



**Thales Melo da Silva<sup>1</sup>**  
**Pablo Santos Werlang<sup>2</sup>**

## RESUMO

Segundo a Cartilha do Censo 2010 Pessoas com Deficiência[1], Das 45.606.048 pessoas com deficiência no Brasil, 1,6% são totalmente cegas, e estas pessoas possuem muitos problemas de acessibilidade, como por exemplo a dificuldade de identificar produtos em uma prateleira de supermercado ou saber o que está escrito em uma embalagem. Através do uso de tecnologia assistiva (TA) este estudo visa auxiliar na inclusão social de deficientes visuais. Por meio de um aplicativo para dispositivos móveis capaz de capturar uma imagem de um objeto e através de uma *Application Programming Interface* (API) de reconhecimento de imagens, pretende-se reconhecer o objeto e descrever por meio de voz para o usuário o que é aquele objeto e suas características. Já existem algumas soluções análogas ao projeto proposto, como por exemplo os aplicativos Aipoly Vision e Blindo, porém, elas são pagas, não atendem completamente as necessidades das pessoas com deficiência visual e não estão disponíveis em português.

**Palavras-chave:** *deficientes visuais, acessibilidade, tecnologia assistiva.*

## 1 INTRODUÇÃO

A Cartilha do Censo 2010 Pessoas com Deficiência [1], diz que 45.606.048 de pessoas no Brasil têm algum tipo de deficiência, ou seja, 23,9% da população total, sendo 3,46% com deficiência visual severa e 1,6% totalmente cegas. As pessoas com deficiência visual possuem muitos problemas de acessibilidade, como por exemplo a dificuldade de diferenciar produtos com embalagens parecidas, de saber informações contidas nas embalagens de produtos, na leitura de uma revista ou jornal, de identificar objetos em geral, de identificar cores, entre outras.

Segundo Alves et al [2]:

A visão é o sentido mais utilizado pelos seres humanos, que possibilita o conhecimento do mundo ao seu redor. Para uma pessoa que nunca teve problemas visuais é improvável saber como seria se tivesse nascido definitivamente cego, assim como não há como imaginar sua reação se ficasse cego mais tarde e enfrentasse as dificuldades de adaptação.

Através do uso de Tecnologia Assistiva (TA) alguns destes problemas podem ser reduzidos ou contornados, segundo Bersch [3], "a TA deve ser então entendida como um auxílio que promovera a ampliação de uma habilidade funcional deficitária ou possibilitara a realização da função desejada e que se encontra impedida por circunstância de deficiência ou pelo envelhecimento".

As pessoas com deficiência visual tem a necessidade de descobrir recursos, equipamentos e estratégias que facilitem o aprendizado e ajudem nas atividades diárias, principalmente de leitura e

<sup>1</sup> Instituto Federal do Sul-rio-grandense – Campus Charqueadas - Autor

<sup>2</sup> Instituto Federal do Sul-rio-grandense – Campus Charqueadas - Orientador



escrita. O uso de TA pode ser um potencializador para o desempenho das pessoas com deficiência visual neste tipo de atividades.

Gasparetto [4] diz que a tecnologia assistiva é uma área do conhecimento e de atuação que desenvolve serviços, recursos e estratégias que auxiliam na resolução de dificuldades funcionais das pessoas com deficiência para a realização de suas tarefas no cotidiano.

Se houvesse uma TA capaz de ajudar pessoas com deficiência visual a contornar os problemas de acessibilidade citados acima, então estas pessoas não precisariam pedir ajuda para outra pessoa para identificar objetos, placas, embalagens, animais, formas e cores, poderiam realizar sozinhas a “leitura” de livros, jornais e revistas. A existência de uma TA que realizasse tais funções seria muito útil para as pessoas com deficiência visual, pois aumentaria muito a inclusão social, a independência e o processo de aprendizado delas.

