

Visão auxiliar: aplicação de machine learning no auxílio a visão**Auxiliary vision: machine learning application in vision aid**

DOI:10.34117/bjdv5n8-072

Recebimento dos originais: 14/07/2019

Aceitação para publicação: 21/08/2019

Lucas Oliveira de Jesus

Graduado em Análise e Desenvolvimento de Sistemas pela Fatec Arthur de Azevedo
Instituição: Fatec Arthur de Azevedo de Mogi Mirim
Endereço: Rua Ariovaldo Silveira Franco, 567 - Jd. 31 de Março, CEP 13801-005 - Mogi
Mirim(SP) - Brasil
E-mail: lucasc10f@gmail.com

Derik Teruo Barbosa Hito

Graduado em Análise e Desenvolvimento de Sistemas pela Fatec Arthur de Azevedo
Instituição: Fatec Arthur de Azevedo de Mogi Mirim
Endereço: Rua Ariovaldo Silveira Franco, 567 - Jd. 31 de Março, CEP 13801-005 - Mogi
Mirim(SP) - Brasil
E-mail: derikteruo@gmail.com

Giovani Kelvin da Rocha

Graduado em Análise e Desenvolvimento de Sistemas pela Fatec Arthur de Azevedo
Instituição: Fatec Arthur de Azevedo de Mogi Mirim
Endereço: Rua Ariovaldo Silveira Franco, 567 - Jd. 31 de Março, CEP 13801-005 - Mogi
Mirim(SP) - Brasil
E-mail: giovani_kelvin@hotmail.com

Marcio Rodrigues Sabino

Mestre em Matemática Aplicada pela Universidade de Campinas, UNICAMP
Instituição: Fatec Arthur de Azevedo de Mogi Mirim
Endereço: Rua Ariovaldo Silveira Franco, 567 - Jd. 31 de Março, CEP 13801-005 - Mogi
Mirim(SP) - Brasil
E-mail: marcio.sabino@fatecmm.edu.br

RESUMO

Segundo a Organização Mundial de Saúde (OMS), existia no ano de 2016 no mundo, aproximadamente 45 milhões de pessoas cegas e outros 135 milhões sofrendo limitações severas de visão. Neste mesmo ano, segundo o IBGE, existiam no Brasil por volta de 6,5 milhões de pessoas com deficiência visual (SBO, 2018). acessibilidade é uma característica indispensável do ambiente que assegura a melhoria da qualidade de vida das pessoas, ocasionando resultados sociais positivos e colaborando para o desenvolvimento inclusivo e sustentável. Espera-se que esta esteja presente nos espaços e serviços públicos e particulares, no transporte, na educação, nos sistemas e tecnologias da informação e de comunicação, estando estes presentes na cidade ou no campo (SDH/PR, 2018). Ante o exposto, utilizando-se de tecnologias como *machine learning* e por meio de Interface de Programação de Aplicativos (APIs), este projeto propõe o desenvolvimento de um aplicativo *mobile* para

peças que possuem deficiência visual. O aplicativo consiste em transcrever imagens em textos, e, após essa transcrição, converter estes textos em áudio, fornecendo assim, a informação que a pessoa necessita. Complementando a acessibilidade, foi proposto um formato padrão para contas de consumo às empresas. Estas devem possuir um QRCode com as informações da conta. Desta maneira, o usuário poderá ter acesso ao conteúdo de sua conta utilizando a leitura de QRCode do aplicativo. Assim sendo, este projeto fornece um recurso de acessibilidade minimizando algumas dificuldades vividas por deficientes visuais.

Palavras-chaves: Aprendizado de máquina, aplicativo, acessibilidade, tecnologia assistiva.

ABSTRACT

According to World Health Organization (WHO), in the year of 2016 there were 45 million of blind people in the world and 135 million that had been with severe vision limitations. In the same year, there were in Brazil about 6,5 million people with visual impairment (SBO, 2018). The accessibility is an indispensable characteristic in an environment that assure the improvement on people's life quality, create positive social results and it helps for an inclusive and sustainable development. Expects that it would be present in public and private spaces and services, like transport, education, systems and in the information and communication technology, being present in urban or rural areas (SDH/PR, 2018). Using technologies like machine learning, and through Application Programming Interfaces (API), this project proposes the development of a mobile application for people that has visual impairment. The application consists in transcribe images to text, and after this, it converts the text to audio, providing the information needed. Complementing the accessibility, was proposed for companies, a standard format for utility bills. These would have a QRCode with the bill information. In this way, the user can have access to the contents of his bill using the QRCode reading of the application. Thus, this project provides a resource of accessibility minimizing some difficulties experienced by visually impaired people.

Keywords: Machine learning, app, accessibility, assistive technology.

1. INTRODUÇÃO

O universo das imagens divide-se no domínio de representações visuais de objetos materiais que representam o nosso meio ambiente visual e no domínio imaterial nas quais estas surgem como imaginações, fantasias, visões, modelos, ou seja, como representações mentais (JOLY, 2007; NÖTH & SANTAELLA, 2015).

É da natureza do ser humano fazer o reconhecimento de imagens por meio de padrões, sendo possível notar esse processo desde muito cedo, como no caso de um bebê que identifica a sua mãe (CARLO et All, 2012). Em alguns campos da neurociência, estuda-se como o cérebro reconhece e interpreta objetos. Segundo Charles E. Connor, diretor do Zanvyl Krieger Mind/Brain Institute da Universidade Johns Hopkins, “A visão não acontece no olho, ocorre em fases de processamento múltiplo no cérebro.” (ROSEN, 2018; NMLS, 2018).

Com o constante avanço tecnológico, grandes centros de pesquisas estão utilizando o machine learning, para ensinar as máquinas a visualizar, reconhecer, interpretar e tomar

decisões. Mapeando-se algumas dentre as muitas necessidades enfrentadas por pessoas com deficiência visual, pode-se citar a dificuldade para realizar-se tarefas simples como reconhecer cenários comuns, desviar de obstáculos e fazer leituras básicas. Como a tecnologia machine learning pode contribuir para minimizar tais dificuldades? Como possibilitar um outro tipo de “visão” às pessoas com dificuldades visuais?

O objetivo principal deste trabalho é de prototipar um aplicativo mobile no sistema operacional Android que identifique padrões de letras e palavras, informando por meio de áudio, aquilo que está sendo visualizado por meio de uma conversão de texto em voz. Além disso, será proposta uma nova forma padrão de emissão de contas pessoais, como de água e luz, cuja sua leitura seja facilitada através do aplicativo. Para isso, será realizada a integração entre tecnologia de reconhecimento de imagens com base na biblioteca Google Cloud Vision, a qual é disponibilizada de maneira gratuita com base em sua utilização, podendo ser aplicada para o uso acadêmico e comercial, para o desenvolvimento de aplicativos na área de visão computacional.

Este trabalho será composto por 4 seções. Na primeira seção, é apresentada a introdução, proporcionando ao leitor uma ideia das necessidades, objetivos e expectativas da pesquisa. Prosseguindo, apresenta-se o Referencial Teórico, no qual são apresentadas ferramentas que proporcionam uma fundamentação e validação da pesquisa. Na sequência apresenta-se a Análise e Desenvolvimento o qual dissertará sobre a modelagem, requisitos, materiais, métodos e análises dos resultados obtidos. Em seguida, são salientados os resultados obtidos buscando responder os questionamentos iniciais, validar as hipóteses propostas, cumprimento dos objetivos, mostrar as dificuldades encontradas e propor trabalhos futuros. Por fim, apresenta-se a Bibliografia utilizada para o desenvolvimento e pesquisa.

