótipo por parte de usuários cegos é a mais complexa, levando-se em consideração uma série de fatores necessários, como alinhamento, iluminação e desfocagem por movimento. A utilização mais satisfatória de um aplicativo desse modelo requer a disponibilização de mais recursos por parte do aplicativo e mais treinamento por parte do usuário cego;
- b) usuários com baixa visão apresentaram resultados bastante satisfatórios, tendo em vista que após algumas utilizações foi possível identificar o conteúdo do impresso escaneado;
- c) usuários não portadores de deficiência visual, mas que dão apoio a deficientes visuais, como a equipe de pedagogia e assistência social, que precisam disponibilizar conteúdo e informação a essas pessoas, tiveram total aceitação e facilidade quanto ao funcionamento e utilização do protótipo.

A fundação mostrou muito interesse e predisposição a incorporar a tecnologia a seus recursos utilizados diariamente. Relataram as dificuldades que possuem quanto ao acesso a determinadas informações, tal como apostilas e livros, e ressaltaram o quanto o protótipo pode ajudar nesse processo.

5.2 RESULTADOS OBTIDOS

Com a pesquisa realizada neste trabalho e a aplicação dos fundamentos obtidos através das pesquisas bibliográficas, foi possível obter um protótipo de aplicativo Android capaz de efetuar a leitura de impressos, utilizando a ferramenta de OCR Tesseract e a biblioteca TTS.

Através da realização de testes de eficiência, foi possível constatar que o mecanismo OCR Tesseract possui uma ótima precisão e funciona perfeitamente em conjunto com a plataforma Android. Porém foi possível detectar uma sensibilidade à iluminação e ao foco, o que é comum a qualquer motor OCR. Sendo assim, a aplicação de técnicas de pré-processamento de imagens, conforme pesquisas descritas na seção 3.2.2, pode ser uma alternativa para amenizar tais problemas gerados no processo da aquisição da imagem.

A usabilidade do protótipo foi avaliada em conjunto a uma entidade especializada em deficiência visual, trazendo mais credibilidade ao resultado da avaliação desse critério. Em testes com usuários da entidade, foi possível perceber que o protótipo possui um resultado mais favorável quanto a utilização por parte de usuários com baixa visão e de visão normal. O protótipo mostrou menor capacidade de atender usuários com cegueira total, necessitando da disponibilização de mais recursos e mais tempo de treinamento do usuário.

Sendo assim, o protótipo apresentou-se ideal para utilização em ambientes de aprendizagem, como escolas, universidades, instituições, entre outros, onde um usuário pode digitalizar de forma simples um texto impresso e então disponibilizar a informação em formato de áudio para um deficiente visual.

O protótipo apresentou também ótimos resultados com a utilização por parte de usuários com baixa visão, podendo assim ser utilizado como aplicativo pessoal. Os recursos de acessibilidade pesquisados foram muito importantes para que isso fosse possível. Funções como retorno por voz e visual para auxiliar com o alinhamento da imagem foram imprescindíveis para usuários de baixa visão.

O protótipo se mostrou uma solução de simples aplicação, fornecendo acessibilidade com baixo custo. É importante ressaltar a facilidade de acesso a tecnologia, por se tratar de um dispositivo móvel comum, com a plataforma Android e uma câmera para captura da imagem.

Sendo assim, todos os objetivos propostos foram atingidos, resultando em

um protótipo de aplicativo muito robusto, com diversas possibilidades de aprimoramento, na linha de pesquisa da acessibilidade que possibilita a aplicação de recursos tecnológicos diferentes e ainda assim muito carente de novas soluções.

6 CONCLUSÃO

A execução do presente trabalho de conclusão de curso proporcionou adquirir conhecimentos em diversas áreas. Foi possível perceber a evolução dos dispositivos móveis e o quão poderosa é a plataforma Android, tal como uma breve disposição dos recursos por ela disponibilizados. A acessibilidade foi um dos principais conceitos que direcionaram essa pesquisa, levando a aplicação das diversas outras técnicas abordadas neste trabalho.

As técnicas e tecnologias abordadas nesse trabalho, foram aplicadas buscando atender um objetivo principal: criar um protótipo de aplicativo Android para que usuários possam realizar a leitura de impressos, de forma acessível, visando auxiliar deficientes visuais no acesso a informação.

Os objetivos específicos elencados para a execução desse trabalho foram todos atingidos, os quais foram responsáveis por orientar a pesquisa e o desenvolvimento, além de possibilitar que o objetivo geral do trabalho também fosse alcançado.

Com o estudo das metodologias de acessibilidade para deficientes visuais, tanto para ambiente *desktop* quanto para dispositivos móveis, adquiriu-se embasamento teórico imprescindível para a determinação dos métodos de interação com o usuário aplicados ao protótipo.

Duas principais tecnologias estiveram diretamente ligadas ao objetivo geral, sendo elas a ferramenta de OCR e a biblioteca de conversão de texto em voz TTS.

O motor de OCR utilizado pelo protótipo foi o Tesseract, sendo que este, dentre tantos disponíveis no mercado, mostrou-se uma opção propícia para uma aplicação em dispositivos móveis, considerando que a própria documentação oficial lista o Android como plataforma suportada. Além disso, em testes de reconhecimento, a ferramenta apresentou resultados muito satisfatórios, com boa precisão em diferentes cenários.

Durante o levantamento bibliográfico, buscou-se compreender o funcionamento da biblioteca *TextToSpeech*, um recurso para conversão de texto em voz nativo da plataforma Android. A mesma se mostrou muito robusta e eficiente, disponibilizando uma série de recursos que tornam possível trazer acessibilidade ao desenvolvimento de diversas aplicações. Baseando-se em alguns testes, que

colocaram em prática os conhecimentos obtidos com a pesquisa, foi possível utilizar os recursos da biblioteca de forma integrada ao protótipo desenvolvido.

As pesquisas realizadas sobre as formas de interação com os usuários do protótipo possibilitaram que diversas técnicas fossem aplicadas a este, prevendo possíveis cenários e situações decorrentes de sua utilização, com base nas metodologias de acessibilidade para deficientes visuais. Foram também fundamentais os trabalhos correlatos nessa linha de pesquisa, agregando muito conhecimento ao trabalho desenvolvido.