Aplicativos e sistemas de visão artificial, também chamada de visão computacional, são amplamente utilizados para vários tipos de finalidades, tais como: classificação, reconhecimento e detecção de objetos; estimativa de parâmetros como tamanho, posição, orientação e velocidade, entre outros. Uma das aplicações de visão artificial mais utilizadas é o Reconhecimento Ótico de Caracteres ou *Optical Character Recognition* (OCR). As aplicações deste tipo realizam o reconhecimento ótico de caracteres, letras, algarismos, símbolos escritos, entre outros.

Em virtude das necessidades apresentadas anteriormente, o presente artigo apresenta o desenvolvimento de um aplicativo para *smartphones* com o sistema operacional Android controlado via comandos de voz, por meio de uma *Application Programming Interface* (API) que transforma voz em texto, e capaz de capturar a imagem de um objeto ou texto, e a partir dessa imagem fazer o reconhecimento de características do objeto, como cor, tipo, caracteres, texto e logomarca, utilizando uma API de análise de imagens. Após o reconhecimento das características da imagem, o aplicativo vai reproduzir na forma de áudio essas características para o usuário conforme ele solicitar, através de uma API que transforma texto em áudio.

O artigo está organizado como segue. Na seção 2, é apresentado um levantamento do estado da arte no tema do projeto e os principais trabalhos relacionados ao projeto. Na seção 3, relata-se a metodologia empregada no desenvolvimento do aplicativo e o funcionamento do mesmo. Na seção 4, o artigo apresenta os principais resultados e discussões sobre o projeto. E por fim na seção 5, são apresentadas as principais conclusões e trabalhos futuros.

## **2 TRABALHOS RELACIONADOS**

Esta seção do artigo visa desenvolver um levantamento do estado da arte no tema do projeto, através da revisão de trabalhos relacionados ao tema e aplicativos análogos, identificando os principais pontos de divergência e convergência com o presente projeto.



*LookTel Money Reader* (ou *NantMobile Money Reader*) [5] é um aplicativo para dispositivos móveis com o sistema operacional iOS (antes chamado de iPhone OS) capaz de auxiliar as pessoas com deficiência visual a fazer o reconhecimento do valor de cédulas monetárias de mais de 20 moedas diferentes e informar o valor resultante da classificação através de áudio. O aplicativo faz muito bem o que se propõe a fazer, ele é gratuito e está disponível em português dentre outras línguas, porém só está disponível para dispositivos móveis com o sistema operacional iOS e faz apenas o reconhecimento de cédulas monetárias.

*LookTel Recognizer* [6] é um aplicativo para dispositivos móveis com o sistema operacional iOS que é capaz de auxiliar as pessoas com deficiência visual no reconhecimento instantâneo objetos do dia a dia, como mercadorias embaladas na despensa, cartões de identidade, latas de refrigerante na mercearia ou CDs em uma coleção de músicas. Uma vez que uma biblioteca tenha sido construída, os usuários podem simplesmente apontar a câmera do iPhone para um objeto e o telefone reconhecerá e descreverá o item instantaneamente através de áudio. O aplicativo é pago, não está disponível nas versões mais recentes do sistema operacional iOS e ele utiliza de uma Rede Neural Artificial (RNA) que necessita ser treinada pelo próprio usuário para que funcione, exemplo, para que o aplicativo consiga reconhecer um cavalo, o usuário precisaria inserir diversas fotos diferentes de cavalos, para que o aplicativo consiga reconhecer um cavalo.

Aplicativo identificador de cédulas para deficientes visuais [7] é uma proposta de aplicativo para plataforma Android capaz de auxiliar os deficientes visuais no reconhecimento do valor de cédulas monetárias de Real e informar o valor resultante da classificação de forma sonora. Infelizmente o aplicativo não saiu da fase de desenvolvimento e não foi lançado.

*BlindTool* [8] é um aplicativo para a plataforma Android capaz de auxiliar os deficientes visuais no reconhecimento instantâneo de objetos. Infelizmente o aplicativo não está disponível em português e não saiu da fase de desenvolvimento, foram realizados diversos testes com o aplicativo em questão, porém o mesmo se mostrou muito instável, fechando sozinho sem motivo aparente e realizando o reconhecimento incorreto de diversos objetos em diversas situações, estes problemas devem-se ao fato de o aplicativo não ter saído da fase de desenvolvimento.

