UNIVERSIDADE DE SÃO PAULO

ESCOLA DE ENGENHARIA DE SÃO CARLOS

DEPARTAMENTO DE ENGENHARIA ELÉTRICA E DE COMPUTAÇÃO

SISTEMA ELETRÔNICO BASEADO EM ANDROID E ARDUÍNO PARA AUXÍLIO A PESSOAS COM DEFICIÊNCIA VISUAL

Autor: Guilherme Galdino Siqueira

Orientador: Prof. Dr. Evandro Luis Linhari Rodrigues

São Carlos

2016

Guilherme Galdino Siqueira

SISTEMA ELETRÔNICO BASEADO EM ANDROID E ARDUÍNO PARA AUXÍLIO A PESSOAS COM DEFICIÊNCIA VISUAL

Trabalho de Conclusão de Curso apresentado à Escola de Engenharia de São Carlos, da Universidade de São Paulo

Curso de Engenharia de Computação

ORIENTADOR: Prof. Dr. Evandro Luis Linhari Rodrigues

São Carlos

2016

Galdino Siqueira, Guilherme
G149s SISTEMA ELETRÔNICO BASEADO EM ANDROID E ARDUÍNO PARA AUXÍLIO A PESSOAS COM DEFICIÊNCIA VISUAL / Guilherme Galdino Siqueira; orientador Evandro Luis Linhari Rodrigues. São Carlos, 2016.

Monografia (Graduação em Engenharia de Computação) -- Escola de Engenharia de São Carlos da Universidade de São Paulo, 2016.

1. Android. 2. Arduíno. 3. visão computacional. 4. detecção de obstáculos. 5. deficiência visual. I. Título.

FOLHA DE APROVAÇÃO

Nome: Guilherme Galdino Siqueira

Título: "Sistema eletrônico baseado em Android e Arduíno para auxílio a pessoas com deficiência visual"

Trabalho de Conclusão de Curso defendido em 11/11/2016.

Comissão Julgadora:	**Resultado:**
Prof. Associado Evandro Luis Linhari Rodrigues (Orientador) - SEL/EESC/USP	APROVADO
Prof. Titular Alberto Cliquet Júnior SEL/EESC/USP	APROVADO
Profa. Associada Simone do Rocio Senger de Souza SSC/ICMC	Aprovada

Coordenador do Curso Interunidades Engenharia de Computação pela EESC:

Prof. Dr. Maximiliam Luppe

Agradecimentos

Agradeço primeiramente a Deus por todas as vitórias que obtive antes e durante o período que estive na universidade. Aos meus pais, Marcelo e Angélica, e meu irmão Gabriel por toda paciência e todo apoio que me deram em meio as dificuldades e desafios que enfrentei na graduação. À Universidade de São Paulo, por abrir diversas portas para minha formação, como a oportunidade de estudar no exterior onde começou a se formar a ideia desse projeto, e por oferecer a base do meu conhecimento. Agradeço de modo muito especial meus amigos de turma, com os quais convivi diariamente, e que me deram suporte nos estudos e trabalhos. Ao grupo do Facebook "Cegos e a Tecnologia", pela atenção dada em alguns momentos de dúvida durante o desenvolvimento do projeto. Agradeço ao professor Evandro pela confiança em meu trabalho e por toda orientação que me deu no desenvolver desse projeto e, enfim, aos poucos professores que tive que colocaram a preocupação com o aprendizado dos alunos acima de tudo em suas avaliações.

Guilherme Galdino Siqueira

Um homem sozinho não desenvolve qualquer poder intelectual. É necessário que ele esteja imerso em um ambiente de outros homens, cujas técnicas ele absorve durante os primeiros vinte anos de sua vida. Ele pode então talvez realizar alguma pesquisa de sua autoria e fazer pouquíssimas descobertas que são passadas para outro. Desse ponto de vista, a procura por novas técnicas deve ser tratada como realizada pela comunidade humana como um todo, não apenas por indivíduos.

Alan Turing

Resumo

Este projeto consistiu no desenvolvimento de um sistema composto por um aplicativo para smartphone baseado em Android e uma placa Arduíno, visando contribuir com a acessibilidade de pessoas sob diferentes níveis de deficiência visual. O sistema oferece ao usuário descrições simples de imagens capturadas pela câmera do smartphone e oferece suporte de detecção de obstáculos. O reconhecimento de imagens foi plenamente realizado pela API Google Cloud Vision, que mostrou considerável precisão. A utilização da API permitiu ao projeto seguir a tendência de solicitação de serviços em nuvem, porém tornou o sistema dependente da conexão com a internet. A detecção de obstáculos por sua vez exigiu a criação de um circuito com a placa Arduíno conectada a um sensor de obstáculos e um módulo Bluetooth. A funcionalidade de detecção se mostrou eficiente por abranger distâncias de até 4 metros e alertar o usuário por sentidos não visuais. De modo geral, o sistema mostrou ter boa utilidade e usabilidade.

Palavras-Chave: Android, Arduíno, visão computacional, detecção de obstáculos, deficiência visual.

Abstract

This project was the development of a system composed of an Android based app and an Arduino board, to contribute to the accessibility of people with different levels of visual impairment. The system provides the user with simple descriptions of images captured by smartphone camera and provides obstacle detection support. The image recognition was totally performed by Google Cloud Vision API, which showed considerable accuracy. The use of the API allowed the project to follow the trend of cloud services request, but made the system dependent on internet connection. The detection of obstacles in turn required the creation of a circuit with the Arduino board connected to an obstacle detection sensor and a Bluetooth module. The detection feature was efficient for covering distances of up to 4 meters and alert the user by nonvisual senses. In general, the system proved to have good utility and usability.

Keywords: Android, Arduino, computer vision, obstacles detection, visual impairment.

Lista de Figuras

Lista de Tabelas

Siglas

API	*Application Programming Interface* - Interface de Programação de Aplicativos
OCR	*Optical Character Recognition* - Reconhecimento Ótico de Caracter
TTS	*Text-To-Speech* - Texto-Para-Fala
IoT	*Internet of Things* - Internet das Coisas
IDE	*Integrated Development Environment* - Ambiente de Desenvolvimento Integrado
ETA	*Electronic Travel Aid* - Subsídio Eletrônico de Percurso
SaaS	*Software as a Service* - Software como Serviço

Sumário

1 Introdução

Uma pesquisa publicada pelo Banco Mundial em 2016 procurou avaliar simultaneamente os avanços tecnológicos e a exclusão digital, principalmente com relação ao acesso à internet. Seus resultados mostraram que alguns países estão desvinculados da revolução digital, o que reflete na exclusão socioeconômica de parcela da população do planeta [1]. Esse capítulo terá a intenção de introduzir o leitor ao cenário tecnológico atual, a situação vivida por pessoas com deficiência visual, apresentará o estado da arte, por meio de projetos e irá por fim propor um sistema para aplicar tais tecnologias a esse público alvo.

1.1 Motivação

No primeiro trimestre de 2016, a aquisição mundial de dispositivos móveis ultrapassou a marca de 7 bilhões, e esse número permanece em contínuo crescimento ano a ano. No mesmo período, foi identificado também que 80% de todos os dispositivos móveis são compostos por smartphones e que o número dobrará até 2021. A área de desenvolvimento de aplicações móveis se mostra num momento promissor devido não só à quantidade expressiva de dispositivos, que inclusive já ultrapassa a população de alguns países, mas também devido ao recente surgimento da Internet das Coisas, IoT [2].

Apesar de todo esse progresso, no entanto, muitas pessoas são deixadas de lado por não terem acesso a esse mundo de possibilidades oferecido pela tecnologia digital. Pessoas com deficiência, por exemplo, ainda enfrentam obstáculos para comunicação, interação e acesso à informação. A tecnologia atual é capaz de promover diversos meios alternativos de comunicação desde reconhecimento de voz até interfaces controladas por gestos. Mas a simples existência de tecnologia ainda não é suficiente para preencher a lacuna de inclusão socioeconômica dessas pessoas [1].

A realização desse projeto é, assim, guiada pelo propósito de tentar direcionar os benefícios de uma área promissora de trabalho, e com inúmeras possibilidades de atuação, a um cenário com menor visibilidade, que é o da criação de sistemas para o auxílio a pessoas com deficiência, em especial, as visuais.

1.2 Estado da arte

Para o desenvolvimento do projeto, alguns projetos foram utilizados como base. Assim, a proposta procurou seguir a linha de produtos bem sucedidos no mercado na área de auxílio a pessoas com deficiência visual.

1.2.1 Projetos de descrição de imagens

Na área de leitura e extração de informações de imagens, um exemplo que possui funcionalidades próximas às propostas e que serviram de inspiração é o aplicativo TapTapSee, que fotografa objetos e os identifica em voz alta, além de estar vinculado com contas de usuário, como Facebook e Twitter para compartilhamento[3].

Existe também o aplicativo móvel Be My Eyes[4], um sistema de auxílio em rede que conecta usuários cegos a voluntários por meio de vídeo chamadas e áudio. Mauro Avila et al.[5] fazem uma avaliação do aplicativo que consistiu em uma pesquisa com 15 homens e 15 mulheres através de mídias sociais. A maioria dos entrevistados tinha entre 36 e 65 anos de idade. O trabalho concluiu que os usuários consideram o aplicativo útil para leitura de textos, localização de objetos, assistência a compras, entre outras tarefas do dia a dia.

Outro trabalho interessante é o realizado pela Universidade Hamad Bin Khalifa. H. Kwak e J. An[6] analisaram mais de 2 milhões de fotos de jornais publicadas em Janeiro de 2016 por meio da API Google Cloud Vision. A pesquisa avaliou a frequência de exibição e expressões faciais em fotos dos então candidatos à presidência dos Estados Unidos, nos principais jornais, e concluiu que a API foi o ponto chave de sucesso do trabalho devido à alta acurácia e pontuações de confiabilidade de cada resultado.

1.2.2 Projetos de detecção de obstáculos

Na área de detecção de obstáculos, existem diversos projetos ETAs, Eletrônic Travel Aid, que utilizam sensores ultrassônicos. Wong et al.[7] na intenção de evitar maus hábitos de uso da bengala e contornar suas limitações, propôs um sistema composto por sensores que continuamente procuram por objetos também em níveis de alturas distintos. O sistema utiliza um microcontrolador para calcular a distância baseada em sinais ultrassônicos emitidos e recebidos por dois transdutores, conversores de energia mecânica em elétrica fixados na bengala, um para detecção em pequenas alturas, e o outro para grandes. Então, depois de emitir os sinais,

captura sua reflexão, calcula a distância e gera um sinal sonoro de retorno ao usuário.

Alternativamente, Ben Leduc-Mills et al[8]. apresenta um projeto mais recente envolvendo uma placa IOIO, baseada em PIC, que permite que aplicações Android interajam com dispositivos eletrônicos externos. A placa recebe os sinais de sensores ultrassônicos e os envia ao aplicativo Android via Bluetooth. Então o aplicativo fica responsável por tratar o sinal e alertar o usuário se há algum objeto em um limite mínimo de distância e a que altura está. O alerta por fim se dá via audição e tato.

1.2.3 Cenário no Brasil

No Brasil também há diversos trabalhos nesse sentido. O CPqD Alcance, por exemplo, é um aplicativo que adapta a grande maioria das funcionalidades de um celular em uma interface de maior usabilidade destinado não só para pessoas com deficiência visual, mas também para idosos e pessoas iletradas[9].

Na Universidade de São Paulo, dois projetos merecem destaque. O primeiro é o projeto GuideMe[10], coordenado pelo professor do ICMC-USP Francisco Monaco, que consiste de um pequeno dispositivo ajustável à roupa contendo uma câmera e sensores de ultrassom para identificar rostos de pessoas conhecidas, por meio de algorítimos de processamento de imagem, e conhecimento de obstáculos em percurso, que para tanto, também emite sons cuja intensidade é baseada na direção e proximidade em relação aos obstáculos.

O segundo é o projeto de conclusão de curso de Murilo A. Gallani[11], que apresenta um dispositivo eletrônico de aprimoramento da orientação proporcionada por bengalas à pessoas com deficiência visual. O dispositivo, que fica acoplado à ponta de uma bengala, utiliza seu movimento para fazer uma varredura de possíveis obstáculos, e alerta o usuário por meio de sons, cujos volumes simulam sua emissão pelo obstáculo, e auxiliam na localização espacial.

Nessa linha de pesquisa há também os projetos Bengala Longa Eletrônica[12] e Bengala Automática para Deficientes Visuais[13], projetos universitários da Universidade do Vale do Itajaí e do Instituto Federal da Bahia, respectivamente, que de maneira similar acoplam a uma bengala sensores que auxiliam na identificação de obstáculos.

1.3 Objetivos

Baseado nos projetos apresentados, este projeto busca portanto desenvolver mais uma ferramenta alternativa para suprir algumas das necessidades enfrentadas por pessoas com

deficiência visual, e basicamente propõe-se a promover ao usuário as seguintes habilidades:

- Conhecimento de conteúdo visual escrito.
- Conhecimento de características do ambiente ao redor.
- Conhecimento de possíveis obstáculos pelo caminho.
- Maior independência e autonomia.

A Figura 1.1 apresenta superficialmente, e de maneira ilustrativa, o que se obteve do sistema proposto quanto a extração de informações de imagens capturadas. Basicamente, uma aplicação móvel será capaz, ao capturar uma imagem, de analisar seu conteúdo. Textos inseridos em rótulos de produtos ou bulas de remédio, e características faciais ou de paisagens poderiam ser total ou parcialmente compreendidos e com independência do auxilio de uma pessoa com visão normal, por meio da audiodescrição da informação.

Figura 1.1: Aplicativo - Extração de texto e características faciais, respectivos resultados e saída sonora para o usuário

Já a Figura 1.2 esquematiza o funcionamento do sistema desenvolvido no cenário de uma detecção de obstáculos durante uma caminhada. Com ele seria possível aumentar a eficiência da bengala, ferramenta bastante utilizada por deficientes visuais e que retorna muitas informações do ambiente em um percurso. Como obstáculos fora do chão nem sempre são percebidos pela bengala, o detector de obstáculos poderia servir de auxílio em situações como a ilustrada, em que uma bengala pode até tocar o suporte do telefone público, mas seria possível que a pessoa chegasse perto demais, e ocorresse uma colisão entre a cobertura superior e a cabeça.

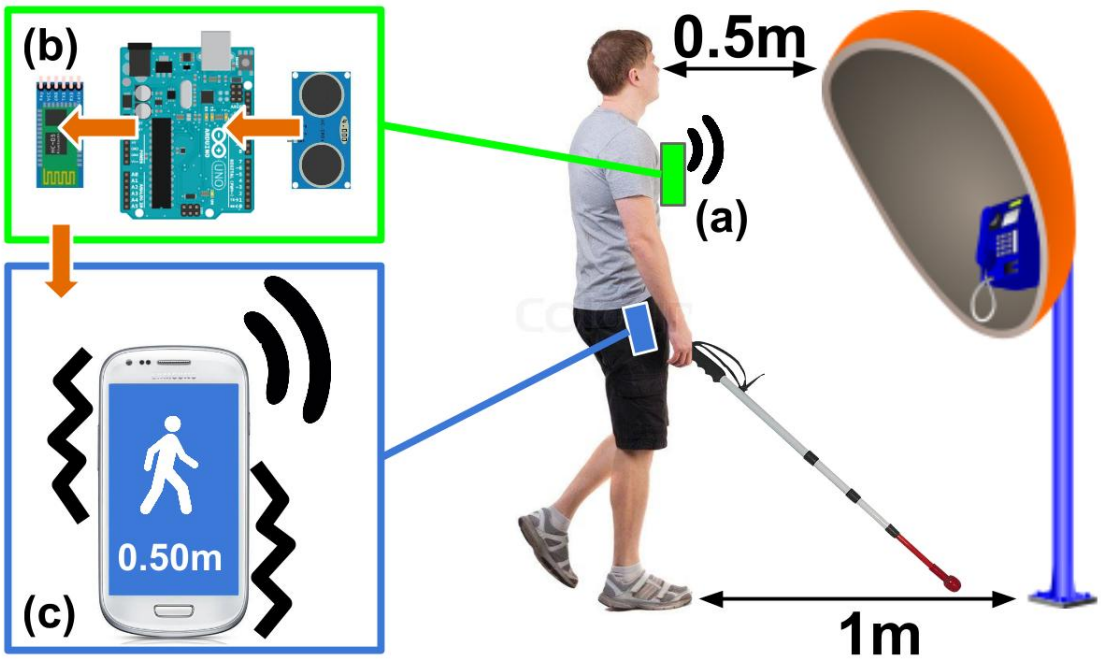


Figura 1.2: Aplicativo e circuito externo - (a) Geração e captura de sinal ultrassônico, (b) Tratamento do sinal e reenvio para o smartphone, (c) Notificação ao usuário por vibração e som

1.4 Justificativas

Em 2015 o IBGE divulgou dados da Pesquisa Nacional da Saúde[14] que demostra a proporção de brasileiros portadores das seguintes deficiências: auditiva, visual, física e intelectual. Segundo o levantamento, dentre os tipos de deficiência analisados, a visual é a que mais afeta os brasileiros, mais de 3% da população. A pesquisa também mostra que 11% desse grupo é composto por pessoas acima de 60 anos, que quase 7% utilizam algum recurso de locomoção, como bengala ou cão guia, e que o grau intenso de deficiência atinge 16% e resulta na impossibilidade de o indivíduo realizar tarefas básicas, como trabalhar.