Desenvolveu-se um protótipo de aplicativo utilizando a plataforma Android, o qual fez a integração entre as técnicas de reconhecimento óptico de caracteres e conversão de texto em voz, com base nos conceitos de acessibilidade e suas metodologias, sendo assim capaz de realizar a leitura de impressos para apoio a deficientes visuais.

Foi avaliada a eficiência do protótipo por meio de uma bateria de testes em diferentes cenários, que mostraram um bom desempenho no processo de reconhecimento de caracteres, com tempo médio de execução de 30 segundos e média de 2,77% de erro no melhor cenário. Os testes ainda demonstraram a sensibilidade que o processo demonstra com relação a fatores provenientes da etapa de aquisição da imagem, como iluminação e foco, os quais podem ser amenizados com aplicação de técnicas de pré-processamento da imagem, segundo teorias e práticas pesquisadas durante o levantamento bibliográfico.

Foram realizados ainda testes de usabilidade do protótipo com usuários em potencial do aplicativo em conjunto com uma entidade especializada em assistência a deficientes visuais. O protótipo apresentou ótimos resultados para com usuários portadores de baixa visão e usuários com visão normal, tendo uma usabilidade mais limitada para usuários cegos.

O protótipo se mostrou muito útil para aplicação em ambientes de ensino, instituições, ou qualquer local onde um usuário possa digitalizar o texto impresso para então disponibilizar a informação em formato de áudio para um deficiente visual. Pode também ser utilizado por deficientes visuais com baixa visão como aplicativo pessoal, para auxiliar na leitura de impressos.

Durante o processo de criação do protótipo, algumas dificuldades foram encontradas, porém compensadas com o embasamento teórico adquirido no levantamento bibliográfico realizado. O próprio desenvolvimento na plataforma

Android foi um desafio, compensado por uma excelente documentação disponível referente a plataforma. A captura da imagem também foi um desafio, em questões como o foco e a rotação da imagem, mas que conseguiram ser revertidas no próprio protótipo, aplicando recursos e conceitos disponíveis na documentação.

Como sugestão para trabalhos futuros, pode-se aplicar técnicas de pré-processamento da imagem, com intuito de otimizar os resultados do processo de reconhecimento de caracteres. Pode-se ainda atribuir diversas funcionalidades ao aplicativo, usando novas soluções de forma criativa. Alguns exemplos de funcionalidades passíveis de implementação:

- a) controle de execução da voz, semelhante a um executor de músicas por exemplo;
- b) opção de leituras agrupadas e continuação de leituras, para impressos de maior volume, como livros, apostilas, entre outros, que podem ter sua leitura interrompida e retomada posteriormente;
- c) criar opções de exportação de áudio, para gravar a leitura e repassar a diversos usuários de forma otimizada;
- d) atribuir técnicas de reconhecimento de enquadramento do texto, para identificar se a câmera esta cortando, em tempo de execução, o texto a ser escaneado;
- e) implementar técnicas de rotação automática da imagem após a captura, identificando o ângulo de rotação da imagem e corrigindo o alinhamento automaticamente.

REFERÊNCIAS

ACCESSWORLD, Afb. **An Accessible, Pricey Answer: A Review of the Mobile Phone Organizer.** Disponível em: <<http://www.afb.org/afbpress/pub.asp?DocID=aw050305>>. Acesso em: 26 set. 2012.

AI SQUARED. **Zoox Text.** Disponível em: <<http://www.aisquared.com/zoomtext/>>. Acesso em: 22 out. 2012.

ALVES, Neide Ferreira. **ESTRATÉGIAS PARA MELHORIA DO DESEMPENHO DE FERRAMENTAS COMERCIAIS DE RECONHECIMENTO ÓPTICO DE CARACTERES.** 2003. 108 f. Dissertação (Mestrado) - Curso de Engenharia Elétrica, Universidade Federal de Pernambuco, Recife, 2003.

APPLE. **Acessibilidade.** Disponível em: <<http://www.apple.com/br/accessibility/iphone/vision.html>>. Acesso em: 10 nov. 2012.

BRASIL. Comissão Especial de Acessibilidade. **Acessibilidade: passaporte para a cidadania das pessoas com deficiência:** Guia de orientações básicas para a inclusão de pessoas com deficiência. Senado Federal, 2005. 53 p.

BRASIL. **DECRETO Nº 5.296 DE 2 DE DEZEMBRO DE 2004.** Disponível em: <http://www.planalto.gov.br/ccivil_03/_Ato2004-2006/2004/Decreto/D5296.htm#art71>. Acesso em: 09 nov. 2012.

BENEDITO, João Paulo Marques Lopes. **MobileSense:** Acessibilidade para pessoas invisuais com diferentes capacidades tácteis. 2010. 96 f. Dissertação (Mestrado) - Curso de Engenharia Informática e de Computadores, Instituto Superior Tecnico, Lisboa, 2010.

BERSCH, Rita. **INTRODUÇÃO À TECNOLOGIA ASSISTIVA.** Porto Alegre: Cedi, 2008.

BURNETTE, Ed. **Hello, Android.** Dallas: The Pragmatic Bookshelf, 2010.

CAMPOS, Américo José de Oliveira. **Tecnologias GRID: serviços de apoio a Bibliotecas Digitais.** 2009. 86 f. Dissertação (Mestre) - Universidade de Aveiro, Aveiro, 2009.

CARVALHO, José Roberto; TURECK, Lucia Terezinha Zanato. **ALGUMAS REFLEXÕES SOBRE A INCLUSÃO ESCOLAR DE ALUNOS COM DEFICIÊNCIA.** Cascavel: Edunioeste, 2006.

CODE GOOGLE. **Ocropus**. Disponível em: <<http://code.google.com/p/ocropus/>>. Acesso em: 15 nov. 2012.

_____. **Tesseract ocr**. Disponível em: <<http://code.google.com/p/tesseract-ocr/>>. Acesso em: 15 nov. 2012.

CONGRESO INTERNACIONAL DE INFORMÁTICA EDUCATIVA, TISE, 16., 2011, Santiago, Chile. **Estudo de interfaces acessíveis para usuários com deficiência visual em dispositivos móveis touch screen**. Santiago, Chile: Universidad de Chile, 2011. Disponível em: <<http://lebrailletwt.greenbeans.com.br/downloads/Artigo%20TISE2011.pdf>>. Acesso em: 30 out. 2012.