Aipoly Vision [9] é um aplicativo que faz o reconhecimento de objetos e cores, que serve para ajudar os deficientes visuais e cegos a entender o seu espaço. Basta apontar o *smartphone* para o objeto de interesse e pressione qualquer um dos botões de reconhecimento na parte inferior da tela para ativar que a inteligência artificial do aplicativo reconheça o objeto, e depois falar em voz alta o que foi reconhecido. O aplicativo não está disponível em português, e não é totalmente gratuito, já que para o usuário utilizar todo potencial do aplicativo ele deve realizar a assinatura do aplicativo no valor de R\$ 15,99 por mês, caso o usuário não queira pagar a mensalidade, então ele só poderá utilizar de funções básicas do aplicativo, como o reconhecimento de cores. O aplicativo está disponível para os sistemas operacionais Android e iOS.



*TapTapSee* [10] é aplicativo com a proposta semelhante ao *Aipoly Vision*, porém gratuito e disponível para os sistemas operacionais Android e iOS. O aplicativo capta de uma a três imagens do objeto a ser identificado, e depois ele mostra escrito na parte inferior da tela o que conseguiu identificar. O aplicativo *TapTapSee* se diferencia dos demais aplicativos semelhantes, por dar uma descrição muito mais detalhada do objeto, porém ele demora um pouco mais para exibir os resultados, não está disponível em português e não exibe o resultado no formato de áudio.

Assim como o *Aipoly Vision* e o *TapTapSee* existem diversos aplicativos que fazem a mesma função, alguns exemplos são o *BlindSight* [11] e o *Toolwiz Eyes-Voice Guide Cam* [12], porém nenhum deles estão disponíveis na língua portuguesa, possuem uma interface confusa e algumas vezes dão resultados errados.

O aplicativo *Be My Eyes* [13] permite que o usuário deficiente visual conecte-se via chamada de vídeo com outro usuário não cego e voluntário, assim então o usuário cego pode pedir auxílio do voluntário para identificar cores, objetos, fazer leitura de texto, entre outras coisas. Infelizmente para que o aplicativo seja útil o usuário deficiente visual depende de dois fatores, o primeiro é uma ótima conexão com a internet e o segundo é um voluntário que fale a mesma língua, que ele esteja disponível e que saiba identificar o que o usuário deficiente visual precisa identificar. O aplicativo tem uma ideia ótima, porém ele não dá uma autonomia para o usuário deficiente visual.

O dispositivo *OrCam MyEye 2.0* tem uma proposta muito parecida com a do presente projeto, ele está disponível em diversas línguas, incluindo o português, não necessita de um *smartphone* para funcionar, é pequeno, leve, é fácil de ser transportado pelo usuário e é ótimo em fazer o que se propõe.

Um dispositivo de visão artificial com uma câmera inteligente, leve (pesa apenas 22,5 gramas) que lê instantaneamente textos de qualquer superfície e reconhece rostos, produtos, cores e cédulas de dinheiro em tempo real. Por ser dotado de leds, pode operar também no escuro. O dispositivo conta com uma tecnologia avançada de reconhecimento de faces que além de auxiliar o usuário a identificar as pessoas ao seu redor, diz também, o gênero de cada uma delas. Tudo em um pequeno dispositivo do tamanho do seu dedo. Mais Autonomia: *OrCam MyEye®* [14]

Ele é capaz de fazer a leitura de textos, identificação de objetos, cédulas de monetárias, reconhecimento de rostos e cores com rapidez e precisão, porém o dispositivo deve ser comprado e custa R\$ 17.400,00 a vista no Brasil. Infelizmente um dispositivo que custa quase o valor de um carro popular não é algo acessível para todos no Brasil.