2. FUNDAMENTAÇÃO TEÓRICA

Esta seção abordará todas as tecnologias, processos e procedimentos que fazem parte do desenvolvimento e elaboração do projeto. O projeto terá como base obras de autores especialistas em suas áreas, afim de que todo o conteúdo presente no trabalho tenha um embasamento robusto.

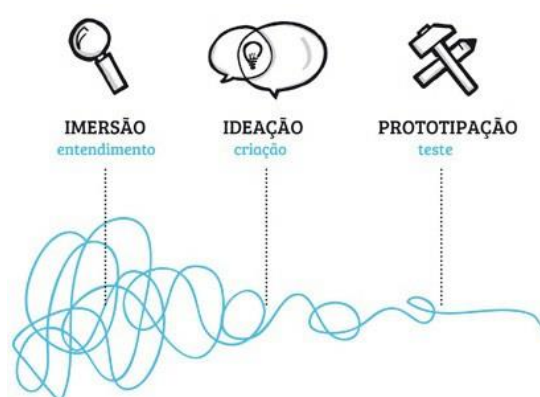
2.1 DESIGN THINKING

O Design Thinking surge da ideia de utilizar os mesmos pensamentos e percepções de um designer.

O designer, segundo Viana et. al. (2012), tem a percepção de que problemas possuem naturezas diversas e devem ser lidados conforme o ambiente, contexto, experiências pessoais e os processos que ocorrem na vida dos indivíduos. Mapeando estes pontos, é possível identificar as barreiras e dificuldades, e, tendo esse conhecimento, é possível ser mais assertivo na solução do cenário.

Segundo Viana et. al (2012), os processo não lineares do Design Thinking podem ser resumidos em: Imersão, Ideação e Prototipação, como apresentado no esquema proposto na figura 1.

Figura 1- Estrutura do Processo Design Thinking.



Fonte: (MODIFICADO SEBRAE, 2017).

2.1.1 Imersão

O processo de imersão consiste em aproximar-se do ponto de vista do cliente/usuário, podendo ser dividido em imersão preliminar e imersão profunda.

Imersão preliminar

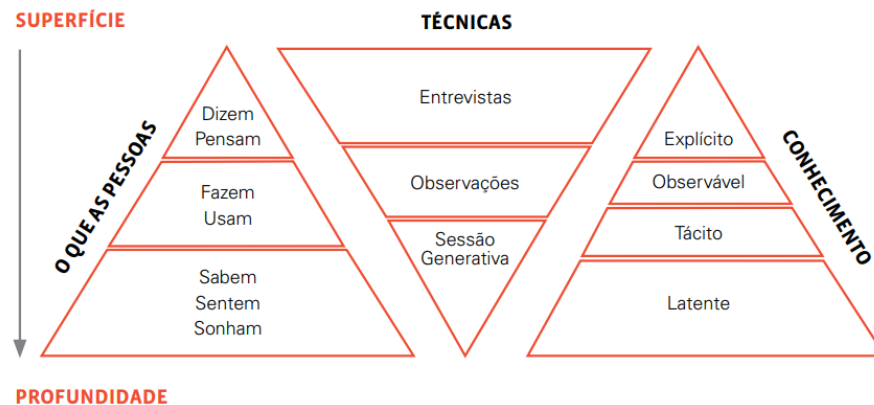
A imersão preliminar

Para Viana et. al. (2012 p. 22): “[...] tem como finalidade definir o escopo do projeto e suas fronteiras, além de identificar os perfis de usuários e outros atores-chave que deverão ser abordados.”

Imersão em profundidade

Esta imersão consiste em mergulhar no ambiente do usuário/cliente, identificando suas necessidades e expectativas. Segundo Viana et. al (2012), nesta etapa é possível utilizar técnicas como entrevistas e observação, para mergulhar no contexto do que está sendo explorado no projeto.

Figura 2 – Imersões



Fonte: Viana et. al. (2012)

A figura 2 ilustra na forma de diagramas o que ocorre na fase de imersão, quais as técnicas possíveis de serem utilizadas e o conhecimento obtido.

Análise e síntese

Após a conclusão das duas etapas anteriores, toda a massa de dados e informações são organizadas afim de gerarem padrões para auxiliarem na compreensão do problema.

2.1.2 Ideação

Nesta fase, a criatividade é o ponto chave para desenvolver ideias e soluções para os problemas no contexto apresentado. Segundo Viana et.al. (2012), também são utilizadas as ferramentas do processo de análise para ter sempre a síntese dos dados.

Para Viana et. al. (2012 p. 101): “A fase de ideação geralmente se inicia com a equipe de projeto realizando *Brainstormings* (uma das técnicas de geração de ideias mais conhecidas) ao redor do tema a ser explorado e com base nas ferramentas.”

Brainstorming

O brainstorming, geralmente executado em grupo, consiste na geração do maior número de ideias no menor tempo possível. Os participantes ficam à vontade, mas evitando perder o foco do contexto do problema a ser solucionado.

Segundo Viana et. al. (2012 p. 101): “A qualidade e a assertividade das ideias geradas se atinge através da quantidade. Quanto maior a quantia de ideias geradas pela equipe, maior é a chance de produzir uma solução inovadora e funcional.”

Ideias ousadas são bem-vindas, pois podem levar a soluções inovadoras. Críticas a essas ideias, não devem atrapalhar este processo. Dessa maneira, a avaliação fica para um momento posterior.

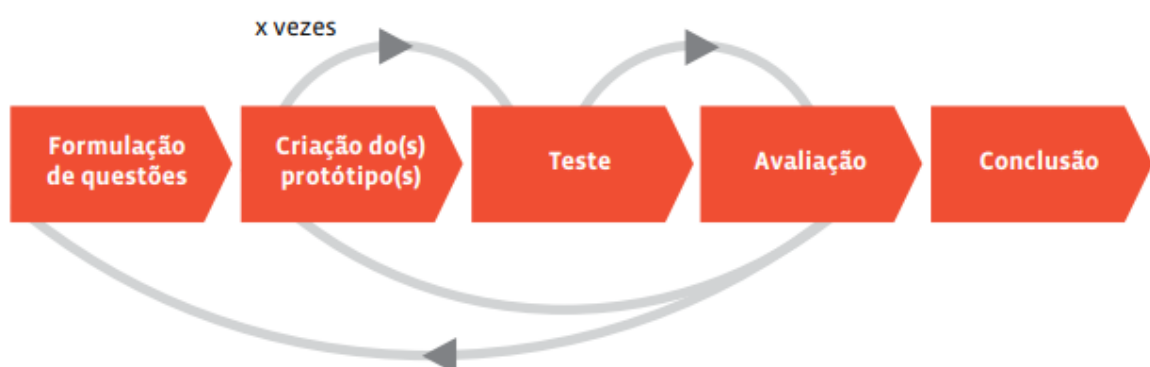
2.1.3 Prototipação

No processo de prototipação, todas as ideias, outrora abstratas, agora são representadas fisicamente. Essa representatividade, se dá sob dois aspectos:

- **Da ótica da equipe de projeto:** a equipe dá mais detalhes a forma física da ideia, aumentando sua fidelidade conforme o processo avança.
- **Do ponto de vista do usuário:** com bases em suas interações, os usuários podem fornecer avaliações e ideias para o avanço do protótipo.

Para Viana et. al. (2012 p. 124): “Protótipos reduzem as incertezas do projeto, pois são uma forma ágil de abandonar alternativas que não são bem recebidas e, portanto, auxiliam na identificação de uma solução final mais assertiva.”

Figura 3 - Prototipação



Fonte: Viana et. al. (2012)

Esses protótipos podem variar desde uma simples representação da ideia em uma folha de papel (Baixo nível), até algo que seja representado de maneira que chegue próximo da solução final (Alto Nível).