DAVID H. SHEPARD, 84, DIES; OPTICAL READER INVENTOR. New York, 11 dez. 2007. Disponível em: <http://www.nytimes.com/2007/12/11/us/11shepard.html?_r=2&>. Acesso em: 28 out. 2012.

DEVELOPER ANDROID. **API Guide**. Disponível em: <<http://developer.android.com/guide/components/index.html>>. Acesso em: 12 nov. 2012.

FIGUEIRA, Emilio. **CAMINHANDO EM SILÊNCIO**: Uma introdução à trajetória das pessoas com deficiência na história do Brasil. São Paulo: Giz, 2008.

FREEDOM SCIENTIFIC. **JAWS for Windows Screen Reading Software**. Disponível em: <<http://www.freedomscientific.com/products/fs/JAWS-product-page.asp>>. Acesso em: 19 out. 2012.

_____. **MAGic Screen Magnification Software**. Disponível em: <<http://www.freedomscientific.com/products/lv/magic-bl-product-page.asp>>. Acesso em: 15 out. 2012.

_____. **The PAC Mate Omni Line**. Disponível em: <<http://www.freedomscientific.com/PDF/visionloss/flyers/PACMate-Omni-Flyer.pdf>>. Acesso em: 2 nov. 2012.

GARGENTA, Marko. **Learning Android**. Sebastopol: O'reilly, 2011. 266 p.

GNOME BRASIL. **Sobre o GNOME**. Disponível em: <<http://br.gnome.org/GNOMEBR/Gnome>>. Acesso em: 13 out. 2012.

GNU. **Ocrad - The GNU OCR**. Disponível em: <<http://www.gnu.org/software/ocrad/>>. Acesso em: 15 nov. 2012.

GOOCR. **Gocr**. Disponível em: <<http://jocr.sourceforge.net/>>. Acesso em: 15 nov. 2012.

GUERREIRO, Tiago et al. **NavTap**: um estudo de longa duração com utilizadores cegos. Porto Salvo, 2009.

GUERREIRO, Tiago. **NavTouch**: Making Touch Screens Accessible to Blind Users. Disponível em: <<http://m-accessibility.blogspot.com.br/2008/08/navtouch-making-touch-screens.html>>. Acesso em: 30 set. 2012.

INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATÍSTICA. **Tabela de Resultados Censo 2010**. Disponível em: <http://www.ibge.gov.br/home/estatistica/populacao/censo2010/resultados_preliminares_amostra/default_resultados_preliminares_amostra.shtm>. Acesso em: 20 maio 2012.

LE MOS, Edison Ribeiro; CERQUEIRA, Jonir Bechara. **O Sistema Braille no Brasil**. Disponível em: <<http://www.ibr.gov.br/?itemid=10235>>. Acesso em: 26 ago. 2012.

MARQUES FILHO, Ogê; VIEIRA NETO, Hugo. **Processamento Digital de Imagens**. Rio de Janeiro: Brasport, 1999.

MELO, Ricardo De. **10 ampliadores de telas**. Disponível em: <<http://www.movimentolivre.org/artigo.php?id=138>>. Acesso em: 21 set. 2012.

_____. **5 leitores de telas**. Disponível em: <<http://www.movimentolivre.org/artigo.php?id=50#cp>>. Acesso em: 21 set. 2012.

OPEN HANDSET ALLIANCE. **Android**. Disponível em: <http://www.openhandsetalliance.com/android_overview.html>. Acesso em: 23 out. 2012.

ORACLE. **Guia de acessibilidade do Oracle Solaris 11 para o GNOME Desktop**. Disponível em: <http://docs.oracle.com/cd/E26923_01/html/E26306/ats-2.html>. Acesso em: 17 nov. 2012.

PASSERINO, Liliana Maria; MONTARDO, Sandra Portella. Inclusão social via acessibilidade digital: Proposta de inclusão digital para Pessoas com Necessidades Especiais. **E-compós**, Pelotas, n. , p.1-18, 01 abr. 2007.

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

SILVA, Otto Marques da. **A EPOPÉIA IGNORADA: A Pessoa Deficiente na História do Mundo de Ontem e de Hoje.** São Paulo: Cedas, 1987.

SOUZA, Vanderval Borges de. **Pré-Processamento de Imagens Para o Reconhecimento Ótico de Caracteres.** 2011. 78 f. Trabalho de Conclusão de Curso (Bacharel em Ciência da Computação) – Universidade do Extremo Sul Catarinense, Criciúma.

STEELE, James; TO, Nelson. **The Android Developer's Cookbook: Building Applications with the Android SDK.** New York: Addison-Wesley Professional, 2010.

SUPPORT GOOGLE. **Sobre o TalkBack.** Disponível em: <<http://support.google.com/android/bin/answer.py?hl=pt-BR&answer=2492748>>. Acesso em: 12 nov. 2012.

TELECO. **Mercado mundial de smartphones.** Disponível em: <<http://www.teleco.com.br/smartphone.asp>>. Acesso em: 6 nov. 2012.

TOPSOFT. **SnapVision Reading Machines.** Disponível em: <<http://www.topocr.com/index.html>>. Acesso em: 15 nov. 2012.

VIRTUAL VISION. **Virtual Vision.** Disponível em: <<http://www.virtualvision.com.br/sobre.html>>. Acesso em: 22 out. 2012.

WEB, Object. **What is Middleware.** Disponível em: <<http://middleware.objectweb.org/>>. Acesso em: 22 out. 2012.

APÊNDICE(S)

APÊNDICE A – ARTIGO CIENTÍFICO

Protótipo de Aplicativo Android para Conversão de Texto de Imagens em Voz, Orientado ao Apoio à Leitura para Deficientes Visuais

Claiton de M. Marcílio¹, Fabrício Giordani², Evânio R. Nicoleit²

¹Acadêmico do Curso de Ciência da Computação – Unidade Acadêmica de Ciências, Engenharias e Tecnologias (UnaCET) – Universidade do Extremo Sul Catarinense (UNESC)
Av. Universitária, 1105 - Bairro Universitário – Criciúma – SC – Brasil

²Professor do Curso de Ciência da Computação – Unidade Acadêmica de Ciências, Engenharias e Tecnologias (UnaCET) – Universidade do Extremo Sul Catarinense (UNESC)
Av. Universitária, 1105 - Bairro Universitário – Criciúma – SC – Brasil

claiton_melo@hotmail.com, fgiordani@gmail.com, evaniorn@gmail.com

Abstract. *The visually impaired access to reading occurs mainly through language Braille and audio files, a fact that limits access to certain printed information. This limitation is due to, among other factors, the delay between original publication of information and its transcription to the accessible means (Braille / audio), and certain information that end up not being transcribed. This article describes the development of a prototype application on the Android platform that, applying the techniques of optical character recognition and text-to-audio, is able to convert the text on images captured by the camera of the device for voice format.*