### 3 DESENVOLVIMENTO

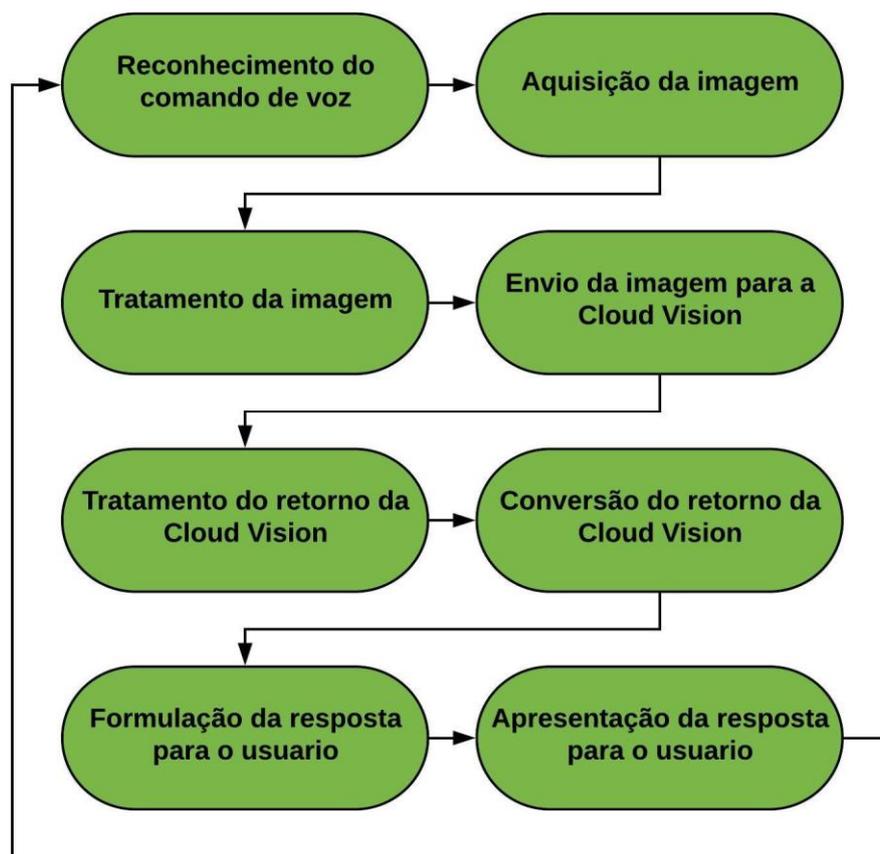
A partir da contextualização e definição do problema apresentado nas seções anteriores o objetivo deste projeto foi criar um aplicativo para dispositivos móveis, capaz de auxiliar deficientes visuais a realizarem sozinhos a leitura de textos, identificação de objetos e cores.



O projeto está sendo desenvolvido em linguagem de programação Java na plataforma Android e utiliza de quatro APIs: a *Cloud Vision API* que é capaz de fazer análise, classificação e detecção de objetos em imagens, a *Speech To Text (STT)* que transforma fala em texto através do uso do microfone do *smartphone* e que já é uma API integrada do Android, a *TextToSpeech (TTS)* que transforma texto em fala através do uso do alto-falante do *smartphone* e que já é uma API integrada do Android e a *Translation API* que é utilizada para traduzir texto dentro do aplicativo.

A Figura 1 mostra um fluxograma básico do funcionamento do aplicativo.

**Figura 1. Fluxograma do funcionamento do aplicativo**



## **A. Reconhecimento do comando de voz**

O usuário vai interagir com o aplicativo via comandos de voz, falando para o aplicativo o que ele deseja saber, exemplo: caso ele queira saber a cor de algo, ele vai apontar a câmera do *smartphone* para o que ele quer saber a cor e vai dizer "COR" ou caso ele queira saber o que está escrito, então ele



vai dizer "LEIA". Então o aplicativo vai identificar através da STT API o que o usuário falou e vai transformar em texto.

## **B. Aquisição da imagem**

Após a interação do usuário com o aplicativo por meio de um comando de voz válido, então o aplicativo automaticamente vai utilizar a câmera do *smartphone*, através da API de câmera própria do sistema Android, para fazer a captura da imagem.