2.2 MACHINE LEARNING

Turing (1950) levantou o seguinte questionamento: “Poderiam as máquinas pensar?”.

Para responder a esse questionamento, primeiramente necessitamos entender o que é “pensar”. Quando pesquisamos nos dicionários, temos as definições: “Submeter (algo) ao processo de raciocínio lógico; exercer a capacidade de julgamento, dedução ou concepção”, “determinar pela reflexão”, “procurar lembrar-se, imaginar”, entre outras. Harnad (2008) propõe que o questionamento seja substituído, afim de termos a seguinte pergunta: “Poderiam as máquinas serem capazes de fazer o que nós, como entidades pensantes, somos capazes de fazer? E, se sim, como?”. Seguindo essa linha de raciocínio, temos a tecnologia Machine Learning.

Machine learning deriva do termo “Learning Machines” citado por Turing (1950), como máquinas que seriam capazes de aprender de maneira autômata, e foi cunhado por Samuel (1959).

Em outras palavras, podemos dizer que Machine Learning é o desenvolvimento de algoritmos com a capacidade de “aprender” sozinhos.

Mitchell (1997 p. 2, tradução nossa) define Machine Learning da seguinte maneira: “Pode-se dizer que um programa de computador aprende pela experiência, respeitando uma classe de tarefas T e uma medida de performance P, se sua performance nas tarefas T, medidas por P, aumenta por meio da experiência E”. Em outras palavras, Mitchell (1997) exemplifica que um programa de computador que é desenvolvido para aprender a jogar damas melhorará sua performance ao ser medido por sua capacidade de vencer nas tarefas que envolvem o jogo, por meio da experiência obtida jogando contra si.

Sendo assim, temos T, como a tarefa: “Jogar damas”; P, como uma medida de performance para a porcentagem de jogos ganhos; e E, como a experiência obtida com as partidas jogadas contra si. A cada nova partida, mais dados serão coletados para o aprendizado da máquina. Esses dados serão processados e irão prever jogadas que podem ser feitas. De maneira que quando a máquina jogar contra um humano, terá a informação de todas as possibilidades de jogadas existentes, conseguindo prever movimentos e estratégias.

Com a evolução da tecnologia, o Machine Learning passou a ser mais difundido e utilizado, muitas vezes para reconhecimentos de padrões e aplicações para inteligência artificial.

Dentre as tecnologias que utilizam o Machine Learning, podemos citar:

- Reconhecimento de imagens;
- Reconhecimento de voz;
- Reconhecimento facial;
- Aplicações para automatizar filtros de e-mail;
- Selecionar anúncios mais relevantes aos usuários;
- Informar as melhores rotas para desviar do trânsito;
- Recomendar filmes e músicas com base no gosto pessoal dos usuários;

entre diversas outras.

2.3 TIPOS DE MACHINE LEARNING

Os tipos de machine learning podem ser classificados como:

- Supervisionado: Consiste apresentar a máquina dados de exemplo com respostas, afim de que possam ser mapeadas as entradas e saídas;
- Semi-Supervisionado: São apresentados dados incompletos a máquina;
- Aprendizado por reforço: Os dados são apresentados apenas como um retorno as ações da máquina em um ambiente dinâmico;
- Não supervisionado: A máquina fica responsável por agrupar e rotular os dados de maneira a encontrar padrões sem interferência externa.

2.4 INTERNET DAS COISAS

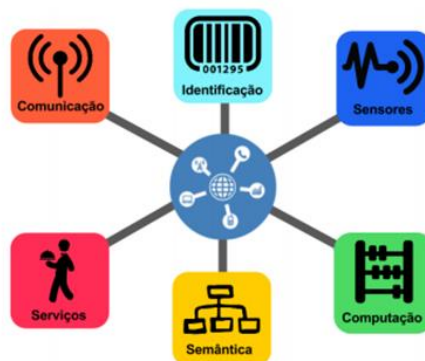
O conceito da Internet das Coisas (Internet of Things ou IoT) refere-se a um ambiente que engloba e conecta informações de diversos dispositivos como por exemplo smartphones, óculos, relógios, fones de ouvido, consoles de jogos, webcams, computadores, veículos, luzes da casa, e aplicações como compras, reabastecimento, avisos, dentre outros. Neste recente cenário, a variedade é crescente e conjectura-se que mais de 40 bilhões de dispositivos estarão conectados até 2020 (FORBES, 2014).

Além da conectividade de rede com ou sem fio, propõe-se como unidade básica de hardware, à concepção IoT, ao menos uma das seguintes características (LOUREIRO et al., 2003):

- unidade(s) de processamento;
- unidade(s) de memória;
- unidade(s) de comunicação e;
- unidade(s) de sensor(es) ou atuador(es).

A Figura 4 apresenta uma forma esquemática para a essência do IoT.

Figura 4 - Blocos básicos da essência IoT.



Fonte: Santos et Al (2016).

Segundo Goussevskiaia (2017), os blocos apresentados na Figura 1 podem ser resumidos em:

- **Identificação:** forma de identificar os objetos unicamente para conectá-los à Internet. Tecnologias como RFID, NFC (Near Field Communication) e endereçamento IP podem ser empregados para identificar os objetos.
- **Sensores/Atuadores:** os sensores possuem a função de coletar informações e, em seguida, encaminhar ou armazenar esses dados para data warehouse, clouds ou centros de armazenamento. Atuadores podem manipular o ambiente ou reagir de acordo com os dados lidos.
- **Comunicação:** refere-se às muitas técnicas utilizadas para conectar objetos inteligentes, além da função da atenção no consumo de energia dos objetos. Algumas das tecnologias usadas são WiFi, Bluetooth, IEEE 802.15.4 e RFID.
- **Computação:** abarca unidades de processamento como microcontroladores, processadores e FPGAs, responsáveis por executar algoritmos locais nos objetos inteligentes.
- **Serviços:** a IoT pode atender inúmeras classes de serviços, como por exemplo: Serviços de Identificação (temperatura, imagem, coordenadas geográficas, dentre

outros); Serviços de Agregação de Dados que coletam e resumem dados homogêneos/heterogêneos obtidos dos objetos inteligentes; Serviços de Colaboração e Inteligência que agem sobre os serviços de agregação de dados para tomar decisões e reagir de modo adequado a um determinado cenário; e Serviços de Ubiquidade que visam prover serviços de colaboração e inteligência em qualquer momento e qualquer lugar em que eles sejam necessários.

- **Semântica:** diz respeito à competência de extração de conhecimento dos objetos na IoT. Trata da descoberta de conhecimento e uso eficiente dos recursos existentes na IoT, a partir dos dados existentes, com o objetivo de prover determinado serviço.

Mas qual o propósito de dispor de uma grande diversidade destes dispositivos conectados?

A sua versatilidade, possibilita estabelecer comunicação entre e com outros dispositivos, tornando a IoT deslumbrante. Para Goussevskaia (2017), com a IoT é possível controlar, trocar informações entre os dispositivos, acessar serviços da Internet e interagir com pessoas possibilitando com isso criar cidades inteligentes (SmartCities), saúde (Healthcare) e casas inteligentes (SmartHome).

2.5 RECONHECIMENTO DE PADRÕES EM IMAGENS

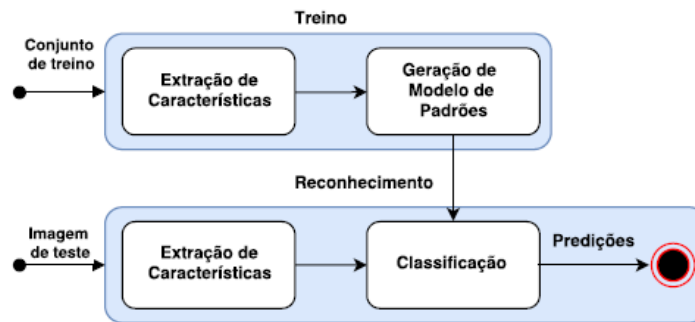
Para Lucena, Veloso e Lopes (2016), Queiroz e Gomes (2001), as dificuldades e os desafios no processamento de dados para o reconhecimento de objetos em imagens devem-se aos deveres dos sistemas serem eficientes no quesito taxas de reconhecimento, invariância das imagens quanto à rotação, ruído, luz, dentre muitas outras variáveis.