Adicionalmente do ponto de vista global, segundo a HDR, Human Development Resour-

ces, o Brasil ocupa a posição 75 no índice de desenvolvimento humano [15]. De fato, o país historicamente apresenta algumas dificuldades em promover o bem estar social e a igualdade da população.

Diante desse cenário, foi possível perceber que há uma parcela significativa da população que necessita de auxílio específico para suprir a falta de visão para realizar as mesmas atividades de quem possui visão normal. Assim, o desafio desse projeto foi o de desenvolver um sistema que fosse capaz de amenizar as necessidades dessa parcela da população.

Espera-se que com isso seja possível contribuir minimamente com a qualidade de vida a nível pessoal de parte da sociedade que possui necessidades especiais e ao mesmo tempo permitir o desempenho da função de engenheiro na sociedade, que é o de colocar o conhecimento científico a serviço do conforto e desenvolvimento da humanidade.

1.5 Organização do trabalho

Este trabalho está distribuído em 5 capítulos, incluindo esta introdução, dispostos conforme a descrição que segue:

- Capítulo 2: Descreve o embasamento teórico sobre o qual o projeto foi desenvolvido, definindo conceitos e proporcionando explicações necessárias para a compreensão do desenvolvimento do trabalho.

- Capítulo 3: Destina-se ao planejamento do sistema proposto, listando os requisitos extraídos, que deverão ser considerados, e o planejamento de testes, para garantir seu correto funcionamento.

- Capítulo 4: Discorre sobre os materiais e métodos utilizados no andamento do projeto, explicando as características de cada um dos dispositivos eletrônicos, a razão de sua utilização, e o modo como as partes do projeto se conectam entre si.

- Capítulo 5: Apresenta os resultados obtidos por meio de teste sobre o sistema, e faz uma análise a fim de explicá-los.

- Capítulo 6: Resume os principais pontos de todo o processo de desenvolvimento até a finalização do projeto, apresentando a importância da solução proposta e os problemas encontrados.

2 Embasamento Teórico

Antes de entender o funcionamento do sistema, é de extrema importância que se explique os principais conceitos que dão base a criação do projeto, a fim de que a leitura não se limite apenas a fornecer conhecimento funcional, mas também propiciar uma completa compreensão estrutural do sistema.

2.1 Android

O Android é um sistema operacional desenvolvido pela Android Inc., que posteriormente foi adquirido pela Google. De acordo com Paul Deitel et al[16], do lançamento da primeira geração em 2008 até 2015, o Android já possuía mais de 80% de participação no mercado global. Atualmente, ele pode ser encontrado em smartphones, *tablets*, *e-readers*, robôs, equipamentos eletrônicos domésticos e até em satélites da NASA.

Os aplicativos Android são desenvolvidos em Java, uma das linguagens mais utilizadas no mundo. Além disso, seu código fonte é livre, o que significa que é grande a velocidade com que surgem inovações e melhorias.

2.2 Arduíno

O projeto Arduíno[17] teve origem na Itália, em 2005, num momento em que se procurava encontrar uma maneira barata de fazer com que estudantes de arte e *design* pudessem trabalhar com tecnologia.

As placas, baseadas no microprocessador de 8 bits da Atmel, possuem 14 pinos digitais que podem ser definidos como entrada ou saída, e seis deles podem ser programados para produzir saídas com modulação de largura de pulso. O Arduíno também possui vários protocolos de comunicação, como serial e o bus-serial de interface periférica.

Com o Arduíno é possível produzir os mais variados tipos de projetos, desde jogos de bolinhas, com gráfico monocromático e efeitos de som simples, até robôs seguidores de linha e robôs auto-balanceados.

2.3 Tecnologia assistiva

O Comitê de Ajudas Técnicas[18], instituído pela PORTARIA N° 142, DE 16 DE NOVEMBRO DE 2006 por meio da Secretaria Especial dos Direitos Humanos, propôs a seguinte definição para o termo Tecnologia Assistiva: "Tecnologia Assistiva é uma área do conhecimento, de característica interdisciplinar, que engloba produtos, recursos, metodologias, estratégias, práticas e serviços que objetivam promover a funcionalidade, relacionada à atividade e participação, de pessoas com deficiência, incapacidades ou mobilidade reduzida, visando sua autonomia, independência, qualidade de vida e inclusão social."

De forma resumida, a Tecnologia Assistiva consiste em todo o conjunto de ferramentas tecnológicas que visam promover a melhoria da qualidade de vida de pessoas com alguma limitação.

2.4 Design universal e acessibilidade

Design universal é definido por S. L. Henry et al[19] como o processo de criação de produtos que atendam a usabilidade de pessoas com as mais variadas habilidades e nas mais diversas situações. Em contra partida, o conceito de acessibilidade é mais limitado, sendo melhor definido como o planejamento voltado especificamente para pessoas com alguma deficiência. Mas apesar dessa limitação, todos os estudos focados em acessibilidade terminam por trazer benefícios para todas as pessoas.

Especificamente em relação a dispositivos móveis, ferramentas que permitem a criação de sistemas com acessibilidade tem se mostrado em grande ascensão no ambiente de desenvolvedores de aplicativos. O Android, por exemplo, inclui ferramentas e serviços de auxílio a navegação, como text-to-speech, feedback tátil, navegação por gestos, entre outros, que buscam incluir usuários com limitações visuais, auditivas, físicas ou mesmo relacionadas a idade[20].

2.4.1 Talkback

O Talkback[16] é o leitor de tela fornecido pelo Android que funciona como um recurso de acessibilidade cuja função é permitir que deficientes visuais sejam capazes de utilizar um smartphone. Além de pronunciar todo tipo de texto presente na tela, ele também altera a lógica de toques e permite a descrição dos componentes presentes na tela.

Quando o Talkback está ativado, um clique sobre qualquer item da tela funciona como uma

solicitação de descrição. O dispositivo vibra, e são pronunciados o conteúdo de acessibilidade inscrito no componente visual e todo texto exibido na tela.

2.5 Computação em nuvem

De acordo com a definição de Michael Armbrust et al[21], computação em nuvem se refere tanto a aplicações retornadas como serviço pela Internet, como ao hardware e software dos sistemas nos data centers que proporcionam os serviços. Os serviços são oferecidos por software (SaaS), e o conjunto hardware mais software dos data centers dão origem à nuvem. O serviço vendido por uma nuvem pública é denominado Computação Utilitária, e sua união com os SaaS é, portanto, o que se conhece como Computação em Nuvem. De maneira simplificada, o funcionamento da computação em nuvem pode ser facilmente compreendido pela Figura 2.1 que, inclusive, permite uma classificação do projeto, cujos métodos serão descritos na Seção 4.2: A Google pode ser vista na base do diagrama como a provedora de nuvem por oferecer a infraestrutura física necessária para o processamento de imagem, e também na etapa intermediária por ser uma provedora SaaS, oferecendo o Cloud Vision API e diversas outras aplicações. O aplicativo Android ficaria também no estágio intermediário, no papel de usuário da nuvem, por utilizar o serviço oferecido pela provedora. A pessoa que faz uso desse aplicativo seria, por fim, o usuário SaaS.

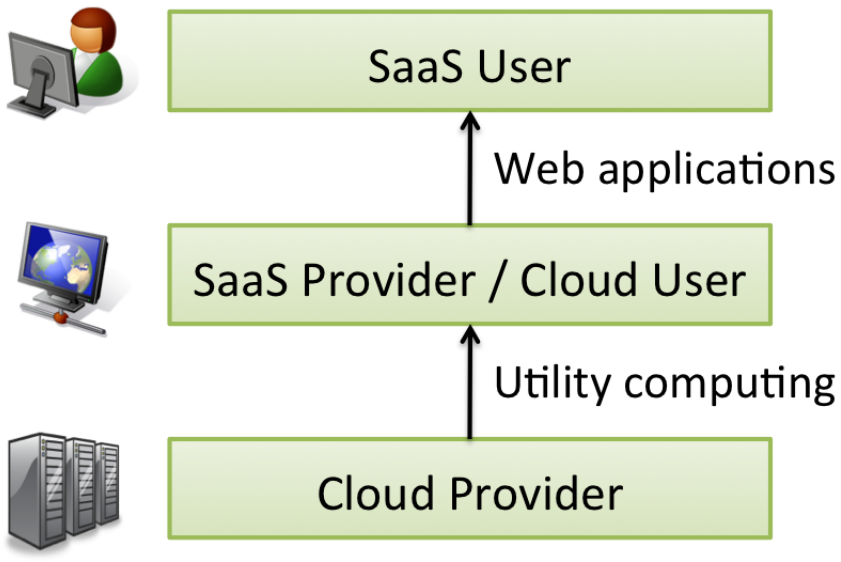


Figura 2.1: Diagrama da computação em nuvem

Fonte: Michael Armbrust et al [21]

2.6 Visão computacional

Visão computacional é o campo da computação responsável pela análise de imagens digitais com o objetivo da extração automática de informações. A informação pode ser tanto simples, como responder qual a cor da imagem, quanto dizer de quem é a face em uma foto [22]. No contexto desse projeto, dois pontos importantes devem ser compreendidos: o reconhecimento óptico de caracteres e a classificação de imagens, que são apresentados a seguir.

2.6.1 Reconhecimento óptico de caractere

Reconhecimento óptico de caractere (OCR) é considerado como o problema de reconhecimento automático de letras, dígitos, ou algum símbolo em imagens. A utilidade dessa abordagem está no fato de que muita informação é armazenada em palavras impressas. Ao aplicar uma página de texto, por exemplo, como entrada para um sistema que possua tal função, ocorre primeiramente uma confirmação da orientação do texto, seguida de uma segmentação em preto e branco dos pixels, uma divisão em linhas de texto, e por último em símbolos individuais. Ao fim desse processo, um algoritmo de reconhecimento é aplicado a cada símbolo. Se o sistema tiver sido previamente treinado para reconhecer tal símbolo, calcula-se uma probabilidade de sua interpretação estar correta, são agrupados em palavras e em sentenças e a informação completa é retornada em ordem [22].

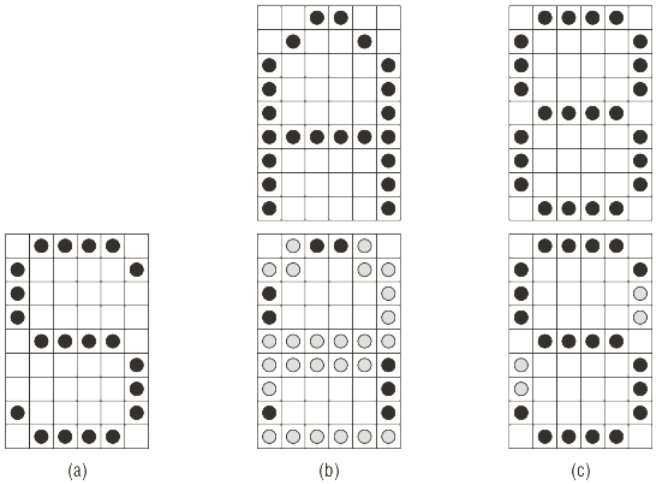


Figura 2.2: Análise classificatória de um simbolo sobre os demais da base de dados. (a) Simbolo de entrada. (b) Pixels coincidentes, em preto, entre o simbolo de entrada e o simbolo 'A'. (c) Pixels coincidentes, em preto, entre o simbolo de entrada e o simbolo '8'.

Fonte: Parker, J. R. [22]

Na Figura 2.2, um simbolo é enviado como entrada do sistema para ser comparado com os símbolos da base de dados. Nela, o simbolo apresenta maior similaridade com o símbolo ’8’ (coincidência de 20 pixels), do que com ’A’ (coincidência de 8 pixels), e sua classificação seria dada de acordo com o símbolo com o qual ele tivesse maior valor de semelhança, medido pelo número de pixels coincidentes.

2.6.2 Classificação de imagens

Para que um sistema seja capaz de classificar uma imagem, e posteriormente promover uma descrição, que é o caso desse projeto, é necessário que esse sistema esteja previamente treinado com imagens. Um processo como esse, na verdade envolve busca e comparação de imagens. J. R. Parker[22] sugere em seu livro um método para se realizar buscas de imagens, inserindo imagens como entrada.

Primeiramente, assume-se que existe um conjunto de imagens, previamente rotuladas e devidamente agrupadas de acordo com seu conteúdo. Então, o que ocorre em é uma análise computacional da imagem para extração de dados em busca de padrões, como explicado na secção 2.6.1, para o caso específico de OCR. Esses dados são comparados com os dados das imagens cujas características já são conhecidas e baseado em seu nível de similaridade, ocorre o agrupamento da imagem, que reflete seu conteúdo. A Figura 2.3 ilustra a tentativa de classificar um objeto por meio da medida de semelhança contra outras imagens da base de dados. Os mais similares são considerados iguais, apesar de o resultado não ser exatamente igual.

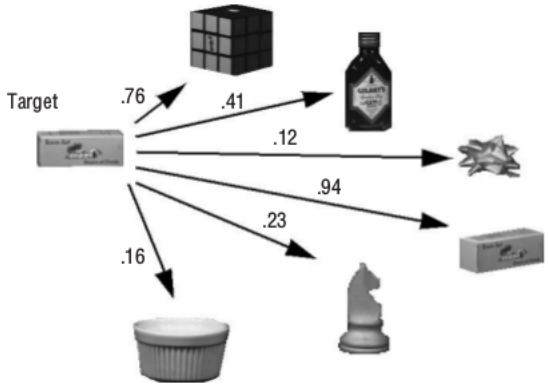


Figura 2.3: Classificação de imagem de acordo com sua similaridade em relação ao conjunto de imagens da base de dados.

Fonte: Parker, J. R. [22]

Uma pergunta que poderia surgir é: Que padrão poderia ser extraído de uma imagem para servir de critério de comparação? A cor é uma possibilidade. Através da contagem de pixels com cada cor presente na imagem é possível criar um histograma da distribuição dessas cores. E assim, a comparação poderia ser realizada sobre a similaridade entre histogramas.

2.7 Ecolocalização e localização sonora

A habilidade de se locomover independentemente pelo espaço, localizar lugares ocultos e planejar trajetórias é de extrema importância para se realizar as tarefas do cotidiano. Não é difícil encontrar razões para afirmar que essa capacidade, em pessoas, resulta em grande dependência do sentido visual, já que a quantidade de informações que podem ser captadas visualmente é consideravelmente maior que a dos outros sentidos. Os objetos com os quais se interage no dia-a-dia possuem partes visíveis, porém não necessariamente emitem outros sinais que possam permitir sua percepção não visual. Apesar de ser considerada uma medida mais imprecisa, sinais sonoros permitem uma aproximação do cálculo de distancia. O som varia sua intensidade ao se propagar de acordo com o inverso da distância até seu emissor, assim, sua intensidade se perde mais rapidamente, o que a torna um método limitado [23].

A localização sonora se baseia nesse efeito, ao permitir que um indivíduo estime a sua distância até o objeto emissor de som, e é uma técnica utilizada e bastante desenvolvida por pessoas com deficiência visual para mapear a sua posição e a dos elementos no ambiente ao redor. Entretanto, existe também uma técnica capaz de complementar a eficiência da localização conhecida como ecolocalização. Schenkman e Nilsson[24] afirmam, num estudo que compara pessoas com e sem deficiência visual sobre a capacidade de detectar sons refletidos, que as que têm deficiência possuem a audição mais acurada e podem ser capazes de utilizar do eco de sons emitidos intencionalmente para estimar a distância e o material de objetos e possivelmente ter o caminhar facilitado. Esse fenômeno que também é encontrado em outras espécies, como golfinho e morcego, por exemplo, podem ser gerados não só biologicamente pela voz, mas também por toques com sapatos ou bengalas.

3 Planejamento do Projeto

Antes de apresentar a implementação do sistema, é importante exibir seu planejamento. Apesar de o projeto não ter seguido estritamente uma metodologia de desenvolvimento, como bem sugere a Engenharia de Software para auxiliar em todo o processo, este capítulo, dividido em duas seções, será destinado a apresentar apenas a análise de requisitos e o planejamento de testes, que são parte importante do processo de desenvolvimento de qualquer projeto de software. Informações sobre decisões de projeto e a implementação podem ser encontradas no Capítulo 4, assim como os resultados dos testes são apresentados no Capítulo 5.

3.1 Análise de requisitos

Definir o que deve ser feito é a primeira tarefa, e a mais importante, durante o desenvolvimento de um sistema. Nesta seção serão apresentados os requisitos funcionais e não-funcionais.

Requisitos funcionais

- O sistema deve oferecer ao usuário as funcionalidades de extração de informações visuais, como textos impressos ou escritos a mão e características visuais de objetos, e do ambiente ao redor, como elementos de paisagens e expressões faciais.

- O sistema deve permitir o registro das informações extraídas e permitir que o usuário acesse esses registros.

- O sistema deve permitir que o usuário insira nos registros o nome das pessoas detectadas pelo sistema.

- O sistema deve auxiliar o usuário a caminhar com autonomia, sendo alertado por sentidos não visuais a localizar obstáculos pelo caminho.

- O sistema deve exibir textualmente as informações extraídas e permitir a alteração da fonte do texto.

- O sistema deve possibilitar a leitura em voz das informações extraídas para o usuário.