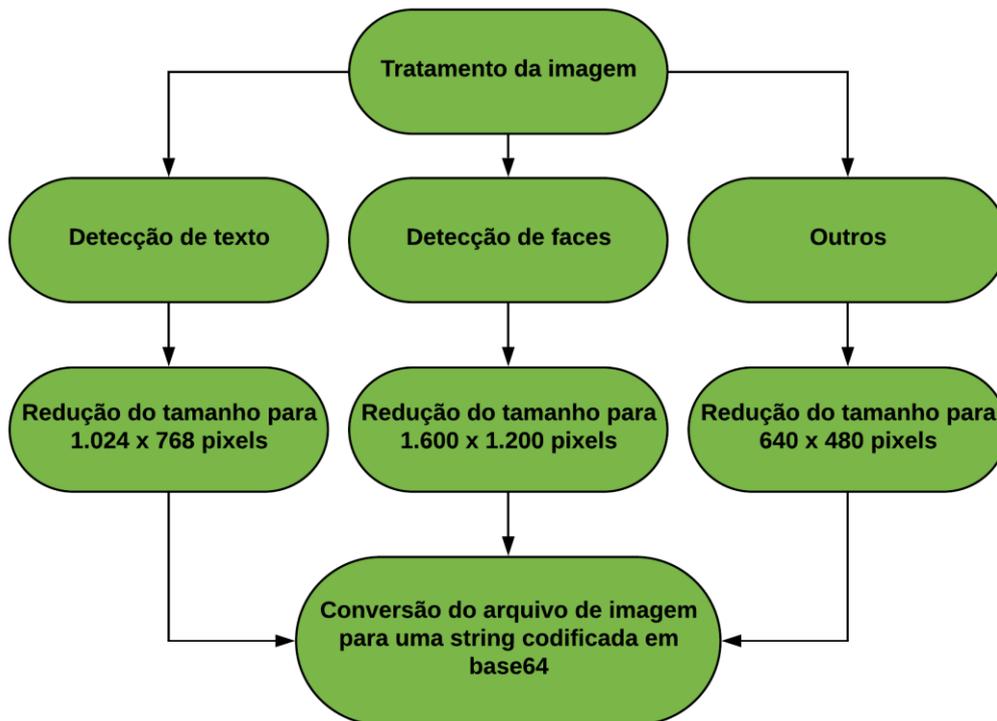
## **C. Tratamento da imagem**

Após a interação do usuário com o aplicativo por meio de um comando de voz válido e a aquisição da imagem, então o aplicativo vai passar para a parte de tratamento da imagem. A imagem vai passar por um tratamento diferente conforme o tipo de solicitação feita pelo usuário, caso ele faça uma solicitação do tipo de detecção de faces, então a imagem será redimensionada para o tamanho de 1.600 x 1.200 *pixels* (Um *pixel* é o menor componente que forma uma imagem digital), caso ele faça uma solicitação do tipo de detecção de textos, então a imagem será redimensionada para o tamanho de 1.024 x 768 *pixels* e caso ele faça algum outro tipo de solicitação, como por exemplo a de detecção de cor ou a detecção de características, então a imagem será redimensionada para o tamanho de 640 x 480 *pixels*, estes são os tamanhos recomendados na documentação da *Cloud Vision API*, porém através de testes notou-se que a API responde bem a tamanhos menores também, porém optou-se por utilizar os tamanhos recomendados para que o aplicativo tenha melhores resultados. Esse tratamento é necessário para que o tamanho em *Bytes* (Um *Byte* é um conjunto de oito *Bits*, um *Bit* é a menor unidade de armazenamento digital) seja reduzido, sem perda de qualidade da imagem, é necessária essa redução para que o aplicativo faça um menor consumo de tráfego de rede. Após o redimensionamento da imagem, então ela deve ser transformada em uma *String* (é uma sequência de caracteres) codificada em base64 (é um método de codificação de dados para transferências na internet), para que possa ser feita a comunicação com a *Cloud Vision API*, devido ao fato de que a *Cloud Vision API* usa o *JavaScript Object Notation* (JSON) para solicitações e para respostas.