Resumo. *O acesso de deficientes visuais à leitura ocorre principalmente por meio da linguagem Braille e arquivos de áudio, fato este que limita o acesso a determinadas informações impressas. Essa limitação deve-se, dentre outros fatores, ao atraso entre a publicação original de uma informação e sua transcrição ao meio acessível (Braille/áudio), além de determinadas informações que acabam não sendo transcritas. Este artigo descreve a elaboração de um protótipo de aplicativo na plataforma Android que, aplicando as técnicas de Reconhecimento Óptico de Caracteres e conversão de texto em áudio, é capaz de fazer a conversão de textos em imagens capturados pela câmera do dispositivo para o formato de voz.*

1. Introdução

A leitura tem um papel muito importante no desenvolvimento social e intelectual humano. A partir do ingresso na vida escolar, ela se torna imprescindível para o acesso ao conhecimento e a informação. Porém, muitas vezes o acesso a esse meio se torna difícil, principalmente para deficientes visuais, seja por falta de recursos tecnológicos, como livros digitalizados e arquivos de áudio, ou por falta de exemplares especializados, como livros em Braille, por exemplo.

Atualmente, há uma tendência de que os novos livros sejam digitalizados ou disponibilizados na internet. Porém, existe uma quantidade muito grande de exemplares importantes que poderão continuar disponíveis apenas em meio impresso. Existem ainda documentos e informações não formais, como textos, artigos, trabalhos, relatórios, entre outros, que também podem não estar disponíveis em meio digital.

Com a evolução das tecnologias, diversos recursos surgem buscando tornam atividades cotidianas mais fáceis, tal como promover alternativas para a resolução de problemas rotineiros. A aplicação desses recursos na promoção da acessibilidade a pessoas portadoras de deficiência recebe o nome de Tecnologia Assistiva. O objetivo dessas tecnologias é ampliar a comunicação, trazer mobilidade, possibilitar e facilitar a execução tarefas, promover o desenvolvimento de habilidades e de conhecimento (BERSCH, 2008).

A popularização e a evolução dos dispositivos móveis torna possível a elaboração de novas soluções para auxiliar a execução de tarefas, inclusive na aplicação como Tecnologia Assistiva. Recursos como Reconhecimento Óptico de Caracteres, bibliotecas de conversão de texto em áudio, captura de imagens em alta definição, entre outros, tornam possível a elaboração de inúmeros tipos diferentes de aplicações, para as mais variadas finalidades.

Este artigo descreve a utilização de técnicas de Reconhecimento Óptico de Caracteres e conversão de texto em voz para a elaboração de um protótipo de aplicativo para Android, voltado para o apoio à leitura de documentos impressos por parte de portadores de deficiências visuais. Para isso foi realizada uma fundamentação teórica referente as metodologias de acessibilidade para deficientes visuais e formas de interação, assim como das tecnologias aplicadas para o desenvolvimento do protótipo. Após a elaboração do protótipo, realizaram-se testes objetivos e subjetivos sobre este, os quais puderam comprovar a eficiência e usabilidade do mesmo.

2. Acessibilidade à Leitura para Deficientes Visuais

A sociedade atualmente possui diversos problemas na inclusão de pessoas portadoras de deficiências. Seja na falta de conscientização pública, na escassez de infraestrutura adequada ou na indisponibilidade de recursos, a dificuldade de acesso igual a todos é facilmente identificada.

Ao analisar os acervos disponíveis em diferentes bibliotecas, por exemplo, podem-se constatar diversos problemas, como escassez de exemplares transcritos em Braille ou em meio digital. O número de livros disponíveis nesses formatos é muito inferior aos disponíveis na disposição original. Além disso, o espaço de armazenagem necessário para volumes em Braille é muito grande, o que se torna um problema para o usuário e para o estabelecimento que o fornece.

Existe ainda um atraso entre a publicação original e a transcrição em Braille ou digitalização do exemplar, fato este que acaba prejudicando o leitor. Textos, artigos, documentos, trabalhos e outros impressos acabam não convertidos para uma forma acessível ao deficiente, o que agrava mais ainda mais o problema. Percebe-se também que a portabilidade de livros em Braille é comprometida, pois os livros são pesados e tem um grande volume, dificultando o transporte dos mesmos.

Em outras situações também é possível identificar problemas no acesso a informação. Em cartazes explicativos, folders, rótulos de embalagens, manuais de instrução, cardápios, listas telefônicas, correspondências, dentre outros, que dificilmente são disponibilizados em meio acessível.

Em 2010, dados do Instituto Brasileiro de Geografia e Estatística (IBGE) demonstram que o número de pessoas no Brasil que apresentam alguma deficiência visual ultrapassa os 6,5 milhões, sendo que desses mais de 528 mil declararam que não conseguem de modo algum enxergar (IBGE, 2010). Dessa forma, é possível perceber que o número de pessoas que sofrem com os problemas de falta de acessibilidade é muito grande, o que exige a elaboração de diferentes alternativas para a solução destes problemas.

Com o desenvolvimento de uma aplicação para digitalização instantânea de textos e conversão em arquivo de áudio, problemas como espaço de armazenagem, portabilidade, acesso a impressos e atraso na publicação de versões acessíveis, por exemplo, podem ser resolvidos. Com a aplicação das técnicas de reconhecimento óptico de caracteres e conversão de texto em voz, torna-se possível desenvolver uma aplicação que amplia a acessibilidade de deficientes visuais, tornando possível a leitura de livros e muitos outros meios informativos ainda no formato impresso.

3. O Protótipo de Aplicativo Android

As etapas metodológicas utilizadas durante todo o desenvolvimento foram: fundamentação teórica; exploração da ferramenta de reconhecimento óptico de caracteres Tesseract; pesquisa sobre desenvolvimento em Android e a biblioteca *TextToSpeech* para conversão de texto em voz; implementação do aplicativo e realização de testes; análise de desempenho e descrição dos resultados obtidos.