Requisitos não-funcionais

- O sistema deve garantir que pessoas com deficiência visual utilizem com facilidade o sistema.
- O sistema deve ser capaz de estabelecer conexão contínua com a internet e comunicação sem fio entre dispositivos.
- O sistema deve possuir saída de som e sintetização de voz.
- O sistema deve possuir entrada e saída de imagem, por meio de dispositivo de captura de imagem e display respectivamente. Além disso, deve permitir que o usuário tenha diferentes opções de resolução de captura.
- O sistema deve possuir uma fonte de energia que possibilite sua fácil troca ou recarga quando necessário.
- O sistema deve possuir uma fonte de energia que o atenda ininterruptamente durante no mínimo 2 horas.
- O sistema deve garantir a possibilidade de utilização de suas funções ininterruptamente, respeitando possíveis limites de fonte de energia.
- O sistema deve exibir os resultados da extração de informações com confiabilidade acima de 80%.
- O sistema não deve exigir que o usuário segure dispositivos com peso, tamanho e formato que tornem desconfortável seu uso.
- Ações atribuídas a interface do sistema devem ter execução iniciada em tempo real, definido como limite máximo de 2 segundos.
- O sistema deve ser capaz de armazenar dados como foto e texto.

3.2 Planejamento de testes

Com a intenção de apresentar dados de performance do sistema, foi considerado importante realizar os seguintes testes ao final da implementação dos módulos relacionadas às respectivas funções.

- Verificar a relação entre resolução de captura de imagens textuais e o tempo entre a solicitação e retorno do resultado, para definir a resolução adequada que retorna resultados corretos, que possuem erros inferiores a 5%, no menor tempo.

- Avaliar os resultados da extração de textos danificados, rotacionados ou escritos a mão em letra de forma e cursiva, assim como durante a extração de características de imagens não textuais para definir as limitações do sistema.

- Comparar o consumo de energia entre diferentes funcionalidades do sistema para evidenciar a relação entre número de recursos utilizados e consumo.

- Realizar sucessivas simulações de detecção de obstáculos para ao menos 5 distâncias distribuídas entre 0 e 3m para avaliar a performance da localização de obstáculos. Classificar como corretos resultados com erro inferior a 5cm de desvio padrão das amostragens.

- Testar o sistema sob diferentes versões da plataforma escolhida para implementação (sistema operacional e hardware) a fim de avaliar sua portabilidade.

- Percorrer todas as funcionalidades do sistema para verificar simultaneamente sua corretude funcional e o atendimento aos requisitos.

- Validar o sistema com potenciais usuários.

4 Materiais e Métodos

Mais do que uma simples aplicação Android, o projeto englobou o sensoreamento de sinais ultrassônicos, comunicação sem fio e um circuito gerenciado por um Arduíno. Por essa razão, diversos componentes eletrônicos e métodos de comunicação foram utilizados.

Basicamente, o usuário do smartphone, por meio do aplicativo Android captura uma imagem daquilo que deseja obter informações. A imagem é enviada aos servidores em nuvem da Google, onde é processada e tem suas informações traduzidas em palavras. O resultado é então transferido de volta para o smartphone e apresentado em forma textual apropriada sob a qual pode se obter uma audiodescrição.

Além disso, um sensor conectado a um Arduíno emite ondas ultrassônicas periodicamente e retorna a distância estimada ao obstáculo. O resultado é então transferido ao smartphone, por meio de um módulo Bluetooth também conectado ao Arduíno, e a aplicação por fim alerta o usuário. A Figura 4.1 apresenta um esquemático que exemplifica o funcionamento do sistema.

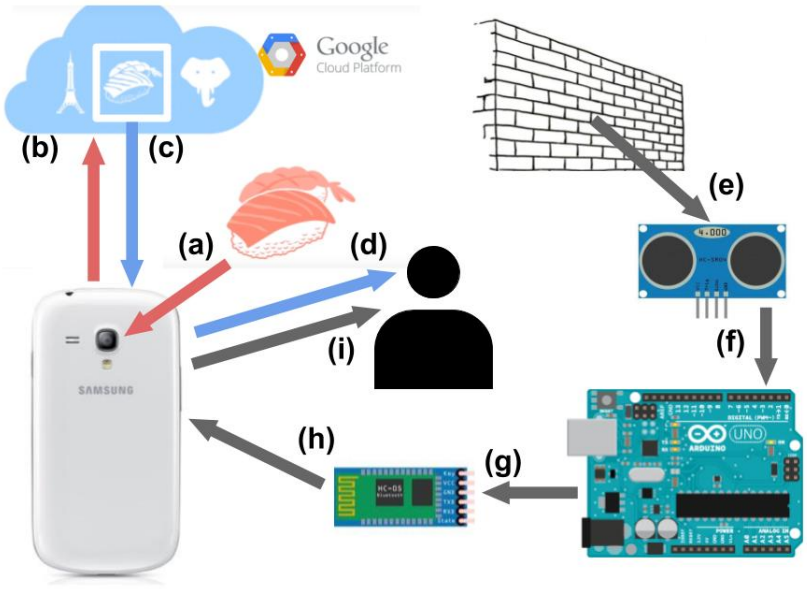


Figura 4.1: Esquemático de comunicação entre os módulos eletrônicos do projeto. (a) Captura da imagem. (b) Envio para processamento na nuvem. (c) Recebimento da descrição. (d) Exibição e audio-descrição. (e) Detecção de obstáculo. (f) Dados para cálculo da distância. (g) Distância enviada ao Módulo Bluetooth. (h) Distância enviada para o aplicativo. (i) Notificação do usuário.

4.1 Materiais

Os materiais utilizados, bem como a descrição detalhada de suas propriedades e sua função no projeto estão listados a seguir.

4.1.1 Smartphone

Figura 4.2: Smartphone Galaxy S3 Mini

Fonte: www.tudocelular.com

O nó principal do sistema pode ser considerado o smartphone. Por ser um dispositivo multifuncional, programável, com câmera, saída de áudio, vibração e permitir a execução de aplicações, além de possuir tamanho mais reduzido se comparado ao tablet, por exemplo, e ser um dispositivo sempre presente com as pessoas, mostrou-se ideal para o sistema proposto. Com a câmera foi possível a captura das imagens posteriormente tratadas para extração de informação. A saída de áudio em paralelo com a vibração foram essenciais para uma interface útil voltada para usuários com deficiência visual. O tamanho reduzido também foi importante para que o dispositivo pudesse ser manuseado e guardado facilmente no corpo. O smartphone utilizado majoritariamente durante o desenvolvimento do projeto devido sua disponibilidade foi o Samsumg Galaxy S3 mini, Figura 4.2, cuja especificação técnica está descrita na Tabela 4.1.

Tabela 4.1: Especificação técnica do smartphone utilizado no projeto

Sistema Operacional	Android 4.1 Jelly Bean
Dimensões	121.55 x 63 x 9.85 mm
Peso	111.5 g
RAM	1 GB
Memória	16 GB
Resolução - Câmera	2592 x 1944 pixels

4.1.2 Android Studio

Uma vez que o smartphone escolhido para o projeto foi o Galaxy S3 mini, cujo sistema operacional é o Android, para a programação do aplicativo foi necessária a utilização de uma IDE voltada para esse sistema. Como já havia uma maior experiência com a linguagem Java, a IDE escolhida foi Android Studio 1.4.

4.1.3 Arduíno UNO

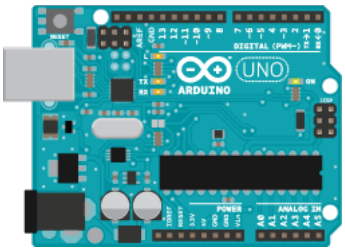


Figura 4.3: Arduíno UNO

Fonte: www.arduino.cc

Para a funcionalidade de detecção de obstáculos, que não poderia ser feita pelo smartphone, foi necessária a utilização de um dispositivo externo a ele. Devido sua disponibilidade para o projeto, foi definido que essa funcionalidade poderia ser facilmente realizada pela placa de programação Arduíno UNO, Figura 4.3. As especificações da placa estão na Tabela 4.2.

Tabela 4.2: Especificação técnica Arduíno

Microcontrolador	ATmega328P
Pinos de E/S	14
Dimensões	68.6 x 53.4 mm
Peso	25 g
Flash Memory	32 KB
SRAM	2 KB
EEPROM	1 KB
Clock	16 MHz
Tensão de Operação	5V

4.1.4 Sensor Ultrassônico de distância

Figura 4.4: Sensor HC-SR04.

Fonte: www.filipeflop.com

O Arduíno como uma placa programável, possibilita um infinidade de aplicações. Entretanto ele não possui sensores acoplados, apenas entradas e saídas genéricas. Para a detecção de obstáculos foi necessária a inserção de um sensor. O HC-SR04, Figura 4.5,é um sensor ultrassônico amplamente utilizado para esse propósito, e cujo alcance se mostrou apropriado para o projeto. Emitindo ondas de frequência ultrassônica, o sensor em seguida capta o sinal refletido e baseado no tempo entre emissão e retorno, e na velocidade do som, permite o cálculo da distância.

4.1.5 Módulo Bluetooth

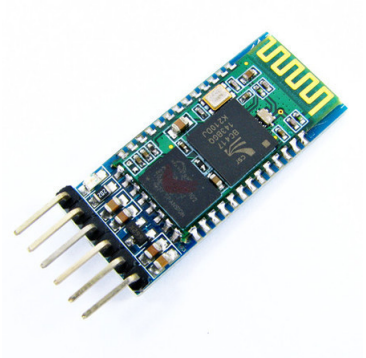


Figura 4.5: Módulo Bluetooth HC-05.

Fonte: www.filipeflop.com

Para a comunicação entre o Arduíno e o smartphone haviam diversas possibilidades de implementação. A troca de dados pela internet já seria utilizada pela aplicação Android para

o reconhecimento de imagens, que será apresentado com mais detalhes na Seção 4.2. No entanto, como o smartphone e o dispositivo detector de obstáculos estariam muito próximos do corpo do usuário, tornando a distância entre ambos os componentes relativamente pequena, e devido a simplicidade de implementação da comunicação, a escolha foi pelo módulo Bluetooth HC-05, Figura 4.5.

4.1.6 Demais componentes eletrônicos

Para a conexão dos componentes do circuito foram necessários uma variedade de fios para conectar VCC, GND, emendas e conexões de entrada e saída de sinais. Foram necessários também 3 resistores de 220Ω para criação de um divisor de tensão.

4.1.7 Caixa de integração do circuito

Para fins de demonstração do protótipo, foi criada uma caixa para envolver o circuito. Para tanto foram utilizados uma garrafa PET de 2L, tesoura, fita adesiva transparente, fita isolante, um interruptor tipo navio de dois pinos e uma bateria de 9V.

4.1.8 Dispositivos auxiliares

Para a programação do Arduíno utilizou-se um cabo USB. Para a medição de valores de tensão e de corrente foi utilizado um multímetro, e para a medição das distâncias retornadas pelo detector de obstáculos, utilizou-se uma trena.

4.2 Métodos

Os métodos utilizados para a se atingir o objetivo do projeto são extremamente importantes não só para a compreensão de como as partes se comunicam, mas também para se expor detalhes da implementação voltados para a garantia de ampla usabilidade e acessibilidade do usuário. Para auxiliar a compreensão da modelagem, as Figuras A.1 e B.1 nos Apêndices apresentam respectivamente um diagrama de classes compactado e um de fluxo referentes ao aplicativo.

4.2.1 Configurações iniciais de programação

Antes de tudo, para o início da programação do aplicativo foi necessária a instalação do Android Studio e para a programação da placa Arduíno, da IDE de mesmo nome.

4.2.2 Construção de interface acessível

Como o objetivo do projeto era permitir que pessoas com limitações visuais pudessem exercer algumas funções exclusivas de pessoas que podem ver, e o meio escolhido para esse fim foi uma aplicação para smartphone, o ponto mais importante, e primeiro a ser planejado foi a interface, para avaliar previamente se o Android 4.1 permitiria a usabilidade por pessoas às quais o projeto se destina.

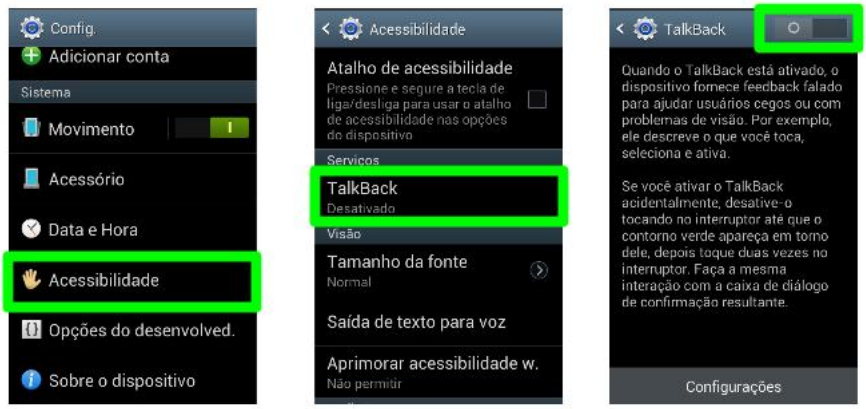


Figura 4.6: Etapas para a ativação do Talkback no Android

De fato, o Android oferece suporte para interface com navegação voltada para pessoas com deficiência visual. A ativação do Talkback, Figura 4.6, que é um serviço nativo do Android, adapta a lógica de interação de toda a interface do Android, facilitando a usabilidade por pessoas com dificuldades ou ausência de visão. Além de fazer automaticamente a tradução texto-áudio de qualquer informação presente na tela, o Talkback muda a lógica de cliques e insere sons e vibrações de resposta a qualquer ação realizada, desde toques em botões até deslizamento em listas. Ele só requer que uma pessoa com visão normal o ative no primeiro uso, e então, mesmo ao ser reiniciado o smartphone, retorna ao estado ativado. Isso permitiu que o projeto avançasse para outro ponto importante da interface: o tamanho dos elementos.

Figura 4.7: Tela inicial do aplicativo

Uma das dificuldades enfrentadas por essas pessoas ao utilizar aplicativos é selecionar o elemento correto na tela devido o tamanho reduzido em relação aos dedos. Por essa razão, o menu principal, Figura 4.7, divide a tela toda em quatro grandes áreas que funcionam como botões. Além disso, a barra de status do Android e de título do aplicativo foram ocultadas, para garantir o melhor aproveitamento de espaço da tela.

Outro problema a ser considerado foi a navegabilidade. Um dos requisitos para que uma pessoa sem visão pudesse utilizar um aplicativo é saber onde está, ou seja, impedir que o usuário se perdesse na sequência de menus. Por isso, além de não haver submenus na interface, o padrão de desenvolvimento de aplicações Android foi respeitado, com a inserção de descrição de conteúdo a todos os elementos da interface. Essas descrições ficam visualmente ocultas, mas são lidas apenas pelo Talkback, que transforma a informação em áudio.

Uma vez que o aplicativo não se limita a atender apenas pessoas com ausência total de visão, outro cenário considerado foi a utilização do aplicativo por pessoas com graus menos intensos de deficiência visual. Baseado na pesquisa de acessibilidade realizada por Shaun K. Kane et al[25] com pessoas de diferentes graus de falta de visão, constatou-se que fontes de tamanho grande e o contraste de cores são considerados características importantes para aplicações. Por isso, um esquema de cores fortes e contrastantes, e ícones grandes foram utilizadas para os componentes do aplicativo. E cada cor utilizada nos botões do menu principal foi também utilizada como cor de fundo da interface correspondente à opção selecionada.

Além disso, o tamanho das letras dos textos de resultado foi aumentado significativamente e formatado em negrito para facilitar uma possível leitura.

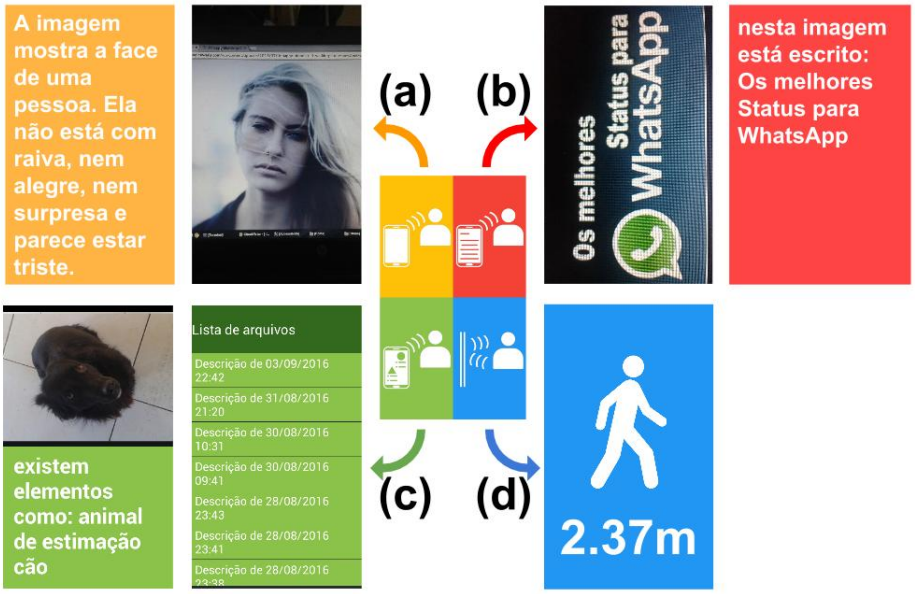


Figura 4.8: Tela principal e respectivas funcionalidades. (a) Descrição facial. (b) Extração de texto (c) Seleção de registros (d) Detecção de obstáculos

A Figura 4.8 ilustra a navegabilidade dentro do aplicativo, que por meio das medidas tomadas durante a construção, levou a obtenção das seguintes características para o aplicativo:

- Interface que permite áudio descrição;
- Menu principal, com apenas quatro botões e cores contrastantes;
- Botão que acessa a câmera e exibe a descrição com o máximo de informações da imagem capturada;
- Botão que acessa a câmera e exibe apenas textos da imagem capturada;
- Botão que leva a uma lista de descrições salvas e permite acessá-las;
- Botão que inicia conexão com o Arduíno;
- Menu de opções que permite inserir ou remover alguns sons/animações de resposta;

4.2.3 Extração de dados em imagem

A solução adotada para o reconhecimento de dados em imagem foi o Cloud Vision, uma API em nuvem da Google de reconhecimento de imagens, e com gratuidade limitada ao número de requisições mensais, cujos valores podem ser encontrados na Tabela 4.3. Como o projeto a princípio não é um produto e não é de grande porte, não exigiria muitas requisições e se mostrou uma solução adequada.