Usualmente as técnicas de processamento de imagens baseiam-se em métodos matemáticos que permitem descrever quantitativamente os mais diversos tipos de imagens. Para descrever-se uma imagem deve-se levar em consideração parâmetros que tenham característica bidimensional ou topológico, cor, luminosidade, dentre outros (ALBUQUERQUE; ALBUQUERQUE, 2004).

As principais tarefas de um software que identifica imagens sustentam-se na identificação de um objeto individual como um elemento de uma classe particular que engloba objetos com características semelhantes, necessitando para isso de descritores e classificadores. Pode-se resumir este processo nas fases de treinamento e reconhecimento, sendo que cada uma destas

possuirá as etapas de extração de características e classificação (LUCENA; VELOSO; LOPES, 2016).

Figura 5 - Esquema geral para algoritmos de reconhecimento de imagens.



Fonte: LUCENA; VELOSO; LOPES (2016).

A fase de treinamento do algoritmo consiste na busca de padrões para representar os objetos de interesse. Os padrões são discriminados por um conjunto de características extraídas utilizando-se um descritor que pode agir sob toda a imagem ou ao redor de pontos de interesse (GONZALEZ; WOODS, 2006).

Basicamente, obtêm-se um conjunto de imagens para treino e cria-se um o descritor através de medidas de superfície, perímetros, comprimentos, espessura, posição, luminosidade, cor, dentre outros (ALBUQUERQUE; ALBUQUERQUE, 2004). As características extraídas são submetidas a um algoritmo de aprendizado de máquina que busca separar as características originadas dos descritores, diferenciando o objeto de interesse dos demais e assim produzindo um modelo dos padrões (ADDICAM; MALIK; TIAN, 2012).

Na fase de reconhecimento, o objetivo é identificar objetos na cena a partir de um conjunto de medições chamados vetores de características (descritores) extraídos de objetos padrões. A classificação é realizada através de uma medida de similaridade entre os descritores da imagem analisada e de todos os modelos de descritores que descrevem os objetos das classes aprendidas. Desta forma, um conjunto de objetos similares, com uma ou mais características semelhantes, é considerado como pertencente à mesma classe de padrões (GONZALEZ; WOODS, 2006) e (LUCENA; VELOSO; LOPES, 2016).

Na tabela 1, com base em Albuquerque e Albuquerque (2004), são apresentadas descrições mais detalhadas das etapas de um sistema de processamento de imagens.

Tabela 1 - Resumo das etapas de um sistema de processamento de imagens

	ETAPA	DESCRIÇÃO
1.	Tratamento fora da imagem	Correção de iluminação, uso de colorantes químicos, etc.
2.	Aquisição da imagem	Amostragem, armazenamento e compactação.
3.	Melhoramento (" <i>image enhancement</i> ") *	Pré-tratamento digital da imagem.
4.	Segmentação da informação *	Extração dos "objetos" do "fundo" da imagem
5.	Parametrização *	Determinação de grandezas sobre cada "objeto": área, perímetro, forma, descrição estrutural, topologia, etc
6.	Reconhecimento *	Classificação dos "objetos"
7.	Análise Quantitativa: * Aplicação da ferramenta a outras áreas científicas.	Associação das grandezas ao problema: determinação de funções de correlação espacial ou temporal, análise de seqüência de imagens, etc.
* - Fases com extrema dependência ao problema onde o Proc. de Imagens está aplicado.		

Fonte: MOREIRA (2000).

2.6 LINGUAGENS MOBILE

O desenvolvimento de aplicativos móveis já é muito mais que realidade no nosso cotidiano, dentre as plataformas móveis que se destacam temos o Android da Google e o iOS da Apple, porém existem outras plataformas menos populares como o Windows Phone e a BlackBerry. Antes de desenvolver uma aplicação para esses tipos de dispositivos é fundamental que se tenha um conhecimento em relação as linguagens de programação que são compatíveis com esse modelo.

As principais linguagens de programação disponibilizam uma enorme base de códigos abertos, bibliotecas e frameworks para a comunidade de desenvolvedores, tornando assim o trabalho dos programadores mais fácil. Podemos destacar as linguagens Java, Python, Objective-C, Swift, C#. Segundo MONTEIRO (2015, p.2):

A plataforma Android desfruta hoje de um papel de destaque no mercado, tanto pela quantidade significativa de dispositivos produzidos como também por oferecer uma API rica, disponibilizando fácil acesso a vários recursos de hardware, tais como Wi-Fi e GPS, além de boas ferramentas para o desenvolvedor. A facilidade de desenvolver utilizando uma linguagem de programação (Java) bastante disseminada, a simplicidade e baixo custo para a publicação de aplicativos na loja Google Play e a quantidade de dispositivos Android em uso no mundo só fazem aumentar a popularidade da plataforma [...].

2.7 JAVA

Java é uma linguagem de programação multiplataforma, orientada a objetos, e tem a possibilidade de rodar em qualquer sistema operacional que possua um interpretador instalado. O interpretador é denominado como JVM, sendo um *software* que transforma o código da linguagem em comandos executáveis pelo sistema operacional.

A linguagem faz uso de um conceito diferente na geração dos códigos binários, ao invés de gerar códigos distintos para plataformas distintas, é gerado um binário que pode ser executado em qualquer plataforma, dentro do interpretador. Esse código “genérico” é conhecido como *bytecode*. Segundo Claro e Sobral (2008, p.15):

Um programa desenvolvido em Java necessita ser compilado, gerando um *bytecode*. Para executá-lo é necessário então, que um interpretador leia o código binário, o *bytecode* e repasse as instruções ao processador da máquina específica. Esse interpretador é conhecido como JVM (*Java Virtual Machine*). Os *bytecodes* são conjuntos de instruções, parecidas com código de máquina. É um formato próprio do Java para a representação das instruções no código compilado.

Pelo fato de ser uma linguagem que não depende de sistema operacional, a mesma pode ser utilizada em várias plataformas distintas, dentre elas os dispositivos portáteis, onde foi feita a programação e as integrações desta pesquisa. Em resumo, qualquer dispositivo que tenha uma JVM instalada pode executar softwares criados em Java, esse fato permite que códigos possam ser reutilizados dinamicamente independentemente da plataforma, posteriormente a vida do desenvolvedor é facilitada.

2.8 ANDROID

O Android é um sistema operacional que foi construído com base no sistema operacional Linux. Foi desenvolvido pela *OpenHandset Alliance*, uma aliança entre várias corporações de grande porte na área de tecnologia, dentre elas a *Google, Dell, Intel, Samsung, Nvidia*, entre outras. Por ser uma plataforma *Open Source*, está disponível gratuitamente para que os fabricantes personalizem, não há configurações fixas de hardware e software (LEE, 2011). Devido a popularidade e fácil acesso dos dispositivos *Android*, foi definida a utilização dessa plataforma para o desenvolvimento da aplicação.