O levantamento bibliográfico teve como objetivo trazer o embasamento necessário referente aos temas que compõem a estrutura do projeto de pesquisa, como aspectos da acessibilidade e tecnologias assistivas, os sintetizadores de voz, o desenvolvimento em Android e o Reconhecimento Óptico de Caracteres.

3.1. Ferramenta OCR Tesseract

A tecnologia de *Optical Character Recognition* (OCR), no português, Reconhecimento Óptico de Caracteres, é um sistema de processamento de imagens que consiste na extração de texto a partir de imagens digitais. Estas imagens podem ser obtidas a partir de páginas impressas como livros, revistas, jornais, documentos, dentre outros, que são digitalizadas por aparelhos scanners ou máquinas digitais, tal como imagens já armazenadas em meio digital (ALVES, 2003).

O processo de OCR consiste em separar os caracteres das outras informações que estão contidas na imagem para obter somente o texto desta (SOUZA, 2011). Normalmente essas informações são compostas por figuras (gráficos, diagramas, fotos, gravuras, e outros) e texto. É importante identificar e classificar essas informações para que o OCR melhore seu desempenho, pois assim não perdera tempo tentando interpretar áreas sem informação textual (ALVES, 2003).

Atualmente existem diversas aplicações de OCR disponíveis no mercado, tanto código fonte aberto quanto código proprietário. Algumas delas se destacam por serem conhecidas e utilizadas em diversos projetos, tal como por sua precisão no processamento de seus motores OCR. A ferramenta escolhida para a realização do reconhecimento do texto foi a Tesseract, pois além de ser uma biblioteca open-source com um excelente histórico de desenvolvimento, tendo na sua lista de desenvolvedores empresas como Hewlett-Packard e Google, ainda apresentou uma pré-disposição para a utilização em dispositivos móveis com sistema operacional Android. Na sua página oficial, a plataforma Android está entre as citadas como suportadas pela ferramenta (CODE GOOGLE, 2013, tradução nossa).

A ferramenta Tesseract tem como princípio básico de funcionamento receber uma imagem, submetê-la ao processo de reconhecimento dos caracteres e então retornar o texto reconhecido. Ela pode ser usada como uma instalação direta no ambiente desktop em diversos sistemas operacionais, funcionando com chamadas através de linhas de comando, passando por parâmetros o caminho da imagem e do arquivo texto de saída. Pode ser usada também como biblioteca no desenvolvimento de outras aplicações, disponibilizando uma série de funcionalidades ao sistema que a utiliza.

Os primeiros testes com a ferramenta Tesseract foram realizados em ambiente desktop, utilizando plataforma Windows 8 e linguagem de programação Java, por meio do ambiente integrado para desenvolvimento de software NetBeans 7.3. Desenvolveu-se então uma aplicação a qual consistia em um simples teste de execução da ferramenta, selecionando uma imagem e submetendo-a ao processo de reconhecimento de caracteres. Tal processo trás como retorno um arquivo com o texto reconhecido. Já nas primeiras execuções o resultado foi satisfatório, tendo em vista que os textos foram devidamente reconhecidos pela ferramenta.

Esse primeiro reconhecimento da ferramenta demonstrou a possibilidade de utilizá-la de forma integrada com a linguagem Java, o que tornaria possível a criação de diversas soluções com a mesma, como um Web Service para o reconhecimento do texto, por exemplo.

Após essa etapa, iniciaram-se os testes com a utilização da ferramenta como biblioteca de forma integrada, diretamente na aplicação Android. A API do Tesseract para Android pode ser encontrada diretamente no site Google Code no formato de projeto de biblioteca. Um projeto de biblioteca permite que outros projetos dependam dele e utilizem seus recursos. Sendo assim, um projeto de biblioteca não é instalado diretamente no dispositivo, mas sim utilizado por outras aplicações.

Esse projeto de biblioteca API faz uso dos recurso *Native Development Toolkit* (NDK) Android, que é um conjunto de ferramentas que permitem a implementação de aplicativos utilizando código nativo, como C e C++, possibilitando a utilização de bibliotecas já escritas nessas linguagens de forma integrada ao aplicativo (DEVELOPER ANDROID, 2013, tradução nossa). Dessa forma, a API Tesseract utiliza as bibliotecas em C e C++ da ferramenta Tesseract. A utilização do NDK Android requer que o código nativo seja construído através do comando *ndk-build*. É necessário também atualizar o projeto através do comando *android update project* para que este fique compatível com a versão do SDK da aplicação a qual fará utilização da biblioteca. Tal comando gera os arquivos e pastas que faltam ou que precisam ser atualizados, quando necessário, para a utilização da biblioteca.

Com a execução desses passos, a biblioteca já se torna disponível para ser importada e utilizada como biblioteca para reconhecimento de caracteres. A partir de então, iniciou-se o desenvolvimento de testes, tal como do protótipo do aplicativo.

3.2. Reconhecimento das Bibliotecas Android

Para o desenvolvimento da aplicação Android, o ambiente integrado para desenvolvimento de software utilizado foi o Eclipse, que conta com o plugin Android Development Tools (ADT), um plugin para desenvolvimento em Android. O dispositivo ao qual foram submetidos os testes foi o Samsung Galaxy SII, por estar disponível ao acadêmico e também por contar com uma câmera de alta definição, que ajuda a obter melhores resultados em processos de reconhecimento de caracteres.

O desenvolvimento da aplicação Android teve como etapa inicial colocar em prática a funcionalidade central da aplicação: o reconhecimento óptico de caracteres. Dessa forma o primeiro objetivo foi desenvolver uma aplicação simples que realizasse a captura da imagem e a submetesse ao OCR. Para isso foi criado um projeto de teste, contendo a biblioteca Tesseract.

Em pesquisas realizadas sobre a captura de imagens o uso da câmera em Android, foram constatadas duas formas de realizar essa tarefa, utilizando uma captura de imagem rápida ou customizando uma câmera. Inicialmente, foi utilizada a captura de imagem rápida com uma *intent* já disponível na API Android: a *intent* do tipo *MediaStore.ACTION_IMAGE_CAPTURE*. O termo *intent* é uma descrição abstrata que

representa uma operação a ser realizada, ou seja, a própria API Android já disponibiliza uma operação pronta para a captura de imagens.