Tabela 4.3: Preços da API Google Cloud Vision por quantidade e tipo de solicitação

	Price per 1,000 units, by monthly usage			
Feature	1 - 1,000 units/month	1,001 - 1 Million units/month	1,000,001 - 5 Million units/month	5,000,001 - 20 Million units/month
Label Detection	Free	$5.00	$4.00	$2.00
OCR	Free	$2.50	$1.50	$0.60
Explicit Content Detection	Free	$2.50	$1.50	$0.60
Facial Detection	Free	$2.50	$1.50	$0.60
Landmark Detection	Free	$2.50	$1.50	$0.60
Logo Detection	Free	$2.50	$1.50	$0.60
Image Properties	Free	$2.50	$1.50	$0.60

Fonte: Google Cloud Platform[26]

A API criada em 2015 apesar de potente e ter atraído usuários interessados em automatizar a classificação de imagens, ainda não tem aparecido com muita frequência em trabalhos científicos. Entretanto, para demonstrar as capacidades da API, a Google apresentou na GCP NEXT 2016 o projeto Cloud Vision Explorer, um ambiente web galáctico contendo milhares de imagens separadas por categorias, que utiliza o Cloud Vision que é modelado pelo TensorFlow, uma biblioteca de software livre para inteligência de máquina também pertencente a Google [27].

Para se ter acesso aos serviços da Google Cloud, é necessário se cadastrar no sistema. Uma vez dentro do sistema, foi criado um projeto sobre o Vision API. Em seguida, para permitir a utilização desse projeto pela aplicação Android, foi necessário gerar uma chave de identificação de API. Por meio da importação dos pacotes em linguagem Java e dessa chave, bastou a implementação da rotina de montagem do objeto de requisição e de início de conexão, como apresenta o Código C.2, nos Apêndices. Logo depois, associou-se ao objeto a imagem fonte, o modo de compressão e de codificação da imagem a ser enviada, o português como idioma

de identificação de textos e por fim as categorias que se poderia buscar na imagem, entre elas, texto, rótulo e expressões faciais. Com o objeto configurado, bastou enviar a requisição ao servidor da Google e aguardar o resultado. Ao final do processo, o retorno da requisição veio em um objeto contendo as informações pedidas separadas por categorias e suas respectivas pontuações de confiança.

4.2.4 Adaptação textual do dado retornado

Após utilizar os serviços do Cloud Vision para a extração de informações da imagem, o passo seguinte foi adaptar as informações escritas para um conteúdo textual de fácil compreensão. O motivo é que as informações extraídas vinham em partes, separadas por categorias, nem sempre preenchidas com informação, e algumas vezes possuíam probabilidade baixa de estarem corretas podendo ser descartadas. Assim, foi necessário filtrar esses dados, acrescentando um limite mínimo de 80% de confiabilidade, número que se mostrou um meio termo adequado entre mostrar muita informação desnecessária ou pouca informação útil. Também, para garantir uma boa fluência no texto resultante, as informações foram filtradas por categoria e dependendo da qual pertencessem, produziriam uma saída textual específica. A rotina referente à essa adaptação pode ser encontrada no Código C.1 dos Apêndices.

4.2.5 Tradução de rótulos

Outra adaptação necessária foi em relação ao idioma. Apesar de a ferramenta ser capaz de identificar um infinidade deles durante a OCR, os resultados retornados de rótulos relacionados a imagem estavam sempre em inglês. A solução encontrada foi traduzir esses rótulos antes de compor a saída textual final. No entanto, não há suporte diretamente do Android para essa funcionalidade, e seguindo o exemplo da solicitação de serviços online, a solução encontrada foi utilizar novamente alguma API de tradução. Como a API de tradução da Google não oferece faixa de solicitações gratuitas, o que é extremamente importante, utilizou-se em vez disso a API de tradução Yandex, que assim como a Cloud Vision oferecia um limite de gratuidade.

4.2.6 Captura e exibição de resultado

Com o processo de obter uma descrição textual a partir de uma imagem finalizado, o passo seguinte foi conectar à entrada desse processo uma imagem capturada pela câmera, e sua saída à áudio descrição. O processo de descrição de imagem inicia-se em uma tela que

exibe as imagens vindas da câmera traseira do smartphone. O aplicativo, porém, não requisita em momento algum acesso a câmera frontal. Com um toque, ou dois quando o Talkback está ativado, o aplicativo captura a imagem, a salva no dispositivo no formato JPG, em uma nova pasta dentro do diretório do aplicativo e a envia para a nuvem. Enquanto o aplicativo aguarda o retorno do serviço solicitado, uma imagem de relógio pisca suave e lentamente na tela, enquanto um som de tic-tac é tocado como forma de informar o usuário que ele deve aguardar o processo. Também, a imagem capturada mantém-se como plano de fundo durante todo o processo de espera, e o toque na tela permanece desabilitado.

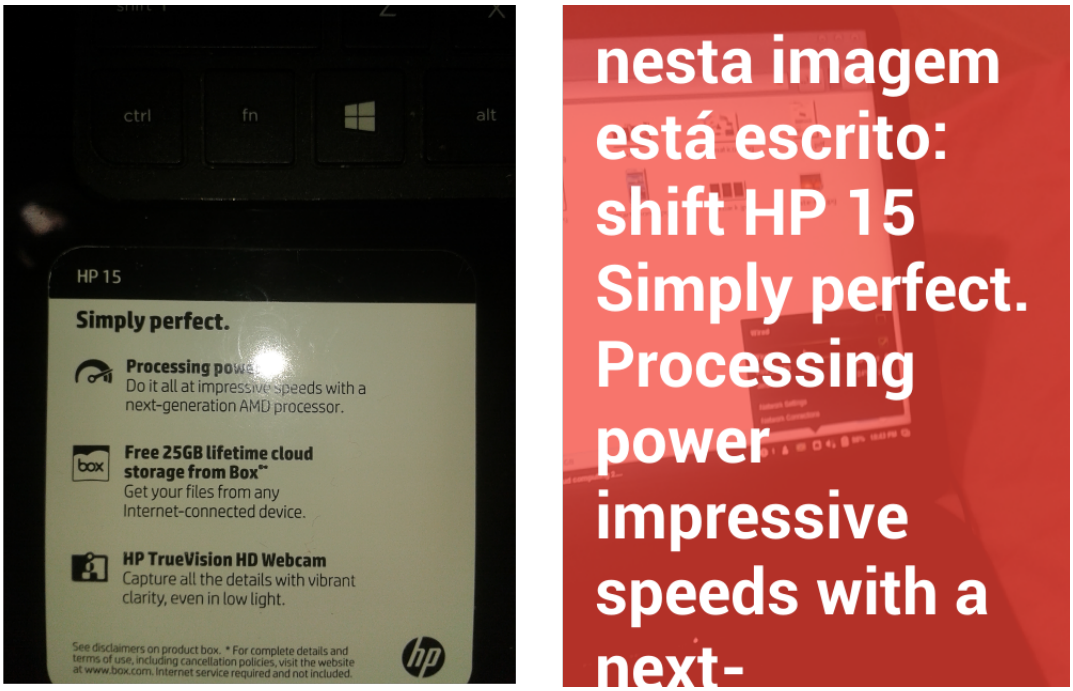


Figura 4.9: Imagem capturada e seu respectivo resultado

Quando o resultado enfim chega ao aplicativo, a animação e o som de processamento são interrompidos, o plano de fundo volta a mostrar as imagens da câmera, e sobre ela, abre-se uma caixa de texto translúcida e deslizável, conforme mostra a Figura 4.9, contendo o texto descritivo em negrito e em fonte grande. Com um longo clique, ou no caso do Talkback estar ativado, um curto clique seguido de um longo sobre a tela, o texto é copiado para a área de transferência. O botão de retorno permite ao usuário voltar para a câmera, e o botão de menu, o permite ativar ou desativar os efeitos visual e sonoro realizados durante a espera do processamento. Ao final, a descrição é salva em um arquivo em formato .txt na mesma pasta da imagem capturada.

4.2.7 Implementação de opções de reconhecimento

Com todo o tratamento dado ao conteúdo transcrito das imagens, para essa funcionalidade, foram criadas duas opções ligeiramente distintas para o aplicativo, como ilustra a Figura 4.10. A primeira requisitaria apenas o serviço de OCR sobre a imagem capturada com o intuito de reduzir o número de requisições, e possivelmente reduzir o tempo de resposta. Essa funcionalidade seria aplicável no caso específico de o usuário ter o conhecimento prévio de que a imagem poderia conter por algum texto, e que essa informação sozinha pudesse ser relevante e suficiente. A segunda funcionalidade solicitaria essa e outras informações ao Cloud Vision, que são: textos, faces, rótulos e pontos turísticos. Por fim, ambas as alternativas de extração de informação de imagens foram inseridas nos dois botões superiores do menu principal.

Figura 4.10: Comparação de funções entre os botões de solicitação de extração de informação de imagem

4.2.8 Registro de descrições

Para prover ao usuário a possibilidade de resgatar resultados anteriores, foi importante também permitir que o aplicativo oferecesse a opção de salvá-los. Nesse ponto surgiu uma questão: Qual seria a melhor forma de o usuário acessar esse registro?

Nomear os registros com parte da descrição poderia causar ambiguidade de nomes e poderia dificultar encontrá-los caso houvesse grande quantidade deles. Assim, foi decidido que os registros seriam nomeados com a data e hora de criação. Além disso, seriam sempre ordenados temporalmente ao serem carregados em uma lista pelo aplicativo. O motivo foi facilitar o acesso, pois os mais recentes apareceriam no topo, e data/horário são identificadores únicos que permitem fácil localização na procura por um registro. A Figura 4.11 apresenta uma lista de registros que foi gerada pela utilização do aplicativo.

Com isso decidido, foi necessário planejar a estrutura desse registro. A princípio considerou-se que o áudio referente à transcrição deveria ser salvo. No entanto, o propósito principal do aplicativo era apenas extrair informações de imagens e encontrar uma forma de fazer com que o usuário com deficiência visual pudesse acessá-la. Salvar o áudio deixou de ser necessário quando se percebeu que a função de áudio descrição é parte exclusiva da interface de acessibilidade do aplicativo, e não é obrigatória para todos os usuários. A intenção de salvar os dados não é pela voz da áudio descrição, mas exclusivamente por seu conteúdo.

Assim, cada registro correspondia a um diretório contendo apenas a foto no formato JPG, para referência de quem pode ver, um arquivo no formato TXT contendo a descrição e um arquivo TXT opcional contendo a posição de faces na imagem. A áudio descrição ficou sob responsabilidade da interface, após o carregamento do registro. Por fim, essa funcionalidade foi atribuída a um dos quatro botões do menu principal.

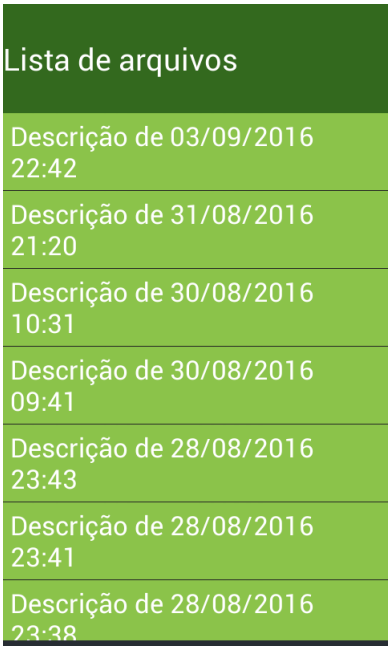


Figura 4.11: Lista de registros de descrição

4.2.9 Inserção de nome de pessoas

Para os registros de descrições foi criada uma funcionalidade extra: a flexibilidade de o usuário inserir na descrição o nome das pessoas em uma foto, no caso de se ter o conhecimento prévio de quem e onde estão na foto. Basicamente, quando o usuário abre o registro da lista, na metade superior da tela se exibe a foto, e na inferior, a caixa de texto rolável contendo a descrição. Como ilustra a Figura 4.12, ao clicar sobre a foto, se o toque for sobre algum rosto, o

aplicativo pergunta se o usuário deseja inserir o nome da pessoa na descrição. Se sim, basta ele escrever o nome na caixa de alerta e confirmar. Automaticamente o nome é salvo na descrição, e pode ser trocado a qualquer momento. Ao tocar na parte inferior da tela, onde fica o texto, a áudio descrição é realizada pelo talkback, se ativado.

Figura 4.12: Inserção de nome de uma pessoa em descrição de foto

4.2.10 Tratamento de sinais ultrassônicos

O tratamento de sinais vindos do sensor HC-SR04 é parte essencial do projeto, porém a mais simples de ser feita. Do ponto de vista de hardware, o sensor possui quatro pinos de conexão: VDD, GND, Trigger e Echo. O VDD foi conectado a alimentação de 5V do Arduíno, assim como o GND conectado ao terra. O Trigger e o Echo são respectivamente entrada e saída do sensor, para que o ele receba comando para emissão de ondas de 40 kHz de frequência e então retorne em sua saída o valor do tempo entre a emissão e a recepção da onda refletida. Esses dois pinos foram conectados respectivamente nos Pinos 4 e 5 do Arduíno. Informações mais detalhadas sobre o sensor podem ser encontradas no Anexo II. Além disso o Código C.3, nos Apêndices, apresenta a rotina executada no Arduíno.

Do ponto de vista lógico, o programa que roda no Arduíno periodicamente emite um sinal baixo de 2μs seguido de um alto de 10μs para gerar a onda, lê o valor recebido de tempo entre emissão e recepção do sinal refletido, calcula a distância segundo a equação 4.1:

$$\Delta S = V \times \Delta T \tag{4.1}$$

com ΔT sendo metade do tempo em μs recebido, pois se quer apenas o tempo entre emissão e reflexão, $V = 34300 cm/s$ representando a velocidade do som no ar, e ΔS a distância

em centímetros entre o sensor e obstáculo. Por fim, repassa o resultado para o módulo Bluetooth. Caso o valor da distância seja superior a 4 metros, limite de precisão do sensor, o valor não é repassado ao Bluetooth.

4.2.11 Comunicação Arduíno-Bluetooth

Para transmitir sinais do Arduíno para o smartphone via módulo Bluetooth, bastou inserir a informação na porta Serial. O mais importante foi, na verdade, decidir que informação deveria ser transmitida. Para garantir a modularidade do sistema, o Arduíno ficou encarregado apenas de enviar ininterruptamente as distâncias calculadas ao smartphone, sem realizar qualquer verificação de proximidade de obstáculos, função que o aplicativo Android ficou encarregado de fazer.

Do ponto de vista de circuitos, apenas quatro dos seis pinos do módulos foram utilizados. A razão foi que o HC-05 pode ser programado para ser mestre ou escravo. No modo escravo, os pinos KEY e STATE podem ser ignorados, e como não havia necessidade de o Arduíno se conectar a nenhum outro dispositivo, nem mesmo de requisitar conexões, esse foi o modo adotado para o Bluetooth no sistema. Foram então conectados VCC e GND aos respectivos pinos do Arduíno, para alimentar o módulo. O pino TXD de transmissão do módulo foi conectado ao pino RX do Arduíno, para recepção de sinais de comunicação vindos do smartphone. Por fim, o pino TX do Arduíno foi conectado ao RXD do módulo para receber os valores de distância a serem transmitidos ao smartphone. Para adaptar a tensão de saída do Arduíno à tensão adequada do módulo, foi necessário implementar um circuito divisor de tensão, pois o sinal proveniente do Arduíno é de 5V porém a tensão recomendada do módulo é da ordem de 3V.

4.2.12 Comunicação Smartphone-Bluetooth

Do lado oposto da comunicação, no aplicativo Android foi necessário receber os dados do Arduíno. Para tanto, um ciclo de comandos foi executado em plano de fundo. Primeiramente, o Bluetooth era ligado e fazia-se uma varredura repetitiva de dispositivos ao redor. Caso encontrasse um com o nome HC-05, a varredura era interrompida e tentava-se iniciar um conexão. Ao ser iniciada, o aplicativo iniciava uma leitura contante do buffer de entrada e o dado, distância até um possível obstáculo, era tratado e gerava-se quatro possíveis classificações:

- Entrada na área de atenção: A distância entre o sensor e o obstáculo entrou na faixa de 30cm a 200cm. Um arquivo de áudio contendo um sinal de ruído branco é executado com

volume inversamente proporcional à distância.

- Entrada na área de alerta: A distância entre o sensor e o obstáculo entrou na faixa inferior a 30cm. Um sinal periódico de alerta sonoro e vibração são executados.
- Dentro da área de atenção: A distância entre o sensor e o obstáculo mantém-se entre 30cm e 200cm. Permanece a execução de som de atenção, atualizando seu volume de acordo com a distância.
- Fora: Desliga todos os sinais.

Essa funcionalidade por fim foi colocada como ação do último dos quatro botões do menu principal. O circuito completo, contendo o sensor de obstáculos, o módulo Bluetooth e o Arduíno, pode ser visualizado na figura 4.13.