A arquitetura da plataforma é um montante de aplicações com base *Linux* de código aberto criado para diversos fatores e formas (ANDROID DEVELOPERS GUIDE, 2018). A

divisão desse agrupamento consiste em *System Apps* na qual ficam os aplicativos nativos como Calendário, Câmera, Contatos, entre outros. Java API Framework que contém um conjunto completo de recursos do SO disponibilizados pelas APIs programadas na linguagem Java, esses blocos de código servem para facilitar o desenvolvimento das aplicações simplificando a reutilização de componentes e serviços já existentes. Native C/C++ Libraries contendo algumas bibliotecas nativas que são programadas em linguagem C/C++, a maior parte dessas bibliotecas tem funcionalidades para manipulação de gráficos. Android Runtime que possui artefatos para gerirem as instâncias e os processos a serem executados visando baixo consumo de memória. A camada de abstração física (Hardware Abstraction Layer) faz o primeiro acesso ao componente físico implementando interfaces específicas do dispositivo, preparando-o para o uso pela Java API de maior nível. Por último a camada Linux Kernel que guarda módulos de segurança, gerenciamento de memória, energia, ou seja, a parte física em baixo nível.

2.9 FERRAMENTAS DE DESENVOLVIMENTO

Neste presente projeto, busca-se um suporte para a maior quantidade possível de aparelhos, desde que suportem as APIs que serão utilizadas como Text-to-Speech e CloudVision. Observando os pré-requisitos para utilização das APIs e buscando o que foi implementado em cada versão (o que foi aprimorado nas APIs), percebe-se que seria necessário a API level 21 do Android pelo TTS (pelo fato do reconhecimento da linguagem da região) e da API level 19 a 20 que é pré-requisito mínimo de uso da CloudVision. Isso nos deixa com a versão mínima de uso do Lollipop (versão 5.0 do Android) com alcance de mais de 70% de usuários do Android.

2.10 API

Nos próximos tópicos, serão apresentadas algumas APIs que serão consumidas neste projeto, mas antes disso, será esclarecido o conceito de API.

API é uma sigla que significa Application Programming Interface, ou seja, Interface de programação para aplicativos (tradução livre). O conceito de API é bem amplo, mas segundo post no DevMedia (2018): “API é um conjunto de padrões de programação que permitem a construção de aplicativos e a sua utilização.”

Uma API possui classes, métodos e campos que deverão ser utilizados. Esta API pode ser de baixa ou alta complexidade, dependendo de quem a projetou, mas é dever do criador da API fazer algo que é intuitivo, fácil de ser utilizado por desenvolvedores (ou seja, pessoas que já

tenham conhecimento técnico), e além disso, o mais genérico possível, já que ela deve interligar várias funções de um aplicativo, e geralmente por vários tipos de aplicativo.

Uma API é considerada boa quando ela é compreensível pelos desenvolvedores que implementam e consomem a API, se ela é consistente com aquilo que ela propõe, se os métodos são fáceis de usar, se ela possui boa documentação e, principalmente, se ela é autossustentável, ou seja, se os custos dela são equivalentes ao retorno que ela provê.

Neste projeto, foi escolhida a utilização de uma API em vez de criá-la, justamente pela complexidade de criação e facilidade do uso. Decidiu-se focar principalmente em trazer a melhor experiência possível para o usuário. Ainda segundo DevMedia (2018), a primeira API não fica boa, ela deve ser testada e estudada com relação à escalabilidade, e como está sendo trabalhado com Machine Learning, seria requisitado uma massa de dados gigantesca, o que não será necessário, pois serão utilizadas as APIs da Google que já estão prontas e testadas. No caso da Google Cloud Text-to-Speech, esta API ainda está na versão Beta, mas ao pesquisar, percebe-se que já está muito bem desenvolvida e documentada.

2.10.1 GOOGLE CLOUDVISION

A Google Cloud Vision API (GCV) é uma ferramenta Open Source da Google que é bem prática de ser utilizada, pois já possui poderosos modelos de Machine Learning encapsulados.

Esta ferramenta visualiza uma imagem e buscando em um banco de dados criado previamente (você pode criar metadados ou usar o banco de dados da Google), reconhece o que está sendo apresentado para ela. Neste trabalho, esta será empregada para fazer a conversão de imagem em texto, e após isso, com outra API da Google, conhecida como Google Speech, o texto será convertido em voz. Esta API será apresentada no tópico adiante.

Ela também possui integrado um recurso de reconhecimento óptico, também conhecido como OCR (Optical Character Recognition) que identifica os caracteres e o idioma automaticamente de textos escritos em imagens.

Apesar de ser uma API Open Source, não significa que ela seja de graça, principalmente se a quantidade de utilização dos recursos da ferramenta for muito grande. Mas em geral, para até 1000 acessos por mês, atualmente todos os recursos são gratuitos. Ultrapassando este valor, é adicionado um custo para cada acesso.

Conclui-se desta forma que para a utilização neste projeto, esta ferramenta não terá custos no início e caso haja algum eventualmente, pode-se colocar alguns anúncios ou estudar uma outra forma de remuneração para cobrir os custos de manutenção do software.

2.10.2 GOOGLE CLOUD TEXT-TO-SPEECH

Outra ferramenta da Google que será utilizada neste projeto, será a Google Cloud Text-to-Speech API (TTS), na qual é possível converter o texto em áudio com a vantagem de ter a sua implementação e utilização mais simples em relação a Google Cloud Vision API.

Estando disponível em diversos idiomas, inclusive o Português, esta ferramenta é capaz de converter o texto (obtido através da imagem pelo GCV) para áudio, que poderá ser ouvido através do smartphone do usuário. Apesar de ser simples de implementar, é uma ferramenta muito poderosa que se baseia em pesquisas de *WaveNet* e redes neurais para entregar com a melhor fidelidade possível (GOOGLE CLOUD, 2018).

É uma ferramenta em expansão, mas que já está disponível em mais de 30 vozes e 12 idiomas, possível de alterar a velocidade de leitura para até quatro vezes mais rápido, sendo possível aumentar o som de saída em até 16 decibéis.

O valor desta ferramenta é também considerado baixo, assim como da GCV, sendo gratuito para até um milhão de caracteres mensais lidos (para vozes *WaveNet*), ou quatro milhões de caracteres mensais (para vozes que não são *WaveNet*), e sendo cobrado US\$4,00 e US\$16,00 para cada um milhão de caracteres que ultrapassarem o uso.

Concluimos que pela facilidade de uso, e pelo poder da ferramenta, esta é uma API indispensável para o nosso projeto, pois já foi desenvolvida e testada por muitas empresas grandes (como a própria Google, Rovio, Mistui, Evernote, Khan Academy, entre muitas outras) e continua em expansão com potencial de crescimento enorme, além de ser nativo Android (que é o nosso foco atual).

2.10.3 GOOGLE APP ENGINE

Repositórios de dados são conhecidos como coleções de informação digital, os mesmos podem ser desenvolvidos de diferentes maneiras e com propósitos distintos. Podem-se caracterizar por ser colaborativos, com um controle não tão intenso das informações e da autoridade dos documentos, tal como os repositórios dirigidos ao público em geral. Mas podem, também, ter um alto nível de controle em sua arquitetura, condensando o acesso às informações, exigindo uma complexidade maior para o acesso aos dados, que é o caso da Google app engine.

Esse mecanismo criado pela empresa Google, tem como principal objetivo fazer o controle das requisições das APIs que consomem as informações armazenadas para o

funcionamento correto das funcionalidades propostas, isso se aplica aos mecanismos citados anteriormente (GCV e TTS). Segundo Google (2018):

Os SDKs do Firebase para Cloud Storage usam o repositório padrão do Google App Engine. Isso significa que, quando um app do App Engine é criado, as App Engine APIs incorporadas podem ser usadas para compartilhar dados entre o Firebase e o App Engine. Isso é útil para executar codificação de áudio, transcodificação de vídeo e transformações de imagem, bem como outros processamentos de computação intensiva de segundo plano.