Com a *activity* de captura de imagem pronta, foi possível iniciar os trabalhos com a biblioteca Tesseract. A API do Tesseract possui uma classe *TessBaseAPI* que faz a interface com o motor OCR do Tesseract. Dessa forma, ao instanciar um objeto do tipo *TessBaseAPI*, muitos dos recursos originais do Tesseract já se tornam disponíveis.

Para realizar um reconhecimento de caracteres com a biblioteca Tesseract, se faz necessário seguir alguns passos básicos. Com a instância *TessBaseAPI* criada, é necessário chamar o método *TessBaseAPI.init()*, definindo por parâmetros o arquivo *tessdata* referente a linguagem desejada. É necessário também definir a imagem a ser reconhecida, admitindo variáveis do tipo *file* ou *bitmap*, através do método *setImage()*. A captura do texto pode ser feita através do método *getUTF8Text()*.

3.3. Conversão Texto-Voz

Para conversão de texto em voz, a plataforma Android disponibiliza a biblioteca *TextToSpeech*, que possui uma série de recursos para gerenciar os mecanismos de saída de voz do dispositivo.

Para que uma instância do objeto *TextToSpeech* possa sintetizar um texto, é necessário que este já tenha concluído sua inicialização. Dessa forma, uma classe que utiliza TTS deve implementar o método *onInit()* da interface *TextToSpeech.OnInitListener* para ser notificado quando a inicialização estiver completa. Alguns métodos também são muito importantes quanto ao gerenciamento do TTS, como o método *setLanguage()*, o qual define a linguagem a ser utilizada, o método *stop()*, que pára a leitura em execução, e o método *shutdown()*, que finaliza e libera os recursos utilizados pelo TTS.

Com o texto reconhecido pela biblioteca Tesseract, instanciou-se um objeto do tipo TTS, aplicando os métodos anteriormente descritos, o qual ficou responsável pela leitura do texto.

3.4. Desenvolvimento do Protótipo

Com base no aplicativo desenvolvido durante o reconhecimento das bibliotecas do Android, iniciou-se o desenvolvimento do protótipo. Baseado nos resultados obtidos nesse reconhecimento, foram elencados os passos para a criação do protótipo, iniciando uma nova etapa, onde foram melhoradas as implementações utilizadas na etapa de reconhecimento, agora aplicando conceitos de usabilidade e modo de interação com o usuário. Alguns dos principais itens elencados foram: desenvolver uma câmera personalizada com foco automático, fornecer auxílio quanto a orientação do dispositivo e disponibilizar saídas por meio de voz.

Como relatado anteriormente, além da utilização da *intent* de captura de imagem disponibilizada pelo Android, existe também a opção de criar uma câmera personalizada, a fim de atender necessidades específicas de uma aplicação. Isto se fez necessário para o caso deste protótipo, pois parâmetros como foco, iluminação, enquadramento, dentre outros, quando otimizados, facilitam a captura da imagem.

A plataforma Android disponibiliza bibliotecas de acesso a recursos dos *hardwares* disponíveis no dispositivo. No caso das câmeras, a biblioteca *android.hardware.Camera* disponibiliza tais recursos.

A câmera personalizada do protótipo foi desenvolvida seguindo os padrões determinados pela própria documentação Android. Esse padrão é composto por etapas, que

vão desde a criação da visualização da câmera, até os recursos disponibilizados por ela. Todas as classes envolvidas com a *activity* da câmera estão organizadas no pacote *br.claiton.camera*. Dentre as classes que compõem a estrutura da câmera, a classe *Autofocus* é a responsável por focar a câmera automaticamente. Nela foi implementada a interface *Camera.AutoFocusCallback* que possui o método *onAutoFocus()* que retorna quando a câmera está focada ou não. Dessa forma, de modo recursivo, é definido o tempo de atraso e chamado o método *autoFocus()* novamente, tornando assim o processo de focagem da câmera automatizado.

O resultado do processo de reconhecimento de caracteres também depende de que a imagem submetida esteja alinhada horizontal ou verticalmente, e que esteja na orientação correta. Para apoiar os usuários deficientes visuais na execução dessa tarefa, buscou-se a utilização do acelerômetro, que é um tipo de sensor que mede a aceleração aplicada ao dispositivo, podendo assim informar a posição a qual este se encontra.

Com base na informação obtida pelos sensores, foram implantados dois recursos para auxiliar o usuário a perceber quando o dispositivo está alinhado, sendo um recurso visual, através de bordas coloridas (Figura 1), e outro auditivo, por meio de sinal sonoro, podendo dessa forma passar informações tanto a usuários com baixa visão, quanto a usuários cegos.

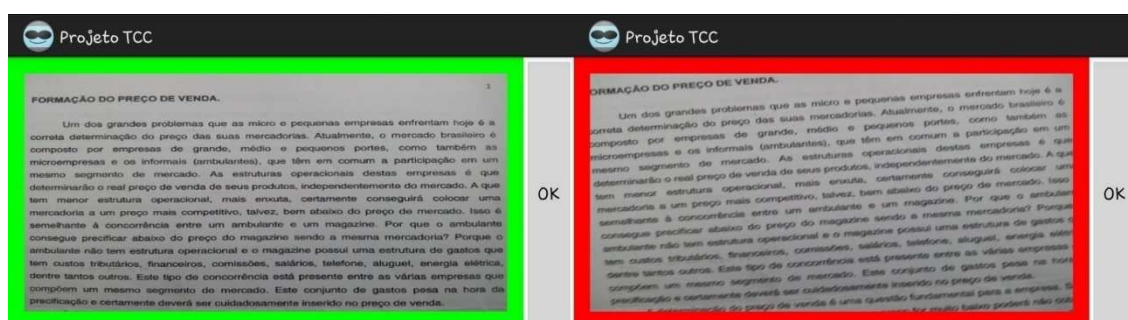


Figura 1. Alinhamento da câmera

4. Testes de Validação

Após a conclusão do desenvolvimento do protótipo do aplicativo, iniciaram-se os testes com intuito de avaliar o mesmo. Para isso, foram elencados dois métodos distintos de validação de dados, um de forma objetiva e outro de forma subjetiva.

Para avaliação objetiva, o protótipo foi submetido a uma bateria de testes, em diferentes cenários de utilização, a fim de medir a eficiência. Já a avaliação subjetiva foi realizada em campo, com usuários em potencial, avaliando a usabilidade do protótipo.