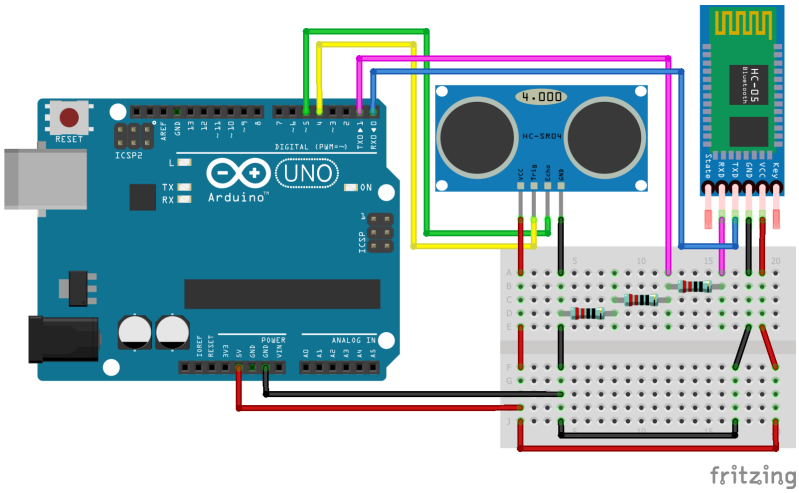


Figura 4.13: Circuito Arduíno com módulo Bluetooth e sensor de obstáculos

4.2.13 Manufatura da caixa do circuito

Para estudo de caso e também facilitar o manuseio do circuito, o protótipo de detector de obstáculos necessitou de uma pequena caixa de proteção. Para isso foi criada uma caixa arredondada utilizando-se da porção central de uma garrafa PET de dois litros. O topo e a base da garrafa foram removidos com uma tesoura, formando um cilindro, e duas dobras longitudinais foram feitas para se ajustar ao comprimento do circuito. Parte do trecho dessas dobras foi cortado em ambas as bases para criar duas abas, reduzir o tamanho e fechar a caixa. Foi inserido também na lateral da caixa um interruptor de tipo navio com dois pinos para facilitar o ligar e desligar. Por fim foram feitos recortes na caixa para adaptá-la ao sensor e ao Bluetooth, e o circuito todo foi inserido.

5 Resultados e Discussões

Com as funcionalidades planejadas para o projeto finalmente implementadas, o passo seguinte foi a realização de testes. Esse capítulo será dividido em três seções tal que na primeira serão tratados dos resultados da extração de informação de imagens, a segunda tratará da performance do detector de obstáculos e a terceira fará uma análise de consumo do sistema.

5.1 Descrição de Imagens

Na descrição de imagens duas considerações foram necessárias para garantir o correto desempenho do sistema: o tempo de resposta, pois não é conveniente deixar o usuário esperando por muito tempo pela informação requisitada, e a porcentagem de erros dos resultados obtidos, para que o usuário receba informação confiável do aplicativo. Como a Google Cloud API é uma ferramenta online, a primeira dificuldade encontrada para eficiência de processamento foi o transporte de dados para o servidor da Google. Quanto mais dados são transmitidos pela rede e processados pela rotina no servidor, maior o tempo de latência para a resposta. Isso significa que a velocidade da Internet do usuário será sempre um limitador.

De modo geral, também foram realizados testes sobre a interface para avaliação da corretude funcional do aplicativo após o término de cada módulo desenvolvido, e sempre que erros surgiam, puderam ser corrigidos. O aplicativo foi submetido ao Android Lollipop, versões 5.0.1 e 5.1.1, onde foram encontrados e corrigidos problemas de foco da câmera e de tamanho distorcido de alguns componentes da interface.

5.1.1 Teste para diferentes dimensões de imagens

Apesar de a velocidade da internet ser um fator limitante, a quantidade de dados transmitida, em alguns casos, pode ser flexibilizada uma vez que a imagem pode ter sua resolução reduzida e exibir um número menor de bytes para ser representada. Entretanto, essa redução resulta em um detalhamento menor da imagem, o que pode dificultar seu processamento, causar erros de interpretação de seu conteúdo e por fim não retornar resultados corretos.

A fim de confrontar resolução de imagem, tempo de processamento e erros dos resultados, para cada solicitação de extração de informação das imagens, a mesma imagem contida na Figura 5.1 foi enviada com diversas resoluções, e os tempos das fases do processo foram medidos. Já a Figura 5.2 apresenta os resultados para três diferentes resoluções da Figura 5.1.

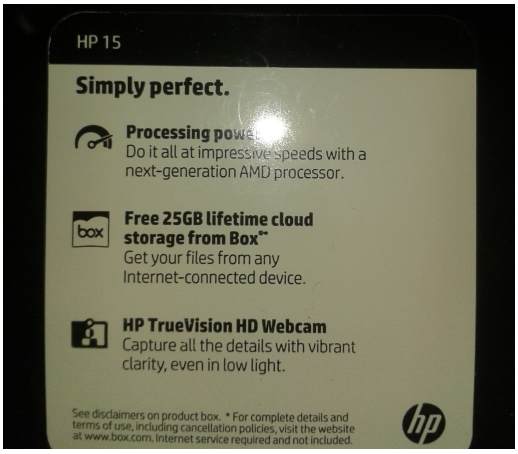


Figura 5.1: Captura de imagem da etiqueta do computador HP, com defeito de flash, para descrição

nesta imagem está escrito: HP 15 Simply perfect. Processing power Do it all at impressive Speeds with a next-generation AMD processor. Free 25GB lifetime cloud 00X storage from Box Get your files from any Internet-connected device. HP TrueVision HD Webcam Capture all the details with vibrant clarity, even in low light. See disclaimers on product box. For complete details and terms of use, including cancellation policies, visit the website at www.box.com, Internet service required and not included.

(a)

nesta imagem está escrito: HP 15 Simply perfect. Processing powe Do it all at impress peeds with a next-generation AMD processor. Free 25GB lifetime cloud storage from Box Get your files from any Internet-connected device. HP True Vision HD Webcam Capture all the details with vibrant clarity, even in low light. See disclaimers on product box. For complete details and terms of use, including cancellation policies, visit the website at www.box.com Internet service required and not included.

(b)

nesta imagem está escrito: HP 15 Simply perfect. Processing powR mpress next-generation ANAL pluce sul Free 25GB lifetime duud storage from Box Get your files Ton Jny Internet connected cevice HP Tru BVision HU Webo am Copture the detais with vibrant darity, even inluw lilli. box for compl-1 use inclue inteinetsarute resu ie:

(c)

Figura 5.2: Resultado da extração apenas de texto sobre a imagem da Figura 5.1 com as resoluções de (a) 2560x1920, (b) 1024x768 e (c) 480x360

A Tabela 5.1 mostra que a resolução da imagem é proporcional ao tempo total de processamento da informação requisitada e inversamente proporcional a quantidade de erros do resultado. É fundamental lembrar que o tempo de processamento será sempre relacionado à velocidade da internet, portanto não são valores absolutos. O erro de cada imagem é calculado

pela equação 5.1 a seguir:

$$Erro = \frac{\text{Número de caracteres na imagem} - \text{Número de caracteres corretos}}{\text{Número caracteres na imagem}} \times 100\% \quad (5.1)$$

Tabela 5.1: Tempos das fases da transcrição de imagem sobre a etiqueta do computador HP, ilustrada pela Figura 5.1

	Resultados		
	5.2a	5.2b	5.2c
Resolução	2560x1920	1024x768	480x360
Erro de reconhecimento	0.8%	1.9%	63.1%
Tempo de compressão	3667ms	939ms	206ms
Tempo de codificação	570ms	69ms	66ms
Tempo de transferência e processamento	103,849s	17,871s	8,285s
Tempo de reescrita	17ms	97ms	4ms

Varias características importantes podem ser extraídas desses resultados. Primeiramente, a imagem possui um defeito causado pelo brilho intenso do flash concentrado num único ponto no momento da captura, e esse defeito afeta diretamente 3 palavras do texto. Essas palavras afetadas pelo brilho correspondem exatamente as malformadas, em vermelho, do resultado apresentado pela Figura 5.2b, erro esse que não ocorre com o apresentado pela Figura 5.2a. Isso permite a inferência de que a redução da resolução pode ter tornado a OCR menos precisa e esse problema ter sido potencializado pelo brilho do flash ao ponto de distorcer completamente a grafia das palavras.

É possível que se o brilho refletido não tivesse sobreposto essas palavras, o resultado de 5.2b fosse quase idêntico ao de 5.2a. Essa hipótese pode ser confirmada pela Tabela 5.2, pois as dimensões recomendadas pela Google para a detecção de texto correspondem as da Figura 5.2b. Além disso, dimensões inferiores reduzem a acurácia do resultado, o que pode ser confirmado pela Figura 5.2c, enquanto superiores aumentam o tempo de processamento e uso da largura de banda sem necessariamente promover melhora significativa da acurácia[26].

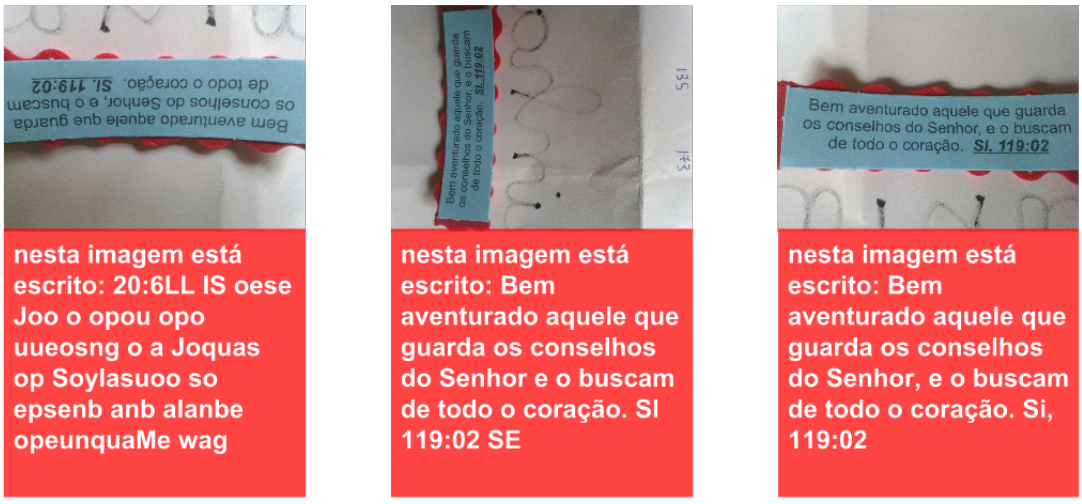
Tabela 5.2: Dimensões de imagem recomendadas para cada característica buscada

Vision API Feature	Recommended Size *	Notes
FACE_DETECTION	1600 x 1200	Distance between eyes is most important
LANDMARK_DETECTION	640 x 480	
LOGO_DETECTION	640 x 480	
LABEL_DETECTION	640 x 480	
TEXT_DETECTION	1024 x 768	OCR requires more resolution to detect characters
SAFE_SEARCH_DETECTION	640 x 480	

Fonte: Google Cloud Platform[26]

5.1.2 Teste para reconhecimento de texto girado

Considerando o fato de que o usuário do aplicativo certamente não saberá de antemão qual a posição do texto do qual deseja obter informações, é possível ocorrer a captura de um texto disposto em diversos ângulos. Ao se realizar os testes sobre textos com diferentes posições angulares, identificou-se que entre -90° e 90° não há qualquer problema na extração de informação. Entretanto, acima desse limite, a aplicação simplesmente não consegue mais identificar os caracteres corretamente. O resultado obtido pode ser visualizado pela Figura 5.3



(a) (b) (c)

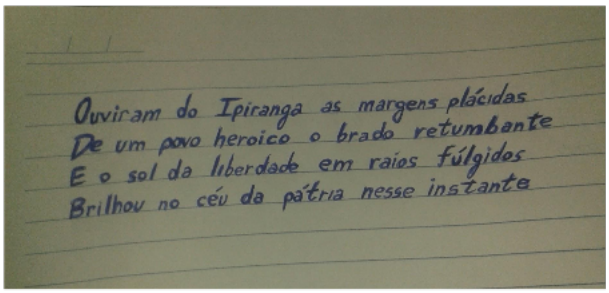
Figura 5.3: Extração apenas de texto de uma imagem sob três diferentes posições. (a) 180°, (b) 90° e (c) 0°

Apesar de esse resultado mostrar que há uma limitação na capacidade de API reconhecer caracteres, é possível compreender a razão. Uma hipótese é que o algoritmo apenas verifique

que há linhas horizontais de textos, e não considere a possibilidade de o texto estar virado em 180°. Então, ele deve comparar o simbolo invertido com os de sua base de dados, e o que se encaixar melhor é considerado o correto. Ao analisar com mais cuidado a Figura 5.3a, pode-se perceber que apesar de o texto retornado não ter qualquer significado real, existe uma razão na formação de cada simbolo. Há uma considerável semelhança entre a letra "a"girada de 180° e a letra "e", entre "L"e "1", entre "w"e "m", e assim por diante. Quanto a Figura 5.3b, praticamente não houve erros durante a OCR, mesmo apresentando um texto girado de 90°, assim como o teste apresentado pela Figura 5.3c.

5.1.3 Teste para reconhecimento de texto manuscrito

Uma das possibilidades de utilização do aplicativo seria a leitura de bilhetes escritos a mão. Alguns testes foram realizados para se ter conhecimentos dos limites das capacidades da API, quanto a escrita a mão, seja ela de forma ou cursiva.



Nesta imagem está escrito: Ouviram do tPuranga as margens placidas De um povo heroico o brado retumbante E o sol da liberdade em raios fulgidos rilhou no ceu da patria nesse instante

Figura 5.4: Resultado da extração de texto sobre imagem contendo escrita manual em letra de forma.

Os resultados mostraram que existe maior limitação da API quanto a identificação de caracteres manuscritos, se comparada a de digitais. A Figura 5.4 apresenta o caso de um texto capturado com a mais alta resolução que o aplicativo oferece, de 2560x1920 pixels, e mesmo assim há alguns erros de identificação. Porém, como é possível observar nas Figuras 5.5 e 5.6, os resultados só ficam gravemente incorretos quando os textos são escritos em letra cursiva. Nesse cenário, o reconhecimento até ocorre, mas de uma quantidade muito pequena do total de palavras, apresentando diversos erros, mesmo em traços largos e cor contrastante.

É possível que a API não ofereça suporte para modelos de letra cursiva, uma vez que nos Estados Unidos, país sede da Google, esse tipo de escrita já vem sendo abandonado com o advento dos computadores pessoais e celulares nas últimas décadas[28]. Apesar de ser ter sido considerada uma decisão polêmica, as escolas já não mais são obrigadas a ensinar os alunos a escreverem em letra de mão. Por essa limitação da API, a transcrição de textos em letra cursiva se mostra, na prática, impossível pela aplicação.

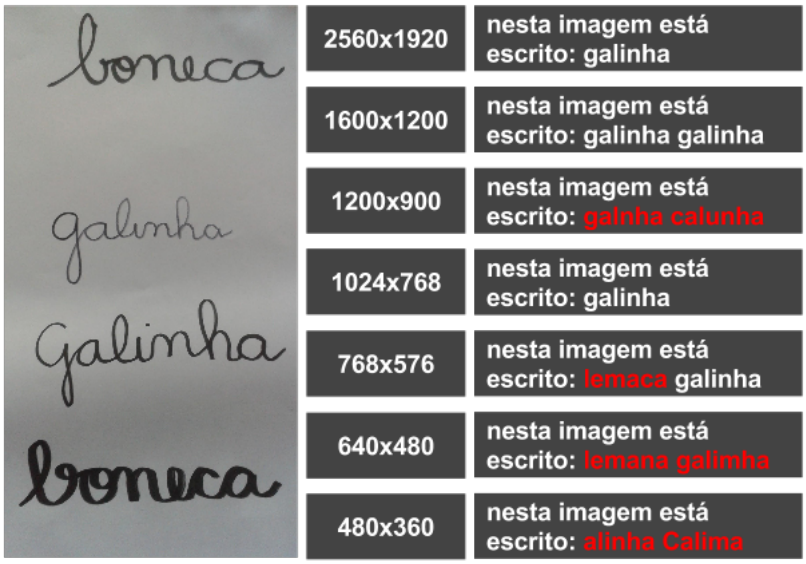


Figura 5.5: Resultados da extração de texto sobre imagem, em diferentes resoluções, contendo palavras manuscristas em letra cursiva.

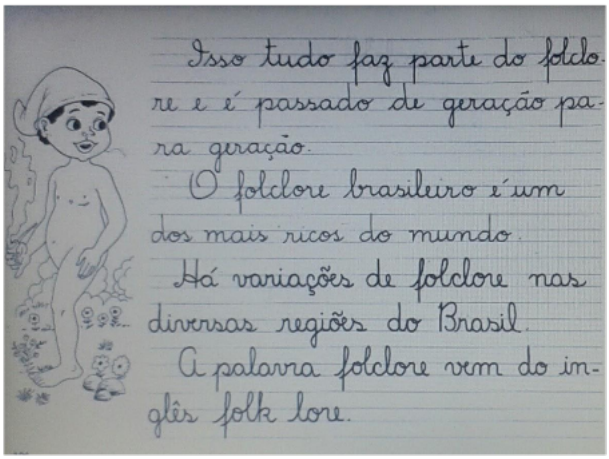


Figura 5.6: Resultado da extração de texto sobre imagem contendo texto escrito em letra cursiva.