2.11 TECNOLOGIA ASSISTIVA

A Tecnologia Assistiva (TA) trata-se de uma área que incorpora metodologias, estratégias, produtos, recursos, práticas e serviços buscando proporcionar um maior grau de autonomia e qualidade de vida a pessoas com necessidades especiais (GOVERNO DO BRASIL, 2017), (FILHO, 2009), (ULBRICHT; VANZIN; VILLAROUCO, 2001).

A Tecnologia Assistiva (TA) trata-se de uma área que incorpora metodologias, estratégias, produtos, recursos, práticas e serviços buscando proporcionar um maior grau de autonomia e qualidade de vida a pessoas com necessidades especiais (GOVERNO DO BRASIL, 2017), (FILHO, 2009), (ULBRICHT; VANZIN; VILLAROUCO, 2001).

- Auxílios para a vida diária;
- CAA (CSA) Comunicação aumentativa (suplementar) e alternativa;
- Recursos de acessibilidade ao computador;
- Sistemas de controle de ambiente;
- Projetos arquitetônicos para acessibilidade;
- Órteses e próteses;
- Adequação Postural;
- Auxílios de mobilidade;
- Auxílios para cegos ou com visão subnormal;
- Auxílios para surdos ou com déficit auditivo;
- Adaptações em veículos.

O Brasil vem desenvolvendo algumas iniciativas através de políticas públicas, leis, decretos, entre outras, para voltados para a Educação Inclusiva no Brasil (GARRETT et al., 2017).

Desta forma, a tecnologia assistiva traz formas de fazerem valer os direitos dos cidadãos que possuem algum tipo de deficiência não anulando suas liberdades fundamentais.

3. DESENVOLVIMENTO

Neste capítulo será apresentado o desenvolvimento das ideias para um aplicativo a partir das técnicas descritas no capítulo anterior. Busca-se com ele atender as necessidades de um público alvo carente de algumas tecnologias possibilitando “novas visões”.

3.1 PÚBLICO ALVO

Este trabalho visa atingir o nicho de pessoas que possuem dificuldades ou deficiências visuais. Estas pessoas podem possuir cegueira, baixa visão, visão subnormal, entre outras.

Além deste público alvo, o projeto pode também atender pessoas não alfabetizadas, possibilitando-as de ouvir o conteúdo escrito.

3.2 OUVINDO O PÚBLICO ALVO

A etapa “Ouvir” iniciou-se por meio de conversas com pessoas com deficiência visual onde os problemas de sua rotina foram apresentados. Desta forma, constatou-se alguns problemas como:

Dificuldades em ler o cardápio;

Identificar imagens;

Falta de ferramentas voltadas para o público alvo;

Usabilidade de smartphones;

Falta de aplicativos;

Entre outros.

3.3 IDEIAÇÃO

Com os problemas do público alvo identificados, iniciou-se a fase de ideação. Foram realizadas algumas reuniões para se discutir possíveis “soluções” e chegou-se a definição do desenvolvimento de um aplicativo mobile o qual faria a conversão de uma imagem em áudio, e, para isto, seria necessário primeiramente converter a imagem em texto e, depois, o texto em áudio.

Com estas informações, foi possível elaborar as fichas com os requisitos funcionais e não funcionais, além de diagramas de caso de uso, atividade e classe, todos fundamentais para o desenvolvimento do App.

Depois de definidas possibilidades viáveis de solução quanto à parte financeira, tempo para o desenvolvimento e viável tecnologicamente, iniciou-se as argumentações e pesquisas para a identidade do protótipo.

3.3.1 Identidade Visual

Inicialmente, pensou-se a respeito de um possível nome para o aplicativo decidindo-se pelo nome *Learning Vision* fazendo uma alusão ao *machine learning* voltada à visão computacional. Com o passar do tempo e muitas outras reuniões, decidiu-se pelo nome **Visão Auxiliar**, pensando-se na facilidade da sua pronuncia e pela carga de informações que o nome carrega por si só.

Além disso, foi debatida uma identidade visual para o aplicativo de forma que este quando observado trouxesse indicativos da função do app. A figura 6 apresenta o logo representativo do aplicativo a ser desenvolvido neste projeto:

Figura 6 – Logo.



Fonte: Próprio Autor

O símbolo refere-se ao nome Visão Auxiliar com elementos que sugerem as letras V e A estando estas continuamente ligada. Observa-se também uma íris remetendo-se novamente o aplicativo a visão.

A fonte utilizada para o nome Visão Auxiliar foi a Bell MT. O nome deve estar abaixo do símbolo além possuir uma proporção em relação a este de 1:6.

Para propostas futuras, já estão sendo estudadas e construídas modificações para a utilização do logo quando necessários fundos de outras cores.

Por fim, foram realizadas pesquisas do nome Visão Auxiliar nos serviços online do Instituto Nacional da Propriedade Industrial (INPI). Segundo o INPI (2018), não foram encontrados nas suas 45 classificações de produtos e serviços o nome proposto ao nosso aplicativo.

3.4 PROTOTIPAÇÃO

A fase de prototipação teve seu início logo após a apresentação da ideia do projeto para a banca avaliadora do TG.

Por meio de suas ideias, críticas e elogios, foi possível tatear um pouco mais sobre a importância que o projeto teria para a vida das pessoas que possuem deficiência visual.

A partir disso, todo o conteúdo absorvido ao longo da pesquisa e interação com o público alvo começou a se modelar nos protótipos iniciais.

3.4.1 INÍCIO DO DESENVOLVIMENTO

Como ponto principal do projeto, o estudo sobre como as APIs poderiam ser integradas foi ponto chave para elaboração do protótipo inicial.

Dividindo os trabalhos, o grupo atuou em duas frentes no projeto, uma para integrar a API Google *CloudVision* e outra para integrar a API Google *Text-To-Speech*.

Por mais que as APIs já estivessem desenvolvidas e testadas pelos seus responsáveis, a integração provou certa dificuldade devido à complexidade e completude de funcionalidades que mesma proporciona.

3.4.2 INTEGRAÇÃO COM AS APIS GOOGLE

A API Google *CloudVision*, responsável pelo reconhecimento de imagem e transformação do conteúdo em texto, foi o ponto principal no desenvolvimento do protótipo.

O desenvolvimento inicial consistiu em fazer a API funcionar integrando-a com a câmera do aparelho com o sistema Android. Após isto, foi utilizada uma funcionalidade para o reconhecimento do texto, tendo esta as seguintes características:

- Mapear todo o texto reconhecido na tela do *Smart Phone*;
- Disponibilizar as palavras em quadros para posterior conversão para áudio na API TTS;

Os primeiros resultados foram simplórios, porém promissores, mostrando todo o poder da integração.

De início, o reconhecimento do texto era feito, porém seu conteúdo era disponibilizado em linguagem estrangeira (Inglês).

Um outro ponto identificado, foi que, para o reconhecimento do texto, era necessário que o usuário efetuasse um “clique” na tela para disparar a ação e ter o conteúdo disponibilizado em texto. Um fator crítico, pois, tendo em vista a acessibilidade, esse provaria certa dificuldade para o usuário final.

Levando os dois pontos identificados em questão, a equipe elaborou discussões sobre como poderia ser realizada a adaptação para a linguagem nativa do público alvo inicial (Português), e como poderia ser feito o reconhecimento do texto apenas com o usuário apontando sua câmera para o mesmo ou, o ativando por meio de comandos de voz.

Concluiu-se que seria possível realizar esta adaptação, porém a mesma exigiria certo esforço para ser bem-sucedida. No caso do gatilho para o reconhecimento, manteve-se o clique na tela, pois o mesmo não se mostrou um empecilho para o reconhecimento do texto.