Os testes de eficiência se basearam em três diferentes cenários, sendo que em cada um deles havia uma iluminação ambiente diferente. No Cenário A, a fonte de luz se encontra acima do dispositivo, projetando uma sombra sobre o texto a ser escaneado. No Cenário B, a luz ambiente provém de uma fonte de posição transversal ao dispositivo, não ocasionando sombra alguma sobre o texto. E por último, no Cenário C, são utilizadas ambas as fontes de luz, projetando uma leve sombra sobre o texto, porém trazendo uma maior luminosidade ao ambiente. Com os resultados coletados da execução dos testes, elaborou-se a Tabela 1, de forma a tabular as informações e facilitar a avaliação dos resultados.

Tabela 1. Resultados dos testes conforme cada cenário

Teste	Iluminação 1		Iluminação 2		Iluminação 1 e 2	
	% Erro	Tempo de Proc.	% Erro	Tempo de Proc.	% Erro	Tempo de Proc.
Teste 1	37,71%	00:00:38	0,73%	00:00:41	2,19%	00:00:32
Teste 2	44,04%	00:00:33	0,49%	00:00:32	21,41%	00:00:56
Teste 3	30,41%	00:00:32	2,43%	00:00:47	1,22%	00:00:27
Teste 4	37,71%	00:00:37	2,43%	00:02:45	0,97%	00:00:26
Teste 5	28,95%	00:00:34	7,30%	00:01:03	0,00%	00:00:26
Teste 6	31,87%	00:00:27	3,65%	00:00:49	1,22%	00:00:27
Teste 7	39,42%	00:00:28	2,68%	00:00:38	0,24%	00:00:27
Teste 8	31,87%	00:00:25	0,73%	00:00:30	0,00%	00:00:26
Teste 9	42,34%	00:00:40	12,17%	00:01:00	0,00%	00:00:26
Teste 10	27,74%	00:00:43	3,89%	00:00:54	0,49%	00:00:26
Média	35,21%	00:00:34	3,65%	00:00:58	2,77%	00:00:30

Com base nos resultados obtidos, é possível perceber a direta influência da luminosidade para com o a eficiência do reconhecimento de caracteres. No Cenário A, avaliando cada arquivo de texto gerado, é possível perceber que os erros de reconhecimento são predominantes nas áreas onde há incidência da sombra sobre o texto, ocasionando assim uma média de 35,21% de erros contabilizados nesse cenário.

No Cenário C é possível observar um ganho considerável em termos de processamento e precisão do reconhecimento, apresentando uma média de 2,77% de erros e tempo médio de processamento de 30 segundos. Sendo assim, esse seria o cenário com a iluminação mais adequada, dentre os testados, para realizar a leitura.

Para avaliar a usabilidade do protótipo, foram realizados testes com usuários em potencial, ou seja, que são ou que tem contato com deficientes visuais. Para isso, foram realizados testes junto a Associação dos Deficientes Visuais do Sul (Advisul), entidade destinada a assistência para deficientes visuais.

Inicialmente o protótipo foi apresentado aos deficientes visuais e colaboradores da entidade, com uma breve descrição dos objetivos e recursos disponibilizados. Em seguida, iniciaram-se os testes com os usuários, sendo compostos por cegos, portadores de baixa visão e com visão normal.

Com base na observação e em relatos dos usuários, algumas considerações foram feitas. Inicialmente, a utilização do protótipo por parte de usuários cegos é a mais complexa, levando-se em consideração uma série de fatores necessários, como alinhamento, iluminação e desfocagem por movimento. Usuários com baixa visão apresentaram resultados bastante satisfatórios, tendo em vista que após algumas utilizações foi possível identificar o conteúdo do impresso escaneado. Usuários não portadores de deficiência visual, mas que dão apoio a deficientes visuais, como a equipe de pedagogia e assistência social, que precisam disponibilizar conteúdo e informação a essas pessoas, tiveram total aceitação e facilidade quanto ao funcionamento e utilização do protótipo.

A fundação mostrou muito interesse e predisposição a incorporar a tecnologia a seus recursos utilizados diariamente. Relataram as dificuldades que possuem quanto ao acesso a determinadas informações, tal como apostilas e livros, e ressaltaram o quanto o protótipo pode ajudar nesse processo.

5. Conclusão

A execução desta pesquisa proporcionou adquirir conhecimentos em diversas áreas. Foi possível perceber a evolução dos dispositivos móveis e o quão poderosa é a plataforma Android, tal como uma breve disposição dos recursos por ela disponibilizados. A acessibilidade foi um dos principais conceitos que direcionaram essa pesquisa, levando a aplicação das diversas outras técnicas abordadas neste trabalho.

As técnicas e tecnologias abordadas nesse trabalho, foram aplicadas buscando atender um objetivo principal: criar um protótipo de aplicativo Android para que usuários possam realizar a leitura de impressos, de forma acessível, visando auxiliar deficientes visuais no acesso a informação.

Os objetivos específicos elencados para a execução desse trabalho foram todos atingidos, os quais foram responsáveis por orientar a pesquisa e o desenvolvimento, além de possibilitar que o objetivo geral também fosse alcançado.

Com o estudo das metodologias de acessibilidade para deficientes visuais, tanto para ambiente *desktop* quanto para dispositivos móveis, adquiriu-se embasamento teórico imprescindível para a determinação dos métodos de interação com o usuário aplicados ao protótipo.

Duas principais tecnologias estiveram diretamente ligadas ao objetivo geral, sendo elas a ferramenta de OCR e a biblioteca de conversão de texto em voz TTS.

O motor de OCR utilizado pelo protótipo foi o Tesseract, sendo que este, dentre tantos disponíveis no mercado, mostrou-se uma opção propícia para uma aplicação em dispositivos móveis, considerando que a própria documentação oficial lista o Android como plataforma suportada. Além disso, em testes de reconhecimento, a ferramenta apresentou resultados muito satisfatórios, com boa precisão em diferentes cenários.

Durante o levantamento bibliográfico, buscou-se compreender o funcionamento da biblioteca *TextToSpeech*, um recurso para conversão de texto em voz nativo da plataforma Android. A mesma se mostrou muito robusta e eficiente, disponibilizando uma série de recursos que tornam possível trazer acessibilidade ao desenvolvimento de diversas aplicações. Baseando-se em alguns testes, que colocaram em prática os conhecimentos obtidos com a pesquisa, foi possível utilizar os recursos da biblioteca de forma integrada ao protótipo desenvolvido.