5.1.4 Teste para rotulação de elementos da cena

Saber o que se passa tendo conhecimento do que há ao redor é uma dos objetivos do projeto. Para saber o quais as limitações da API na detecção de objetos em imagens, foram realizados testes apontando a câmera para os mais diversos objetos a fim de se avaliar sua performance nesse quesito. A Figura 5.7 apresenta o resultado completo retornado pela extração de rótulos da imagem de um cachorro. É possível perceber que características como "Cachorro de brinquedo"e "Yorkshire Terrier"não são aplicáveis ao animal da foto. Como o aplicativo só considera resultados com pontuação acima de 80%, rótulos como esses, em vermelho, foram ignorados no resultado visualizado pelo usuário.

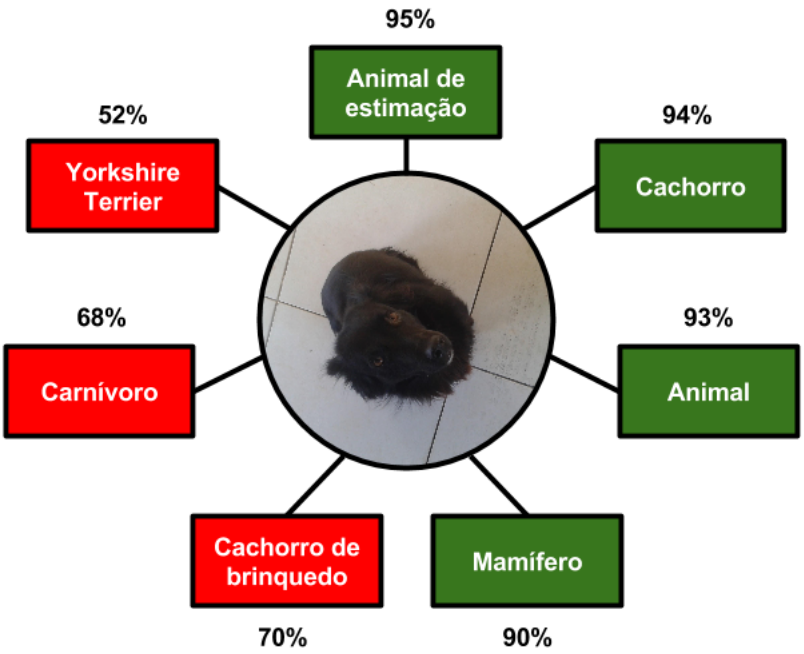


Figura 5.7: Rótulos extraídos da imagem de um cachorro

A escolha do valor mínimo de pontos para que cada rótulo fosse aceito foi puramente empírica. Apesar de ter resultado em boa descrição para a imagem da Figura 5.7, o valor escolhido que elimina alguns resultados subaproveitou os rótulos da imagem da Figura 5.8, descrevendo-a apenas como "Dispositivo", já que "Laptop", palavra mais adequada, foi ignorada.

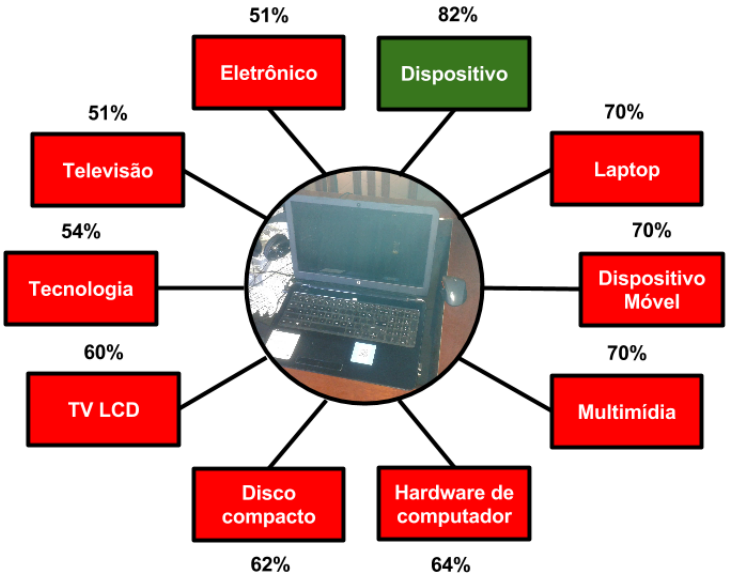


Figura 5.8: Primeira extração de rótulos da imagem de um laptop

A definição de um valor que simultaneamente seja capaz de descrever um elemento da cena e não sobrecarregue o usuário com informações, muitas vezes desnecessárias, não é tão simples, e uma descrição curta e detalhada dificilmente será conseguida. No entanto, o resultado não depende apenas da definição desse valor, mas também da própria imagem. A Figura 5.9 ilustra o mesmo laptop capturado novamente sob iluminação e posição diferentes. Nesse cenário, o resultado foi capaz de promover uma descrição mais detalhada do objeto em cena.

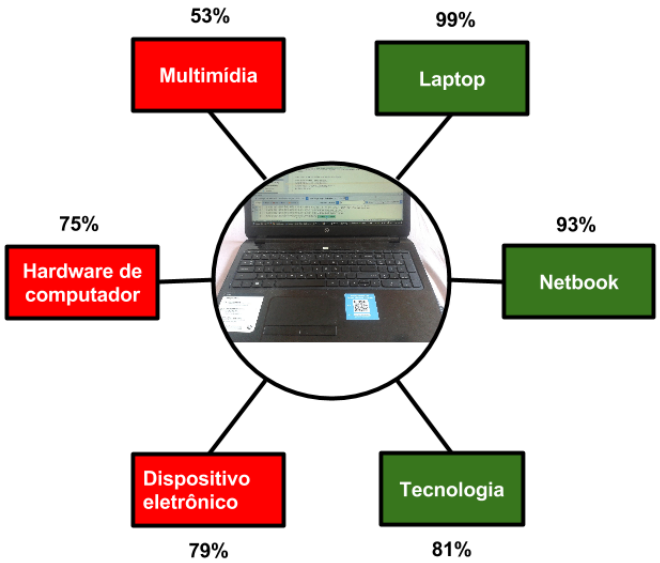


Figura 5.9: Segunda extração de rótulos da imagem de um laptop

Nos casos citados, independentemente de qual foi o critério para ignorar alguns resultados, todos os rótulos quase sempre tiveram relação com o objeto na imagem. Entretanto, em algumas situações a API falhou em identificar ao menos um rótulo corretamente para a imagem capturada. A Figura 5.10 ilustra esse problema. Nela, foi capturada a imagem de uma cadeira, porém além de não haver resultados acima do limite mínimo de pontuação, nenhum dos quatro rótulos tem qualquer relação com a imagem.

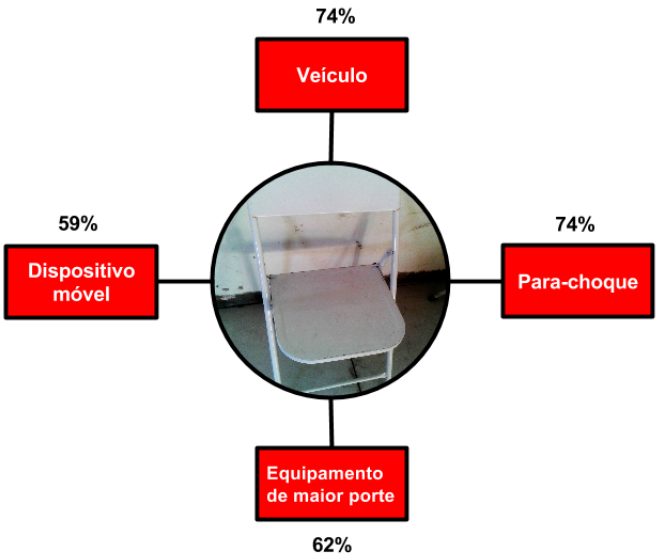


Figura 5.10: Extração de rótulos da imagem de um cadeira

É possível que a explicação para esse resultado seja que o ambiente no qual a cadeira estava inserida tivesse afetado sua imagem capturada ao ponto de a API não ser capaz de identificar o que estava de fato em cena, e confundir o objeto com o para-choques de um veículo. Obviamente não é desejável que confusões como essa ocorram, entretanto, esse é um fenômeno parecido com a "ilusão de óptica". Ao analisar a imagem, percebe-se que as faixas escuras entre a parede e o chão, atrás da cadeira, somadas a sua estrutura em grades no meio podem ter sido avaliados como a parte frontal de um carro: Dois faróis pretos nas laterais, um capô branco no topo e para-choques com grades na porção centro-inferior.

5.1.5 Teste para classificação de expressão facial

Dentre as características que se pode obter de uma imagem, certamente a classificação de expressões faciais é a mais difícil de se obter. Como mostrou anteriormente a Tabela 5.2, as dimensões recomendadas para a detecção de face é a maior dentre todas as outras ca-

racterísticas. Como apresentado na Seção 4.1, o tempo de resposta cresce significativamente conforme a dimensão da imagem aumenta. Isso significa que para se obter resultados corretos é necessário enviar a imagem com alta resolução e aguardar mais tempo. A Figura 5.11 apresenta o resultado obtido pela solicitação do serviço de detecção de faces e extração de suas expressões faciais. As imagens foram tiradas diretamente pela câmera do celular durante a execução do aplicativo a partir da exibição da tela do computador.

Figura 5.11: Expressões faciais detectadas em imagens de rosto

Fonte das fotos: Google Imagens

Apesar dos acertos nas descrições das faces, os resultados não foram sempre corretos. Foi possível perceber que a detecção de expressões de raiva e tristeza dificilmente ocorriam, mesmo com o aumento da qualidade da imagem, ou com faces expressivas. A Figura 5.12 apresenta casos de falha com imagens de faces com expressões de tristeza. Os resultados variaram desde a não identificação da expressão evidente até a incapacidade de encontrar a face.

Figura 5.12: Expressões faciais de tristeza não detectadas em imagens de pessoas tristes

Fonte das fotos: Google Imagens

5.2 Detecção de Obstáculos

Na implementação da funcionalidade de detecção de obstáculos, duas considerações precisaram ser feitas. Primeiramente, para atender aos resultados esperados pelo usuário, o sensor de obstáculos deveria ser preciso o suficiente para garantir um deslocamento confiável. Além disso, procurou-se reduzir ao máximo o tamanho do circuito, para garantir sua usabilidade.

5.2.1 Precisão e acurácia das medidas de distância

Assim que o circuito detector de obstáculos foi criado e apresentava resultados medidos pelo sensor, a tarefa seguinte foi verificar a confiabilidade dos dados. A Tabela 5.3 mostra os resultados das simulações para diferentes distâncias em relação a uma parede.

Tabela 5.3: Dados extraídos da simulação do sensor de obstáculos sobre distâncias variadas e conhecidas

	Simulações							
Distância Real (cm)	10	20	30	40	50	100	200	300
Distância média aferida (cm)	15	27	39	52	64	127	256	384
Desvio padrão (cm)	0.70	0.80	0.80	0.84	0.87	0.92	0.93	1.01
Número de amostras	336	304	348	444	296	453	481	371
Tempo de simulação (s)	21	25	31	37	26	42	41	42
Distância ajustada (cm)	11.01	20.42	29.83	40.03	49.44	98.85	200.03	300.42

Nota-se que o sensor produziu valores precisos, uma vez que o desvio padrão foi no

entorno de apenas 1cm. Entretanto, sua acurácia, que é a medida de quão próximo do valor real está do amostrado, poderia causar erros significantes de classificação de distância pelo aplicativo. Por essa razão, foi necessário inserir uma correção nos valores de distância lidos. Por meio do método dos mínimos quadrados, e baseado nas amostras das simulações foi obtida a equação 5.2 para o ajuste dos valores:

$$Da = 0.78Ds - 0.75 \tag{5.2}$$

em que Da representa a distância ajustada e Ds a medida pelo sensor. Com essa adaptação, os resultados ganharam a acurácia necessária e atribuíram maior confiabilidade ao dispositivo.

5.2.2 Usabilidade do detector de obstáculos

Outro ponto considerado foi o tamanho dos elementos do circuito. Quanto menor e compacto, mais usável ele poderia se tornar. Como se tratava apenas de um protótipo, e devido à limitação dos materiais disponíveis para o projeto, foi criada uma caixa para o circuito com dimensões pouco superiores a média das de um smartphone. O resultado obtido pode ser visto na Figura 5.13.

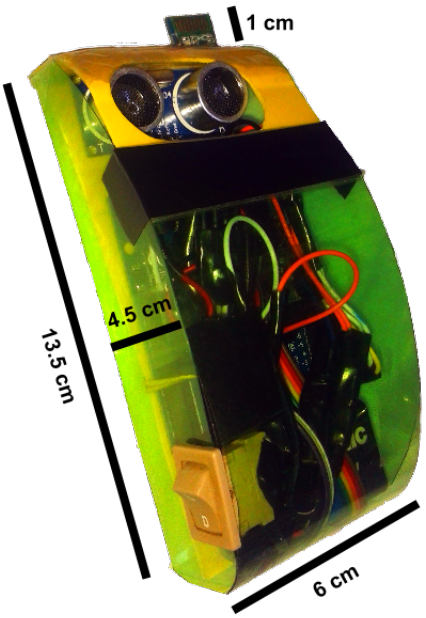


Figura 5.13: Protótipo do circuito detector de obstáculos

5.3 Avaliação de consumo do sistema

O consumo de energia certamente é um grande limitador na implementação de qualquer sistema. É importante portanto ter conhecimento da potência dissipada tanto pelo aplicativo executado no smartphone, quanto do circuito detector de obstáculos, formado pelo Arduíno e demais módulos.

5.3.1 Consumo no smartphone

Na avaliação de consumo do aplicativo rodando no smartphone utilizou-se o Power Tutor, um aplicativo que mede o consumo de outros aplicativos no Android, e pode ser baixado gratuitamente pela Play Store. Uma vez ligado, ele avalia todos os processos que são executados pelo Android. A Figura 5.14 ilustra os resultados obtidos pela avaliação do aplicativo em um intervalo de 300 segundos. Entre os instantes 30 e 90 segundos o aplicativo se manteve conectado ao detector de obstáculos, e entre os instantes 120 e 240 segundos permaneceu na função de reconhecimento de texto.

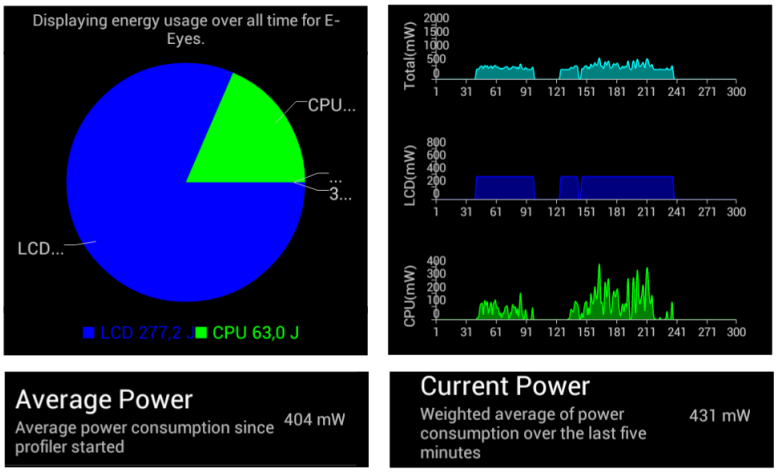


Figura 5.14: Potência consumida pelo aplicativo no smartphone pela solicitação de OCR e pela conexão com o detector de obstáculos

É possível notar que durante a solicitação de reconhecimento de texto houve um maior consumo da CPU do que durante o processo de detecção de obstáculos. O resultado faz sentido, uma vez que na primeira funcionalidade utiliza-se mais recursos como Wifi, câmera (captura, flash e preview), animação e som de processamento. Na detecção de obstáculos apenas o dispositivo Bluetooth do smartphone é utilizado.

Além disso, o consumo da bateria se dá muito mais pela iluminação da tela do que pela CPU. O gráfico de consumo pelo LCD apresentou descontinuidade nos dados devido o fato de o aplicativo ter sido desligado entre os testes. Na média, a potência consumida pelo aplicativo para a utilização das duas principais funcionalidades do sistema foi de 404mW.

5.3.2 Consumo no detector de obstáculos

Para avaliar com maior precisão o consumo de energia do circuito, durante 80 segundos foi medida a corrente consumida. Inicialmente, o circuito permaneceu ligado, porém o aplicativo do smartphone estava desligado. Aos 30 segundos o aplicativo foi ligado e a função de detecção de obstáculos foi acionada. Nesse instante nota-se pela Figura 5.15 que houve uma queda na corrente. Isso ocorreu porque quando o módulo Bluetooth é alimentado, mas não está conectado, ele fica constantemente enviando sinais para notificar sua presença. Quando uma conexão é estabelecida, ele passa a enviar apenas os sinais de comunicação, o que requer menos energia.