3.4.2.1 CONFIGURANDO A LINGUAGEM DE FALA DA API TTS

Para alterar a linguagem em que o texto era falado pela API TTS, foi feita uma adaptação no código, a qual fez com que o aplicativo levasse em consideração o idioma configurado no sistema Android.

3.4.2.2 MAPEAMENTO DO TEXTO RECONHECIDO

Para padronizar como é feita a leitura do texto reconhecido, uma das funções da API foi redesenhada para identificar as coordenadas dos quadros e ser executada da seguinte maneira:

- Mapear os quadros de cima para baixo;
- Mapear os quadros da esquerda para a direita;

Dessa maneira, os quadros são organizados e lidos da mesma maneira que os usuários nativos estão familiarizados.

3.4.2.3 INTERROMPENDO A FALA DA API TTS

Muitas vezes o texto reconhecido pode ser extenso, por conta disso, há uma demora até que se possa ler um novo texto. Tendo isso em mente, foi desenvolvida a seguinte funcionalidade:

- Verifica se a API TTS está ditando algo;
- Se estiver ditando algo, é possibilitado ao usuário interromper a fala do aplicativo;

Para tal, o usuário pode clicar em qualquer ponto da tela para que a fala da API TTS seja interrompida. Dessa maneira, o usuário não necessita aguardar até que todo o texto reconhecido seja dito antes de poder reconhecer um novo.

3.4.3 ACESSANDO O APLICATIVO

A acessibilidade é o principal ponto do projeto. Visando isto, desenvolveu-se um gatilho para que o usuário possa abrir o aplicativo por um comando de voz no sistema Android utilizando o Google Assistant. O comando utilizado deve ser as palavras “*Visão Auxiliar*”.

3.4.3.1 TELA DE CARREGAMENTO

Para que o aplicativo seja carregado corretamente, após abri-lo, há um tempo de espera de 5 segundos em sua inicialização.

Durante este tempo é mostrada uma tela de carregamento contendo o logo do aplicativo. Além disso, é utilizada a API TTS para informar ao usuário em áudio que o mesmo executou o aplicativo “*Visão Auxiliar*”. A tela de carregamento pode ser visualizada na figura 7.

Figura 7 - Tela de carregamento

Fonte: Próprio Autor.

3.4.3.2 Tela inicial

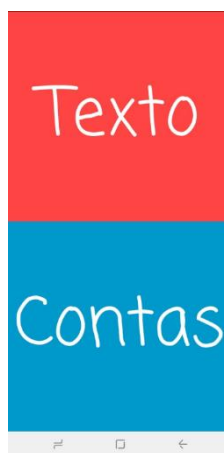
Para a tela inicial do aplicativo, foi escolhido um *design* que proporcionasse maior acessibilidade ao usuário.

Com textos grandes e cores fortes, a tela inicial visa também auxiliar os usuários que possuem baixa visão. Para os usuários que não podem enxergar devido a sua deficiência, ao acessar a tela, é fornecida a seguinte instrução em áudio:

- Clicando na parte superior, o usuário pode reconhecer textos;
- Clicando na parte inferior, o usuário pode reconhecer contatos;
- Arrastando para o lado ou clicando e segurando na tela, o usuário pode finalizar o aplicativo.

A figura 8 ilustra a tela inicial do app.

Figura 8 - Tela inicial



Fonte: Próprio Autor.

3.4.3.3 Telas de reconhecimento

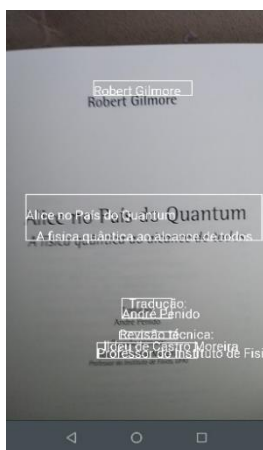
A tela de reconhecimento mostra a imagem que está sendo transmitida pela câmera do aparelho.

Para o reconhecimento do texto, o usuário executa os seguintes passos:

- Aponta a câmera de seu *smartphone* para o conteúdo que deseja reconhecer;
- Clica na tela para que o texto reconhecido seja ditado.

A figura 9 ilustra a captura da imagem e a sua conversão em texto pelo app.

Figura 9 – Captura da imagem e conversão em texto



Fonte: Próprio Autor.

- Caso queira interromper a fala do aplicativo, o usuário pode clicar novamente na tela.

No caso da conta de consumo, o QR code é lido assim que identificado. Um modelo proposto pela equipe de pesquisa apresenta-se na figura 10.

Figura 10 – Sugestão de conta de consumo com QR code



Fonte: Próprio Autor.

3.4.5 Testes com o público alvo

Para verificar a eficiência do aplicativo, foram elaborados alguns testes para serem executados por um usuário com deficiência visual. Os testes executados foram os seguintes:

- **Abrir o aplicativo por comando de voz:** O aparelho *smartphone* utilizado nos testes foi configurado para que conhecesse a voz do usuário. Após a configuração, foi solicitado ao usuário para que falasse o comando “Ok, Google” para iniciar o assistente, e, após iniciado, falar a frase “Abrir Visão Auxiliar” para iniciar o aplicativo;
- **Leitura de textos:** Para este teste, o usuário posicionaria a câmera do *smartphone* sobre o conteúdo e efetuaria um clique na tela para ouvir o texto reconhecido. Foram utilizados dois conteúdos para o teste, um escolhido pelo grupo e outro escolhido pelo próprio usuário, afim de garantir a veracidade do reconhecimento do texto.
- **Leitura de contas:** Seguindo o modelo proposto pelo grupo, foi disponibilizado ao usuário, uma folha impressa com uma conta de energia elétrica com um QRCode armazenando as informações do nome da pessoa, valor, data de vencimento, para que o mesmo pudesse efetuar o reconhecimento.
- **Usabilidade em geral:** Este teste consiste em verificar juntamente ao usuário, toda a questão de comandos existentes no aplicativo.

Os testes foram em sua maioria muito bem-sucedidos, constatando que o aplicativo pode fornecer um auxílio muito grande as pessoas com deficiência visual.

3.4.6 Avaliações do público alvo

As avaliações descritas neste tópico foram coletadas de toda a experiência que tivemos com usuário, sendo estas obtidas nos testes, em conversas e comentários que foram feitos.

Dentre estes podemos destacar:

- **Leitura de textos:** Foi obtida uma avaliação positiva sobre a qualidade na qual a leitura foi executada, sendo esta considerada “quase perfeita” pelo usuário. O mesmo ficou extremamente contente com o resultado obtido.
- **Leitura de contas:** Um dos maiores problemas citados pelo usuário na sua vida pessoal era a leitura de contas, pois esta não possui acessibilidade alguma, levando o usuário com deficiência a ter que pedir para outras pessoas falarem os valores de sua conta. O usuário demonstrou total satisfação quando viu a velocidade na qual a conta foi reconhecida por meio da leitura do QRCode. O usuário também ficou contente com o texto sendo repetido após um determinado período de tempo, pois, segundo ele, algumas vezes uma informação ou outra acaba não sendo ouvida de maneira correta.
- **Usabilidade:** O usuário demonstrou facilidade no uso do aplicativo. O mesmo informou que os comandos por toque estavam fornecendo uma acessibilidade muito grande, e que as instruções por voz guiavam bem o uso do aplicativo.

4. CONSIDERAÇÕES FINAIS

Atualmente já existem alguns aplicativos que são desenvolvidos pensando na acessibilidade de pessoas com deficiência visual, porém ainda são poucos e em sua grande maioria para a plataforma IOS. O foco deste trabalho, não visa auxiliar no uso do smartphone e sim auxiliar na acessibilidade do usuário, possibilitando a leitura de cardápios, placas com textos, embalagens dentre outros objetos textuais, amenizando algumas dificuldades dessas pessoas, possibilitando-lhes uma maior liberdade.