A Figura 2 mostra um fluxograma interno do processo de tratamento de imagem.



Figura 2. Fluxograma interno do tratamento de imagem



#### ***D. Envio da imagem para a Cloud Vision API***

Após o processo de tratamento da imagem o aplicativo faz a comunicação com a *Cloud Vision API* via solicitações *Hypertext Transfer Protocol* (HTTP) do tipo POST (é um método de requisição suportado pelo protocolo HTTP), esta é a única forma de comunicação aceita pela *Cloud Vision API*, O corpo da solicitação POST contém um objeto JSON, o objeto JSON vai conter a *string* da imagem codificada em base64 e os tipos de requisições.

A Figura 3 mostra um exemplo de JSON de requisição do tipo e detecção de faces.



**Figura 3. Exemplo de JSON de uma requisição**

```
[
  "requests": [
    {
      "image": {
        "content": "/9j/7QBUEGhv...string da imagem em base64...xxzj/Coa6Bax//Z"
      },
      "features": [
        {
          "type": "LABEL_DETECTION",
          "maxResults": 5
        }
      ]
    }
  ]
]
```

O campo *content* é onde é colocada a *string* da imagem em base64 e no campo *features* é onde são colocados os tipos de requisições, que podem ser dos tipos:

- LABEL\_DETECTION: Detecção de marcadores;
- FACE\_DETECTION: Detecção facial;
- LOGO\_DETECTION: Detecção de logotipos;
- IMAGE\_PROPERTIES: Atributos de imagens;
- TEXT\_DETECTION: Reconhecimento óptico de caracteres;
- DOCUMENT\_TEXT\_DETECTION: Reconhecimento óptico de caracteres;
- CROP\_HINTS: Dicas de corte da imagem;
- LANDMARK\_DETECTION: Detecção de pontos de referência;
- SAFE\_SEARCH\_DETECTION: Detecção de conteúdo explícito;
- WEB\_DETECTION: Detecção na Web;

No campo *maxResults* é onde é definido o número máximo de resultados que a API deve retornar, a omissão deste campo faz com que não haja limite.

Além do JSON para fazer uma requisição para a *Cloud Vision API* é necessário passar a chave da API como o valor de um parâmetro.

A Figura 4 mostra um exemplo de como passar a chave da API como parâmetro em uma requisição.



**Figura 4. Exemplo de uma requisição**

```
POST https://vision.googleapis.com/v1/images:annotate?key="CHAVE da API"
```

### **E. Tratamento do retorno da Cloud Vision API**

Após o envio da requisição para a *Cloud Vision API*, recebemos um JSON como retorno, contendo as informações que foram solicitadas, após o recebimento deste arquivo JSON, o aplicativo quebra as informações contidas no arquivo em pequenas informações e separa apenas as informações relevantes para a solicitação do usuário. Utilizando como exemplo o arquivo JSON da Figura 5, os dados relevantes vão ser os dados contidos nos campos *description*, que indicam características do objeto, e o campo *score* que indica o grau de certeza que a API tem de que aquela realmente é uma característica do objeto, quanto mais próximo de 1 esse valor for, então mais certeza a API tem.

A Figura 5 mostra um exemplo do arquivo JSON de retorno de uma requisição de detecção de características.

**Figura 5. Exemplo de retorno e uma requisição de detecção de características**

```
{
  "labelAnnotations": [
    {
      "mid": "/m/020lf",
      "description": "mouse",
      "score": 0.94139785,
      "topicality": 0.94139785
    },
    {
      "mid": "/m/07c1v",
      "description": "technology",
      "score": 0.9267815,
      "topicality": 0.9267815
    },
    {
      "mid": "/m/0bs7_0t",
      "description": "electronic device",
      "score": 0.89523095,
      "topicality": 0.89523095
    },
    {
      "mid": "/m/0h8lprf",
      "description": "computer component",
      "score": 0.88192433,
      "topicality": 0.88192433
    },
    {
      "mid": "/m/02dwgb",
      "description": "input device",
      "score": 0.6739148,
      "topicality": 0.6739148
    }
  ]
}
```



## **F. Tradução do retorno da Cloud Vision API**

Depois do processo de tratamento do retorno da Cloud Vision API, o aplicativo utiliza a Translation API para traduzir os dados tratados para a língua padrão do dispositivo do usuário. Utilizando como exemplo o arquivo JSON da Figura 5, como é uma requisição de características da foto de um *mouse* de computador, é possível extrair informações como: *mouse*, *technology*, *electronic device*, *computer component* e *input device* então o aplicativo iria traduzir para rato, tecnologia, aparelho eletrônico, componente de computador e dispositivo de entrada respectivamente.