As pesquisas realizadas sobre as formas de interação com os usuários do protótipo possibilitaram que diversas técnicas fossem aplicadas a este, prevendo possíveis cenários e situações decorrentes de sua utilização, com base nas metodologias de acessibilidade para deficientes visuais. Foram também fundamentais os trabalhos correlatos nessa linha de pesquisa, agregando muito conhecimento ao trabalho desenvolvido.

Desenvolveu-se um protótipo de aplicativo utilizando a plataforma Android, o qual fez a integração entre as técnicas de reconhecimento óptico de caracteres e conversão de texto em voz, com base nos conceitos de acessibilidade e suas metodologias, sendo assim capaz de realizar a leitura de impressos para apoio a deficientes visuais.

Foi avaliada a eficiência do protótipo por meio de uma bateria de testes em diferentes cenários, que mostraram um bom desempenho no processo de reconhecimento de caracteres, com tempo médio de execução de 30 segundos e média de 2,77% de erro no melhor cenário. Os testes ainda demonstraram a sensibilidade que o processo demonstra com relação a fatores provenientes da etapa de aquisição da imagem, como iluminação e foco, os quais podem ser

amenizados com aplicação de técnicas de pré-processamento da imagem, segundo teorias e práticas pesquisadas durante o levantamento bibliográfico.

Foram realizados ainda testes de usabilidade do protótipo com usuários em potencial do aplicativo em conjunto com uma entidade especializada em assistência a deficientes visuais. O protótipo apresentou ótimos resultados para com usuários portadores de baixa visão e usuários com visão normal, tendo uma usabilidade mais limitada para usuários cegos.

O protótipo se mostrou muito útil para aplicação em ambientes de ensino, instituições, ou qualquer local onde um usuário possa digitalizar o texto impresso para então disponibilizar a informação em formato de áudio para um deficiente visual. Pode também ser utilizado por deficientes visuais com baixa visão como aplicativo pessoal, para auxiliar na leitura de impressos.

Durante o processo de criação do protótipo, algumas dificuldades foram encontradas, porém compensadas com o embasamento teórico adquirido no levantamento bibliográfico realizado. O próprio desenvolvimento na plataforma Android foi um desafio, compensado por uma excelente documentação disponível referente a plataforma. A captura da imagem também foi um desafio, em questões como o foco e a rotação da imagem, mas que conseguiram ser revertidas no próprio protótipo, aplicando recursos e conceitos disponíveis na documentação.

Referências

- Alves, Neide Ferreira. Estratégias Para Melhoria do Desempenho de Ferramentas Comerciais de Reconhecimento Óptico de Caracteres. 2003. 108 F. Dissertação (Mestrado) - Curso de Engenharia Elétrica, Universidade Federal de Pernambuco, Recife, 2003.
- Bersch, Rita. Introdução À Tecnologia Assistiva. Porto Alegre: Cedi, 2008.
- Code Google. Tesseract Ocr. Disponível Em: <<http://code.google.com/p/tesseract-ocr/>>. Acesso Em: 15 Nov. 2012.
- Developer Android. Api Guide. Disponível em: <<http://developer.android.com/guide/components/index.html>>. Acesso em: 12 Nov. 2012.
- Instituto Brasileiro de Geografia e Estatística. Tabela de Resultados Censo 2010. Disponível Em: <http://www.ibge.gov.br/home/estatistica/populacao/censo2010/resultados_preliminares_a_mostra/default_resultados_preliminares_amostra.shtm>. Acesso em: 20 Maio 2012.
- Souza, Vanderval Borges de. Pré-Processamento de Imagens para o Reconhecimento Óptico de Caracteres. 2011. 78 F. Trabalho de Conclusão de Curso (Bacharel em Ciência da Computação) – Universidade do Extremo Sul Catarinense, Criciúma.

APÊNDICE B – TEXTO DE REFERÊNCIA

Criciúma – Feira do Livro 2012

A Praça Nereu Ramos sedia a partir desta terça-feira (20), a sétima edição da Feira do Livro, promovida pela Fundação Cultural de Criciúma (FCC) com o objetivo de fomentar cada vez mais a leitura na vida de cada cidadão. Quem passa pelo local já percebe a movimentação dos homens que organizam a estrutura que vai abrigar a feira.

A novidade deste ano fica por conta do tema, que é literatura na praça. A programação do evento, conforme o presidente da Fundação Cultural de Criciúma (FCC), Sérgio Zappellini, será recheada de exposições, música, contação de histórias, apresentações culturais e até cinema. Tudo isso, segundo ele, será feito na praça e fora dela, em palcos alternativos, como a Casa da Cultura.

A Banda Municipal Filho do Mineiro vai animar o evento a partir das 9h30min, na Praça Nereu Ramos. O projeto contação de histórias inicia às 10horas, com a apresentadora do programa Coisas de Mulherzinha, Simone Machado de Andrade. Na Casa da Cultura, haverá a apresentação do filme Doido Lelé, a partir das 10h30min. A escritora Iara Almansa Carvalho vai lançar seu livro intitulado Repaginando a vida em versos a partir das 11 horas.

Sérgio ressalta que no momento em que muitas pessoas utilizam o local para intervalo do almoço, o espaço vai contar com a musicalidade de Jorge Nando, em mais uma edição do Meio Dia Cultural. Além da apresentação de Ari Frello no palco central, fazendo um acústico, com muito blues.

Conforme o presidente, a programação vespertina da Feira do Livro inicia com uma conversa com escritores locais Premiados do Concurso da Academia Criciumense de Letras 2012. Na sequência, o filme Calango será apresentado na sessão de cinema da Casa da Cultura. E o dia será concluído com o lançamento de vários livros regionais até às 18h30min.

Durante os dias da feira, haverá ainda a exposição de los Prêmios Nobel de Literatura em Lengua Española/Castellana, no Colégio São Bento, de Mário Quintana, no SESC e Desenhários na Fundação Cultural de Criciúma. Na Feira do Livro haverá ponto de recebimento de doação dos mesmos, e a execução do programa Amigo do Livro, promovido pela Procuradoria da Infância e Juventude. “Serão muitas atrações durante os 12 dias de feira. Nossa intenção é promover a leitura, pois é só com ela que conseguimos evoluir como ser humano. A leitura é a base de tudo, se não lemos, não nos comunicamos”, afirma o presidente.

Texto extraído de www.tvimagemnet.com.br, publicado em 14 de novembro de 2012