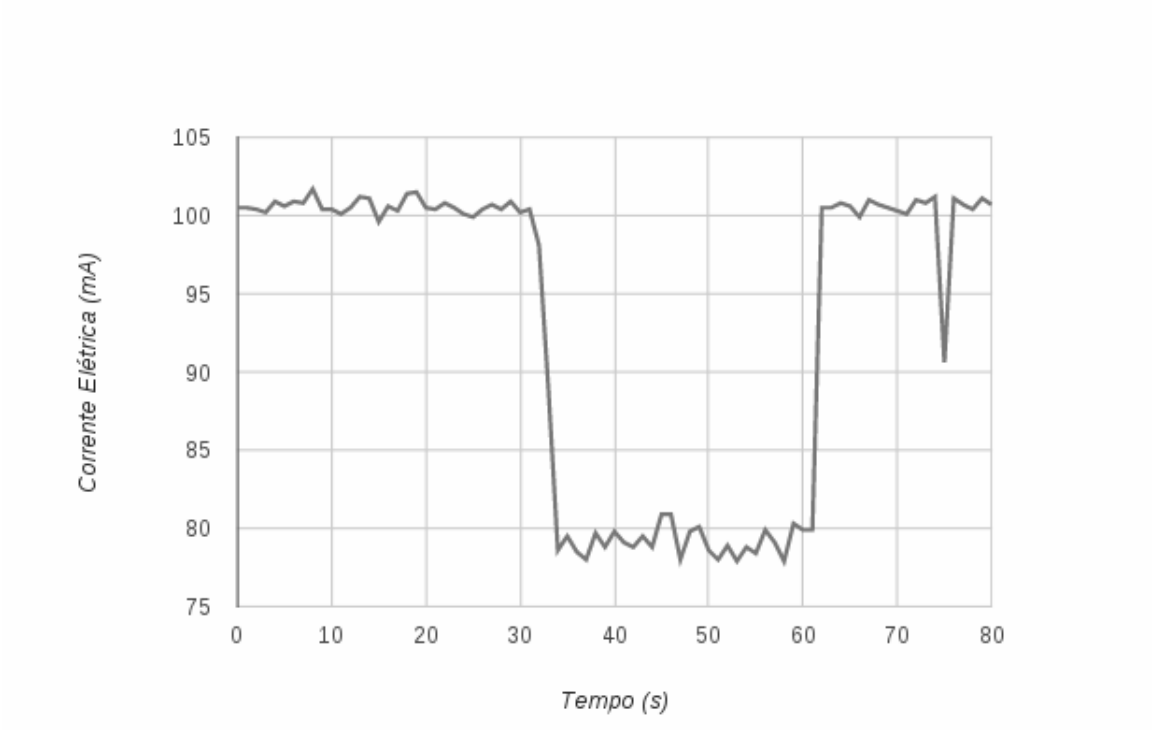


Figura 5.15: Corrente elétrica consumida pelo circuito antes, durante e após a conexão com o smartphone

Além de avaliar o consumo do circuito como um todo, avaliou-se também o quanto cada módulo conectado e devidamente ligado consome do total. Os módulos foram ativados de modo

alternado a fim de se medir a corrente consumida individualmente, e para cada caso teste, o valor foi medido durante 25 segundos. O resultado pode ser visualizado pela Figura 5.16, que mostra que a maior parte do consumo é resultado do funcionamento do dispositivo Bluetooth.

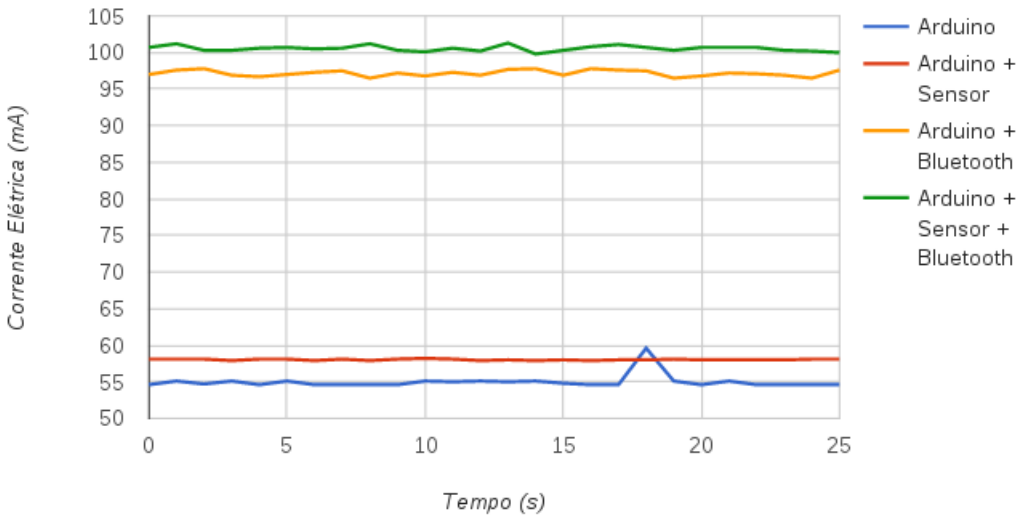


Figura 5.16: Corrente elétrica consumida pelo circuito discriminando o papel de cada módulo

A partir dos valores amostrados de corrente consumida pelo circuito, foi possível calcular a corrente média, o desvio padrão e a potência média, apresentados pela Tabela 5.4. A potência P foi calculada baseado no valor médio de corrente I e na tensão V de alimentação de 9.6V, pela equação 5.3:

$$P = VxI \tag{5.3}$$

Tabela 5.4: Valores médios de corrente e potência consumidos pelos módulos do circuito

	Arduíno	Sensor	Bluetooth	Conjunto
Corrente Média(mA)	55	3.04	42.2	100.55
Desvio Padrão (mA)	1	0.09	0.42	0.38
Potência Média(mW)	529.5	29.3	406.2	968.3

De fato, o módulo Bluetooth consome pouco menos que o Arduíno, mas isso representa quase a metade do consumo total. A Figura 5.17 ilustra o resultado em percentuais de consumo.

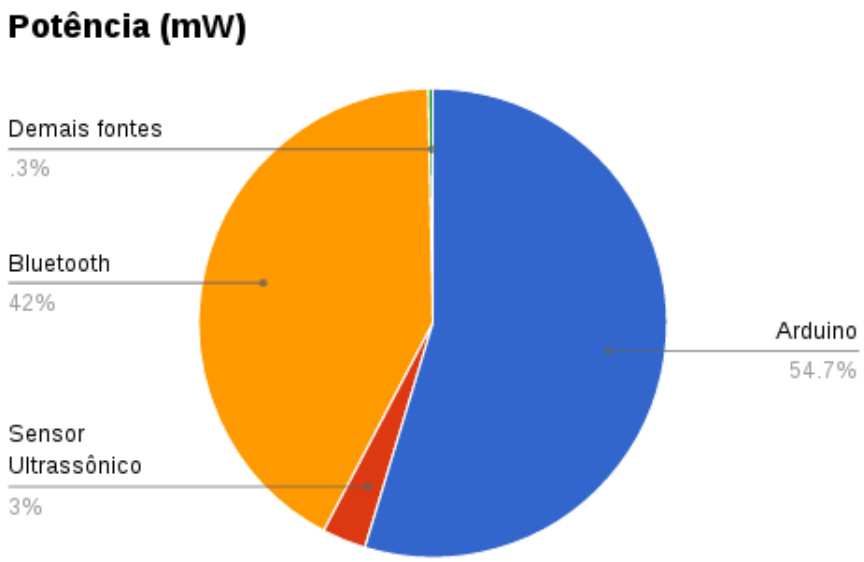


Figura 5.17: Porcentagem de potência consumida por cada módulo do circuito

6 Conclusão

Esse projeto teve como objetivo a aplicação dos conhecimentos adquiridos durante a graduação a um problema real que nem sempre recebe a devida atenção de profissionais da área. Esse problema é o da acessibilidade de pessoas com deficiência visual. Os resultados mostraram que a API Google Cloud Vision é uma ferramenta poderosa no campo de visão computacional e seus serviços podem ser utilizados pelas mais diversas aplicações, incluindo a possibilidade de auxiliar pessoas sem o sentido visual a acessar informações visuais com autonomia. Será apresentado neste capítulo um balanço dos resultados obtidos através dos testes, a fim de avaliar sua utilidade a aplicabilidade.

6.1 Limitações

Apesar de a API da Google ter se mostrado eficiente para o propósito do projeto, a ferramenta causou algumas limitações para os resultados. Um deles é o fato intrínseco de que a API está implementada para operar em nuvem e requer conexão com a internet. Isso poderia de certa forma impedir que pessoas sem acesso a internet pudessem usá-lo. Entretanto, existe uma grande tendência de os dispositivos e as pessoas se tornarem cada vez mais conectadas, devido o avanço da computação e das telecomunicações, e com os preços de smartphones mais acessíveis, e isso poderia preparar um ambiente mais propício para o uso do aplicativo. Outro ponto negativo é a limitação da gratuidade de utilização da API. Com o intuito do projeto sendo também o de inclusão digital, atribuir preço para o acesso ao aplicativo iria, de certa forma, contra esse propósito.

Já com relação ao circuito detector de obstáculos, um dos problemas foi o consumo de energia pelo módulo Bluetooth, que representou quase a metade do consumo total. Num cenário em que a pessoa o utilize sempre ao caminhar, a bateria poderia descarregar com certa rapidez. Além disso, a utilização de um único sensor foi capaz apenas de mostrar que é possível se guiar por ele, entretanto, para um caminhar mais seguro, seriam necessários mais sensores apontando para várias direções. Também, o tamanho do dispositivo se mostrou relativamente grande para ser utilizado na parte frontal do corpo, o que poderia dificultar a usabilidade.

Não foi possível, devido à indisponibilidade de mais recursos, a produção de testes mais elaborados com um número maior de versões do Android, e também, testes de usabilidade com usuários para o qual o projeto é dedicado, o que permitiria sua validação. Por essa razão, não

é possível concluir se o sistema de fato atende as necessidades desses potenciais usuários.

6.2 Acertos

Por outro lado, houve vários pontos positivos que se podem extrair deste projeto. A leitura de textos, que é a funcionalidade mais importante do sistema mostrou-se acurada. Com ela seria possível a leitura de rótulos de produtos, folhetos, livros e até bulas de remédio, mesmo possuindo letras pequenas, cupons fiscais, dinheiro, entre diversos outros conteúdos escritos. Com o aplicativo é possível também oferecer uma descrição superficial automática do ambiente ao redor e de expressões faciais, e permite até que o usuário insira os nomes das pessoas em uma foto.

Com relação ao detector de obstáculos, apesar da limitação do número de sensores, os resultados se mostraram precisos, e o dispositivo tem o potencial de se tornar ainda mais útil e confiável caso o número de sensores seja aumentado e seu tamanho reduzido.

6.3 Opinião de potenciais usuários

Durante o período de desenvolvimento do projeto foi importante procurar ouvir um pouco as pessoas para as quais o projeto se destinaria. Assim, os membros do grupo do Facebook "Cegos e a Tecnologia", em sua maioria com algum grau de deficiência visual, deram algum suporte apresentando algumas de suas dificuldades, ou sugerindo funcionalidades.

Um dos problemas citados foi a dificuldade de posicionamento da câmera para utilização de aplicativos para descrição de valor de dinheiro, e o tempo gasto na procura pela posição ideal. Nesse quesito, pode se dizer que houve sucesso, uma vez que a funcionalidade de descrição de textos mostrou-se precisa para vários ângulos. Outro ponto levantado foi a sugestão de implementação do comando de voz. A ideia da descrição de expressões faciais foi considerada de modo geral útil, assim como a possibilidade de guardar suas informações junto às fotos tiradas. Uma preocupação apresentada foi o tamanho do detector de obstáculos que dificultaria seu uso e a limitação do raio de abrangência do sensor.

Entre sugestões e críticas, os membros do grupo se mostraram contentes com um projeto que pudesse vir a auxiliá-los, o que é extremamente gratificante e incentiva a continuidade do trabalho, algo que segundo eles próprios, muitas vezes não ocorre após estudantes extraírem informações deles e seus projetos serem apresentados.

6.4 Disciplinas base

Foi possível por meio desse trabalho aplicar conceitos de algumas das disciplinas do curso de Engenharia de Computação. Essas disciplinas são apresentadas na lista a seguir e as que estão marcadas com (*) foram cursadas na Northern Arizona University - EUA, durante intercâmbio pelo programa Ciência Sem Fronteiras.

- Laboratório de Física: Métodos de medidas físicas e amostragem.
- Circuitos Elétricos: Modelagem de circuitos elétricos.
- Engenharia de software: Extração de requisitos, planejamento de testes e codificação do sistema.
- Programação Orientada a Objetos: Conceitos de orientação a objetos e linguagem Java.
- Estatística: Amostragem e cálculo de incerteza
- Laboratório de Circuítos Eletrônicos: Medidas de variáveis elétricas e construção de circuitos eletrônicos.
- Engineering Design - The Process*: Planejamento e implementação de projetos de automação utilizando Arduíno e sensores.
- Pattern Recognition*: Projetos e aplicações de reconhecimento de padrão.
- Multimídia e Hipermídia: Codificação em XML e desenvolvimento de layout.
- Microprocessadores e Aplicações II: Visão computacional, Integração entre sistemas embarcados e desenvolvimento Android.

6.5 Trabalhos futuros

Com a intenção de continuidade do projeto, alguns pontos que necessitam seja de melhoria, seja de correção, devem ser tratados.

- Número de sensores utilizados pelo circuito: Mais sensores permitiriam maior acurácia e precisão das distâncias retornadas pelo sistema.

- Tamanho do dispositivo: A fabricação do circuito diretamente em uma placa eletrônica dedicada, e de tamanho reduzido permitiria maior usabilidade ao usuário.

- Taxa de uso da API: Do lado do aplicativo, é indispensável buscar maneiras de não repassar ao usuário o valor que é cobrado pela utilização da API, quando o limite é ultrapassado. Uma possibilidade seria aplicar uma solução diferente ao problema, deixando de lado a API da Google. A utilização de OpenCV poderia ser uma solução mais adequada ao problema, porém demandaria esforços para implementação de algoritmos complexos de visão computacional. Outra possibilidade seria a inserção de propaganda como um modo de compensar essa cobrança.

- Integração com redes sociais: Com as redes sociais, em especial o Facebook e o SnapChat, desempenhando importante papel na comunicação entre as pessoas, integrar funcionalidade de compartilhamento das descrições poderia contribuir ainda mais com a acessibilidade de outras pessoas com deficiência visual.

- Expansão para outras plataformas: é importante considerar que existem usuários com smartphones rodando sistemas operacionais diferentes do Android. Uma possibilidade seria migrar o aplicativo para essas plataformas.

- Elaboração de testes de usabilidade com usuários para avaliação e melhorias das funções existentes, e inserção de novas, e testes de compatibilidade com as variadas versões do Android.

- Inserção de entrada por comando de voz ao aplicativo para tornar a navegação mais fácil.

Referências

[1] “World Development Report 2016: Digital Dividends,” The World Bank, Washington DC - EUA, Relatório Técnico, 2016.

[2] “Ericsson Mobility Report ON THE PULSE OF THE NETWORKED SOCIETY,” https://www.ericsson.com/res/docs/2016/ericsson-mobility-report-2016.pdf, Ericsson, Estocolmo, Suécia, Relatório Técnico, Junho de 2016.

[3] “TapTapSee,” http://www.taptapseeapp.com/, Acesso em: 01 de agosto de 2016.

[4] “BeMyEyes,” http://www.bemyeyes.org/, Acesso em: 01 de agosto de 2016.

[5] M. Avila, K. Wolf, A. Brock, e N. Henze, “Remote Assistance for Blind Users in Daily Life: A Survey about Be My Eyes.” Corfu Island, Grécia: The 9th ACM International Conference on Pervasive Technologies Related to Assistive Environments - PETRA’16, 2016.

[6] H. Kwak e J. An, “Revealing the Hidden Patterns of News Photos: Analysis of Millions of News Photos through GDELT and Deep Learning-Based Vision APIs.” Qatar: The Workshops of the Tenth International AAAI Conference on Web and Social Media, 2016.

[7] C. Wong, D. Wee, I. Murray, e T. Dias, “A Novel Design of Integrated Proximity Sensors for the White Cane.” Austrália: Seventh Australian and New Zealand Inteligent Information System Conference, Novembro de 2001, pp. 197–201.

[8] B. Leduc-Mills, H. Profita, S. Bharadwaj, P. Cromer, e R. Han, “ioCane: A Smart-Phone and Sensor-Augmented Mobility Aid for the Blind,” University of Colorado Boulder, Boulder, Colorado - EUA, Relatório Técnico, 2013.

[9] “CPqD Alcance,” https://www.cpqd.com.br/solucoes/cpqd-alcance/, Acesso em: 01 de agosto de 2016.

[10] “Projeto de auxílio às pessoas com deficiência visual vence concurso Intel de tecnologia,” http://napsol.icmc.usp.br/pt-br/node/272, Acesso em: 12 de novembro de 2016.

[11] M. A. Gallani, “Dispositivo microcontrolado para auxílio na orientação de deficientes visuais,” Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, São Paulo, Relatório Técnico, 2015.

[12] “Bengala Longa Eletrônica,” http://www.fapesc.sc.gov.br/entregues-a-cegos-bengalas-eletronicas-criadas-por-professor-da-univali/, Acesso em: 01 de agosto de 2016.

[13] “Bengala Automática,” http://www.correio24horas.com.br/detalhe/noticia/estudantes-baianos-criam-bengala-automatica-de-baixo-custo-para-deficientes-visuais-entenda/?cHash=4c9bbb1b6f2462e61435cbc1e7fa16ac, Acesso em: 01 de agosto de 2016.

[14] “Pesquisa Nacional de Saúde 2013: Ciclos de Vida,” IBGE, Rio de Janeiro - Brasil, Relatório Técnico, 2015.

[15] “Human Development Reports,” http://hdr.undp.org/en/countries/profiles/BRA, United Nations Development Programme, Relatório Técnico, Acesso em: 09 de agosto de 2016.

[16] Paul Deitel, Harvey Deitel e Alexander Wald, *Android 6 for Programmers: An App-Driven Aproach*, Terceira ed. Pearson Education, Inc., 2016.