## **G. Formulação da resposta para o usuário**

Depois do processo de tradução dos dados, o aplicativo vai utilizar os dados foram traduzidos para formular uma resposta para ser apresentada para o usuário. Utilizando como exemplo o arquivo JSON da Figura 5, o seu tratamento e a sua tradução, onde foram extraídos os dados *mouse*, *technology*, *electronic device*, *computer component* e *input device* então o aplicativo iria traduzir para rato, tecnologia, aparelho eletrônico, componente de computador e dispositivo de entrada respectivamente, o aplicativo vai criar a seguinte frase, "Muito provavelmente seja um rato". Então o usuário vai poder através de um determinado comando de voz, pedir mais características, e o aplicativo vai retornar algo do tipo: "encontrei estas características: tecnologia, aparelho eletrônico, componente de computador, dispositivo de entrada. "

O *score* de cada característica que foi extraída do JSON, é o que possibilita o aplicativo formular a frase "Muito provavelmente seja um rato", pois no exemplo da Figura 5, o *score* do dado *mouse* foi acima de 0,9 que é muito próximo de 1, isto significa que a API tem muita certeza de que aquilo é um *mouse*, caso o valor fosse abaixo de 0,6 então significaria que a API não tem tanta certeza. As demais características servem para ajudar o usuário a ter um maior grau de certeza do que se trata o objeto, no exemplo acima, inicialmente o usuário iria imaginar que se tratava de um rato (animal), porém com as outras características iria perceber que se trata de um *mouse* de computador, esse é um ponto muito positivo do projeto, pois os demais aplicativos no mercado, só exibem uma resposta, e em casos como esse o usuário pode ficar confuso.

## **H. Apresentação da resposta para o usuário**

Depois do processo de formulação da resposta para o usuário, o aplicativo vai fazer a apresentação da resposta para o usuário na forma de áudio, o aplicativo vai pegar a frase que foi formulada no processo anterior e reproduzir na forma de áudio através da API TTS. Após este processo ser finalizado, o aplicativo retorna para o estágio inicial.

## **4 PROBLEMA**

O projeto encontra-se ainda em fase de desenvolvimento e aprimoramento, para testes iniciais foi construído um protótipo *Minimum Viable Product* (MVP), o protótipo MVP se mostrou muito promissor



nos testes realizados, em todos os testes realizados o protótipo MVP teve sucesso em dizer corretamente características dos objetos testados, inclusive em fotos com mais de um objeto. Foram realizados diversos testes e em diversas situações, os testes foram realizados com diversos objetos de diferentes tipos, como por exemplo, *mouse*, teclado, televisão, pacote de espaguete, pacote de chocolate, geladeira, cadeira, mesa, pacote de extrato de tomate, cédulas de real, animais, notebook, entre outros. Em todos os testes realizados o protótipo conseguiu dizer corretamente as características dos objetos testados, para fins de testes o protótipo MVP foi criado com interface gráfica, para que possa haver um maior controle sobre os testes e uma maior agilidade, porém o projeto final não vai possuir interface gráfica, ele será totalmente comandado via comandos de voz, para que possa haver uma interação mais fluida entre o aplicativo e o usuário. O aplicativo vai ter apenas uma tela preta e vai setar o brilho do *display* do *smartphone* do usuário no nível mínimo, para que assim possa haver uma maior autonomia no consumo de energia do aplicativo no *smartphone* do usuário.