Utilizou-se da metodologia *design thinking* para dar vida à um projeto bem fundamentado e viável frente aos quesitos tempo, financeiro e exequibilidade. Foram coletadas

e analisadas informações sobre as necessidades de pessoas com deficiência visual, identificando-se algumas necessidades que poderiam ser tangenciadas para uma melhora com um auxílio à visão. A proposta foi modelar um aplicativo mobile que auxilie a leitura e interpretação do mundo, buscando-se minimizar os obstáculos encontrados por quem tem algum tipo de deficiência (de visão, ou até mesmo de leitura).

Ao longo do ano de 2018 foram desenvolvidas as integrações com as APIs e o aplicativo proposto. Após a conclusão do desenvolvimento, iniciou-se um período de testes com o público alvo. Estes testes forneceram uma visão do quão é importante promover algo que traga acessibilidade a pessoas com algum tipo de deficiência. A alegria no rosto das pessoas que conseguem acesso a uma informação que antes não conseguiam traz um sentimento de felicidade muito grande.

De maneira geral, o objetivo do aplicativo foi cumprido: reconhecer um conteúdo em texto e o proporcionar em áudio ao usuário.

Para o futuro, visando obter mais precisão e qualidade para o aplicativo, serão feitos estudos e análises a respeito de acessibilidade afim de seguir regras e padrões instituídos pelo governo e pelas Diretrizes de Acessibilidade para o Conteúdo da Web (**WCAG**, do inglês *Web Content Accessibility Guidelines*). Além disso, serão analisadas possíveis conflitos do aplicativo com o função de acessibilidade do Android chamado “*TalkBack*”. Esta função é responsável por fazer a leitura da tela do aparelho e disponibilizar ao usuário seu conteúdo.

Para visualização do funcionamento do aplicativo por meio de um vídeo explicativo, o leitor pode utilizar o QR code ou o link disposto na figura 10.

Figura 10 – Vídeo demonstrativo do aplicativo.



Fonte: Próprio Autor: https://youtu.be/J_EEjEt8xcs

REFERÊNCIAS

ADDICAM, S. MALIK, S. and TIAN, P. **Building Intelligent Systems**. Intel Press., 2012.

ALBUQUERQUE, Márcio Portes de; ALBUQUERQUE, Marcelo Portes de. **Processamento de Imagens: Métodos e Análises**. Rio de Janeiro: Centro Brasileiro de Pesquisas Físicas – Cbpf/mct, 2004.

ANDROID DEVELOPERS GUIDE. Disponível em: <https://developer.android.com/guide/platform/>. Acesso em: 30 de abril de 2018.

APPLICATION PROGRAMMING INTERFACE: DESENVOLVENDO APIS DE SOFTWARE. Disponível em: <<https://www.devmedia.com.br/application-programming-interface-desenvolvendo-apis-de-software/30548>> Acesso em 20 de maio de 2018.

BRASIL. Decreto nº 3.956, de 08 de out. de 2001. **Convenção Interamericana para a Eliminação de Todas as Formas de Discriminação contra as Pessoas Portadoras de Deficiência**, Brasília, DF, out 2001.

CARLO, James J. Di Carlo. ZOCCOLAN, Davide. RUST, Nicole C. **How Does the Brain Solve Visual Object Recognition?** Neuron. Volume 73, Issue 3, p.415-434, 9 de fevereiro de 2012.

CLARO, Daniela; SOBRAL, João Bosco. **Programação em Java**. 1ª edição, Florianópolis: Editora Pearson, 2008.

CLOUD-TO-SPEECH. Disponível em: <<https://cloud.google.com/text-to-speech/>> Acesso em 20 de Maio de 2018.

CLOUD VISION API. Disponível em: <<https://cloud.google.com/vision/?hl=pt-br>> Acesso em 20 de Maio de 2018.

Documentation of Text-to-Speech. Disponível em: <<https://developer.android.com/reference/android/speech/tts/TextToSpeech>> Acesso em 20 de Maio de 2018.

FILHO, T. A. A. G. **Tecnologia Assistiva: de que se trata?** In: MACHADO, G.J. C.; SOBRAL, M. N. (Orgs.). *Conexões: educação, comunicação, inclusão e interculturalidade*. 1 ed. Porto Alegre: Redes Editora, p. 207-235, 2009.

GARRETT, Juliana de Oliveira Barbosa et al. **TECNOLOGIA ASSISTIVA: INCLUSÃO EDUCACIONAL E SOCIAL.** *Revista Perspectiva Online: Humanas & Sociais Aplicadas*, Online, v. 7, n. 19, p.10-22, jul. 2017.

GONZALEZ, R. C. WOODS, R. E. **Digital Image Processing.** 3rd Edition. Prentice-Hall, Inc., 2006.

GOUSSEVSKAIA, Olga N. et al. **Internet das Coisas: da Teoria à Prática.** UFMG, 2017.

Governo do Brasil. **Tecnologia assistiva ajuda a melhorar a qualidade de vida de pessoas com deficiência.** 2017. Disponível em: <<http://www.brasil.gov.br/noticias/educacao-e-ciencia/2010/08/tecnologia-assistiva>>. Acesso em: 05 nov. 2018.

HARNAD, Stevan. **The Annotation Game: On Turing (1950) on Computing, Machinery, and Intelligence.** 2008. Disponível em: <<https://eprints.soton.ac.uk/262954/1/turing.html>>. Acesso em: 15 de abril de 2018.

Integrar com o Google Cloud Platform. Disponível em: <<https://firebase.google.com/docs/storage/gcp-integration?hl=pt-br>> Acesso em 9 de junho de 2018

INDUSTRIAL, Instituto Nacional da Propriedade. **Serviços Online:** Pesquisa de marcas. Disponível em: <<https://servicosonline.inpi.pt/pesquisas/main/marcas.jsp?lang=PT>>. Acesso em: 02 de Novembro de 2018. Notas de Aula, 2014. 38 slides, color.

JOLY, Martine. **Introdução à Análise Da Imagem.** Papirus Editora, 11 edição, 2007.

ROSEN, Jill. **How the brain instantly separates trash from treasure.** *Science+Technology*. Disponível em: <https://hub.jhu.edu/2018/02/08/brain-identifies-valuable-items/>. Acesso em: 26 de Março de 2018.

SAMUEL, A. L.. **Some Studies in Machine Learning Using the Game of Checkers.** *Ibm Journal Of Research And Development*, [s.l.], v. 3, n. 3, p.210-229, jul. 1959. IBM. <http://dx.doi.org/10.1147/rd.33.0210>.

SANTOS, B.P. et Al. **Internet das Coisas: da teoria à Prática.** Anais do Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC), 2016.

SDH/PR. **Acessibilidade.** Secretaria de Direitos Humanos da Presidência da República. Disponível em: <http://www.pessoacomdeficiencia.gov.br/app/acessibilidade-0>. Acesso em: 10.06.2018.

SEBRAE. **Entenda o design thinking.** Disponível em: <http://www.sebrae.com.br/sites/PortalSebrae/artigos/entenda-o-design-thinking,369d9cb730905410VgnVCM1000003b74010aRCRD>>. Acesso em: 03 out. 2017.

TURING, Alan M. **Computing Machinery and Intelligence.** *Mind*, [s. L.], v. 59, n. 236, p.433-460, out. 1950.

ULBRICHT, Vania Ribas; VANZIN, Tarcísio; VILLAROUCO, Vilma. **Ambiente Virtual De Aprendizagem Inclusivo.** Florianópolis: Pandion, 2001.

VIANNA, Maurício et al. **Design thinking: inovação em negócios.** Rio de Janeiro: Mjv Press, 2012.