O projeto se diferencia em muitos aspectos das soluções que existem atualmente no mercado, um dos principais aspectos positivos em que o projeto se diferencia é o fato de ser totalmente gratuito, outro aspecto positivo em que o projeto se diferencia é o fato de ser de fácil acesso, visto que atualmente a grande parte dos deficientes visuais possuem *smartphones* com sistema operacional Android e conexão com internet, que são os únicos requisitos para que um deficiente visual possa utilizar o aplicativo, além do fato do aplicativo ser de fácil acesso, outro ponto em que ele se destaca é no fato de que ele promove uma maior autonomia e independência para os deficientes visuais, visto que ele não vai precisar do auxílio de outra pessoa e vai poder interagir facilmente com o aplicativo via comandos de voz.

## 5 CONCLUSÕES E TRABALHOS FUTUROS

O trabalho alcançou resultados interessantes, mas para o desenvolvimento de um sistema robusto é necessário continuar o desenvolvimento do mesmo. Conforme já citado pretendesse estender o aplicativo para reconhecimento de outras características. Pretende-se também como já foi citado, realizar a remoção da interface gráfica e implementar os comandos de voz. Com o intuito de atingir o objetivo do presente projeto, forem definidas as seguintes etapas a serem realizadas:

- **Atividade 1:** Adicionar a funcionalidade de reconhecimento de mais tipos de características.
- **Atividade 2:** Implementação do reconhecimento dos comandos de voz.
- **Atividade 3:** Adicionar outras línguas.
- **Atividade 4:** Remoção da interface gráfica.

## 6 REFERÊNCIAS

[1] L. M. B. Oliveira, Cartilha Do Censo 2010 Pessoas com Deficiência, 1st ed. Brasília: Secretaria Nacional de Promoção dos Direitos da Pessoa com Deficiência, 2012, p. 32.



- [2] F. A. S. Alves, A. M. M. Neumann, and M. M. G. Jr, "Bengala inteligente neural baseada em aprendizagem por reforço para deficientes visuais. " São Carlos: Encontro Nacional de Inteligência Artificial e Computacional, 2014, p. 6.
- [3] R. Bersch, "Introdução à tecnologia assistiva. " Porto Alegre: Assistiva - Tecnologia e Educação, 2008, p. 20.
- [4] M. E. R. F. Gasparetto, R. de Cassia Ietto Montilha, S. M. C. de Paula Arruda, J. Sperque, T. L. de Azevedo, and M. I. R. de Souza Nobre, "Utilização de recursos de tecnologia assistiva por escolares com deficiência visual. " Porto Alegre: INFORMÁTICA NA EDUCAÇÃO: teoria prática, 2012, vol. 15, p. 18.
- [5] **LookTel Money Reader.** Disponível em: <http://www.looktel.com/moneyreader>. Acesso em: junho 2018.
- [6] **LookTel Recognizer.** Disponível em: <http://www.looktel.com/recognizer>. Acesso em: junho 2018.
- [7] M. Tozzi and J. J. Ota, "Vertedouro em degraus. " Curitiba: Revista da Vinci, 2004, vol. 1, pp. 9–28.
- [8] **Blindtool.** Disponível em: <https://play.google.com/store/apps/details?id=the.blindtool>. Acesso em: junho 2018.
- [9] **Aipoly Vision.** Disponível em: <https://play.google.com/store/apps/details?id=com.aipoly.vision>. Acesso em: junho 2018.
- [10] **TapTapSee.** Disponível em: <https://taptapseeapp.com/>. Acesso em: junho 2018.
- [11] **BlindSight.** Disponível em: <https://play.google.com/store/apps/details?id=com.neuroxlabs.BlindSight>. Acesso em: junho 2018.
- [12] **Toolwiz Eyes-Voice Guide Cam.** Disponível em: <https://play.google.com/store/apps/details?id=com.btows.guidecamera>. Acesso em: junho 2018.
- [13] **Be My Eyes.** Disponível em: <https://www.bemyeyes.com/>. Acesso em: junho 2018.
- [14] OrCam MyEye. Disponível em: <https://www.maisautonomia.com.br/>. Acesso em: junho 2018.