[17] Martin Evans, Joshua Noble e Jordan Hochenbaum, *Arduino in Action*, Third ed. Greenwich, Connecticut - EUA: Manning Publications Co., 2013.

[18] “Tecnologia Assistiva,” http://www.pessoacomdeficiencia.gov.br/app/sites/default/files/publicacoes/livro-tecnologia-assistiva.pdf, Secretaria Especial dos Direitos Humanos, Brasília, Relatório Técnico, 2009.

[19] S. L. Henry, S. Abou-Zahra, e J. Brewer, “The Role of Accessibility in a Universal Web,” 11th Web for All Conference. Seoul, República da Coréia: Association for Computing Machinery, 2014.

[20] “Accessibility,” https://developer.android.com/guide/topics/ui/accessibility/index.html, Acesso em: 3 de setembro de 2016.

[21] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. H. Katz, A. Konwinski, G. Lee, D. A. Patterson, A. Rabkin, I. Stoica, e M. Zaharia, “Above the Clouds: A Berkeley View of Cloud Computing,” Electrical Engineering and Computer Sciences, University of California at Berkeley, Berkeley, Califórnia - EUA, Relatório Técnico, Fevereiro de 2009.

[22] Parker, J. R., *Algorithms for Image Processing and Computer Vision*, Segunda ed. Indianápolis, Indiana - EUA: Wiley Publishing, Inc., 2011.

[23] Catherine Thinus-Blanc e Florence Gaunet, “Representation of Space in Blind Persons: Vision as a Spatial Sense?” in *Psychological Bulletin*, Marseille, França, 1997, vol. 121, no. 1, pp. 20–42.

[24] Bo N Schenkman e Mats E Nilsson, “Human echolocation: Blind and sighted persons’ ability to detect sounds recorded in the presence of a reflecting object,” in *Perception*, Estocolmo, Suécia, 2010, vol. 39, no. 1, pp. 483 – 501.

[25] S. K. Kane, C. Jayant, J. O. Wobbrock, e R. E. Ladner, “Freedom to Roam: A Study of Mobile Device Adoption and Accessibility for People with Visual and Motor Disabilities ,” 11th international ACM SIGACCESS conference on Computers and accessibility, Seattle, Washington - EUA, pp. 115–122, 2009.

[26] “Best Practices,” https://cloud.google.com/vision/docs/image-best-practices, Acesso em: 12 de agosto de 2016.

[27] K. Sato e R. Sakai, “Explore the Galaxy of images with Cloud Vision API,” https://cloud.google.com/blog/big-data/2016/05/explore-the-galaxy-of-images-with-cloud-vision-api, Acesso em: 09 de agosto de 2016.

[28] G. Chacra, “EUA abandonam ensino da letra de mão,” http://www.estadao.com.br/noticias/geral,eua-abandonam-ensino-da-letra-de-mao-imp-,746256, Acesso em: 15 de novembro de 2016.

Apêndice A Diagrama de classes do aplicativo Android

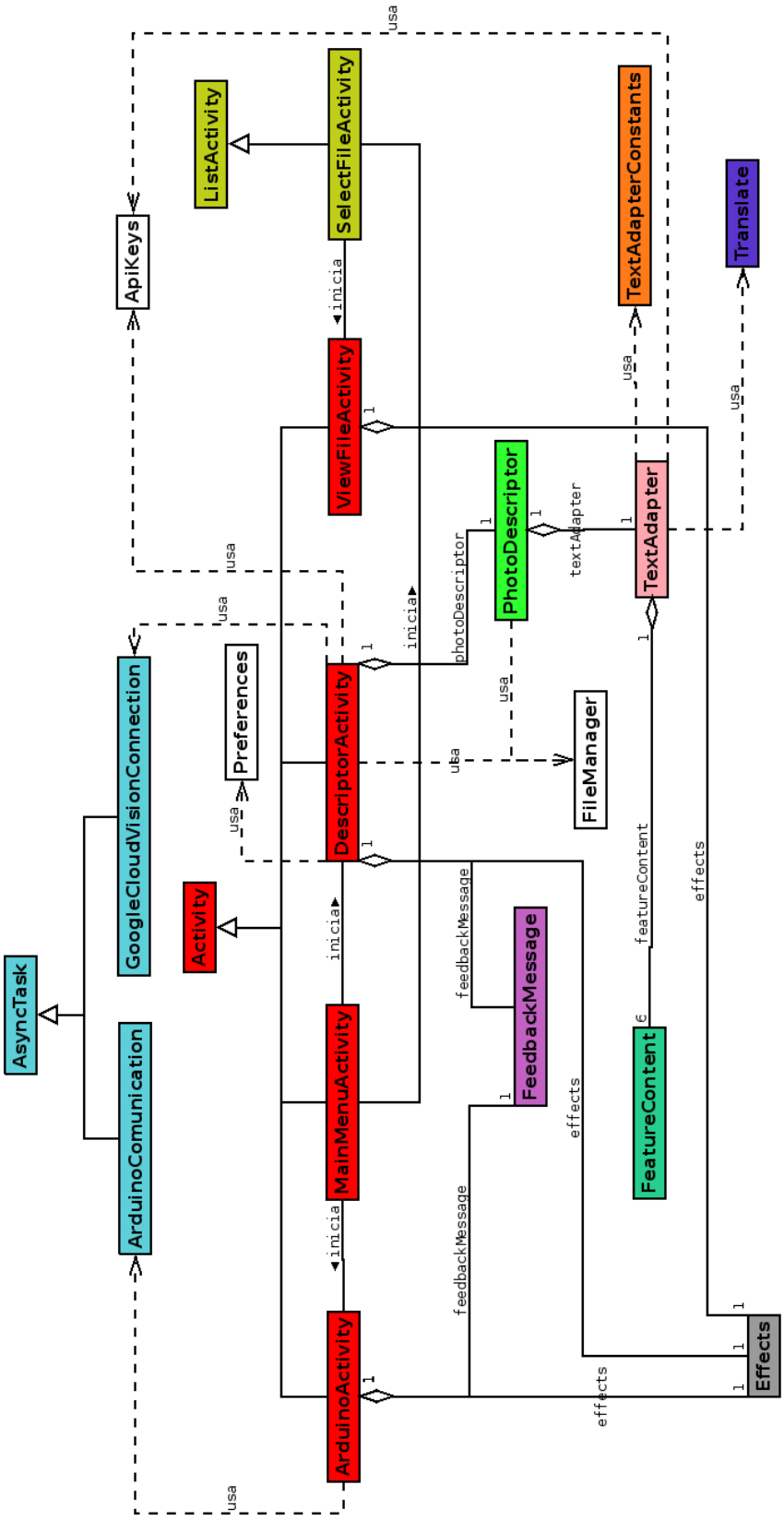


Figura A.1: Diagrama de classes do aplicativo baseado em Android

Apêndice B Diagrama de fluxo do aplicativo Android

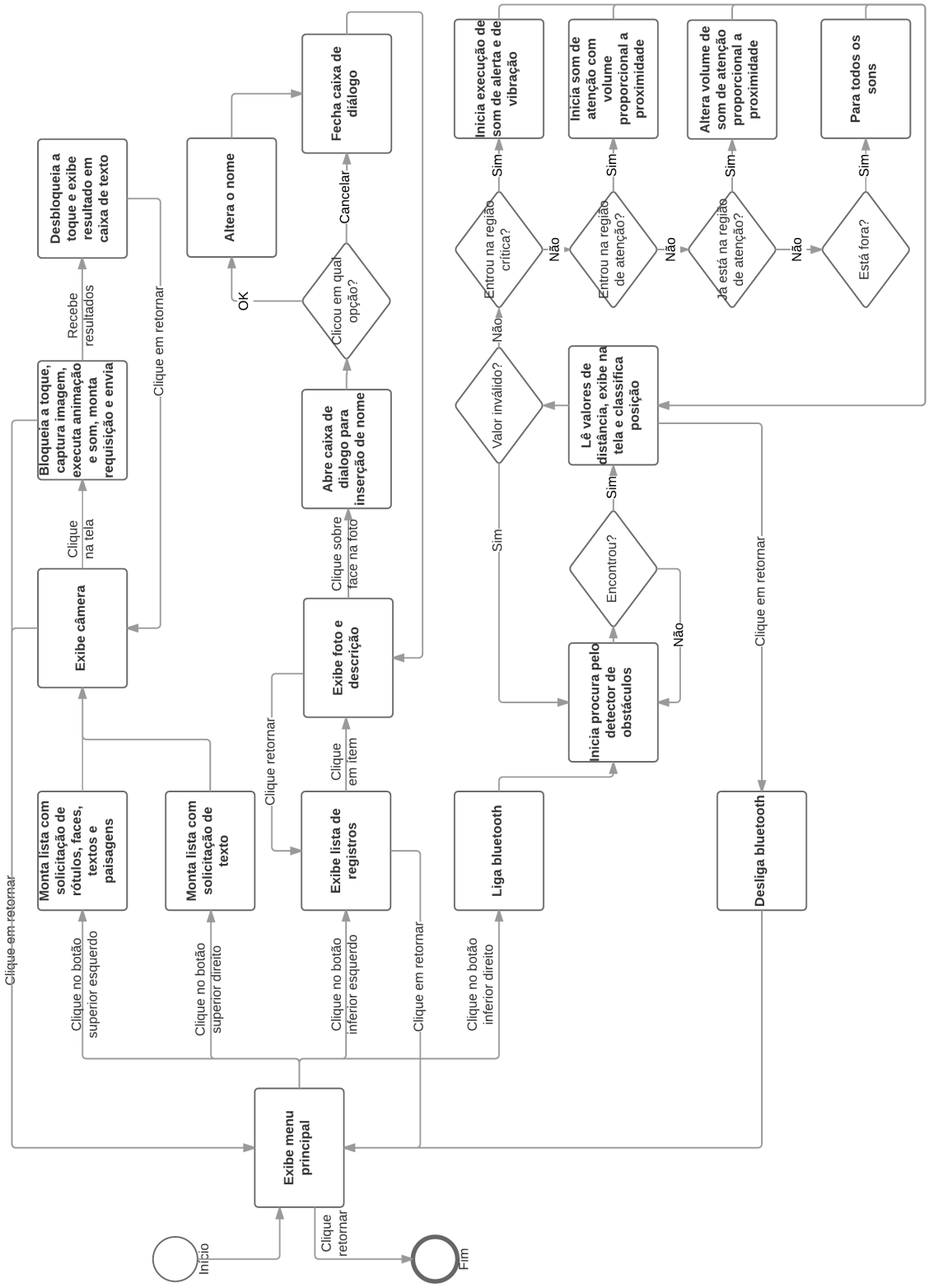


Figura B.1: Diagrama de fluxo do aplicativo baseado em Android

Apêndice C Códigos relevantes

Devido a dimensão do projeto, a exposição de todo o código utilizado como anexo dessa dissertação mostrou-se inviável. Por essa razão, foram selecionados para serem fazerem parte desse documento apenas os códigos considerados chave para o funcionamento do sistema. O código completo pode ser encontrado em `https://github.com/guilherme-siqueira/projeto_final`. No Código C.1, encontra-se a rotina responsável pela adaptação textual da descrição de uma imagem. Nela, considera-se primeiramente se há alguma paisagem identificada, para então avaliar possíveis faces, rótulos ou textos presentes. Caso não haja, o campo é ignorado, e apenas os campos mais internos, faces, rótulos e textos, são considerados. Obter o número de faces, em vez de apenas saber se há ou não faces, permite que o texto possua concordância verbal. No caso de não haver faces na imagem, considera-se apenas a possível presença de rótulos e textos.

```
1  private void writeTextualDescription() {
2    if (landmarkElement.getText() != null) {
3       landmarkElement.translate();
4       textualDescription = LANDMARK_INTRO + landmarkElement.getText() + ". ";
5       if (nFaces == 1)
6          textualDescription += FACE_INTRO + ONE_FACE + faceElement.getText();
7       else if (nFaces == 2)
8          textualDescription += FACE_INTRO + TWO_FACES + faceElement.getText();
9       else if (nFaces > 2)
10         textualDescription += FACE_INTRO + MORE_FACES_BEGINNING + nFaces +
               MORE_FACES_END + faceElement.getText();
11
12      if (labelElement.getText() != null) {
13         labelElement.translate();
14         textualDescription += LABEL_INTRO + labelElement.getText();
15      }
16
17      if (textElement.getText() != null) {
18         textualDescription += TEXT_INTRO + textElement.getText();
19      }
```

```
20    }
21    else if (nFaces != 0) {
22       textualDescription = FACE_INTRO;
23       if (nFaces == 1)
24          textualDescription += ONE_FACE + faceElement.getText();
25       else if (nFaces == 2)
26          textualDescription += TWO_FACES + faceElement.getText();
27       else if (nFaces > 2)
28          textualDescription += MORE_FACES_BEGINNING + nFaces + MORE_FACES_END +
                faceElement.getText();
29
30       if (labelElement.getText() != null) {
31          labelElement.translate();
32          textualDescription += LABEL_INTRO + labelElement.getText();
33       }
34
35       if (textElement.getText() != null) {
36          textualDescription += TEXT_INTRO + textElement.getText();
37       }
38    }
39    else if (labelElement.getText() != null) {
40       labelElement.translate();
41       textualDescription = LABEL_INTRO + labelElement.getText();
42       if (textElement.getText() != null) {
43          textualDescription += TEXT_INTRO + textElement.getText();
44       }
45    }
46    else if (textElement.getText() != null) {
47       textualDescription = TEXT_INTRO + textElement.getText();
48    }
49 }
```

Código C.1: Rotina de construção do texto da descrição da imagem

No código C.2, adaptado do exemplo fornecido pelos tutoriais da Google Cloud Vision [26], encontra-se o método responsável pelo envio da imagem e recebimento de sua descrição. Primeiramente uma instância das funcionalidades da API da Google é criada e construída, por meio da classe Vision. Em seguida é criada uma instância de um objeto que permite múltiplas requisições de imagens, contendo a imagem capturada pelo aplicativo devidamente comprimida, o idioma para possíveis textos escritos, e a lista de diferentes solicitações requisitadas. Então, um objeto de requisição de anotação recebe todas as preferências criadas é executado, e o resultado retornado é recolhido para posterior conversão em linguagem fluente, a partir da classe PhotoDescriptor.

```
1  protected boolean connect(){
2    try {
3       HttpTransport httpTransport = AndroidHttp.newCompatibleTransport();
4       JsonFactory jsonFactory = GsonFactory.getDefaultInstance();
5
6       Vision.Builder builder = new Vision.Builder(httpTransport, jsonFactory,
            null);
7       builder.setVisionRequestInitializer(new
8       VisionRequestInitializer(CLOUD_VISION_API_KEY));
9       Vision vision = builder.build();
10
11      BatchAnnotateImagesRequest batchAnnotateImagesRequest =
12      new BatchAnnotateImagesRequest();
13      batchAnnotateImagesRequest.setRequests(new
            ArrayList<AnnotateImageRequest>() {{
14      AnnotateImageRequest annotateImageRequest = new AnnotateImageRequest();
15
16      Image base64EncodedImage = new Image();
17      ByteArrayOutputStream byteArrayOutputStream = new ByteArrayOutputStream();
18
19      bitmap.compress(Bitmap.CompressFormat.JPEG, 90, byteArrayOutputStream);
20      byte[] imageBytes = byteArrayOutputStream.toByteArray();
21      base64EncodedImage.encodeContent(imageBytes);
22      annotateImageRequest.setImage(base64EncodedImage);
```

```
23
24        ImageContext imageContext = new ImageContext();
25        String [] languages = { "pt-BR" };
26        imageContext.setLanguageHints(Arrays.asList(languages));
27        annotateImageRequest.setImageContext(imageContext);
28        annotateImageRequest.setFeatures(requestsArrayList);
29        add(annotateImageRequest);
30        }});
31
32        Vision.Images.Annotate annotateRequest =
33        vision.images().annotate(batchAnnotateImagesRequest);
34
35        annotateRequest.setDisableGZipContent(true);
36        Log.d(TAG, "created Cloud Vision request object, sending
              request");
37
38        BatchAnnotateImagesResponse response = annotateRequest.execute();
39
40        photoDescriptor.callTextAdaptation(response);
41
42        return true;
43    }
44    catch (GoogleJsonResponseException e) {
45        Log.d(TAG, "failed to make API request because " + e.getContent());
46    }
47    catch (IOException e) {
48        Log.d(TAG, "failed to make API request because of other
              IOException " + e.getMessage());
49    }
50    return false;
51 }
```

Código C.2: Rotina de conexão e envio de dados para a Google Cloud Vision API

Já o Código C.3 contém a rotina de captura dos sinais vindos do sensor de obstáculos HC-SR04, cálculo da distância, e o envio para a porta serial, onde o módulo Bluetooth HC-05 é conectado. Primeiramente, define-se a taxa de transmissão da porta serial, o pino de saída de sinal para emissão de onda pelo sensor, e o pino de entrada, para recebimento do sinal correspondente ao tempo do eco refletido.

No ciclo principal, monta-se a onda a ser emitida, com nível baixo de 2ms, e alto, de 10ms, respeitando as especificações mínimas de utilização do sensor, que podem ser encontradas no datasheet no Anexo II . Em seguida lê-se o valor obtido de eco na porta de entrada, e a distância do obstáculo é calculada com base na equação $S = V \times t$, em que V representa a velocidade do som no ar em centímetros por segundo, 0.034cm/s, e t é a metade do tempo entre a emissão e a recepção do sinal da onda, ou seja, o tempo entre a emissão da onda e o encontro com o obstáculo. Por fim, verifica-se se o valor lido não ultrapassou o limite de precisão do sensor, de 4 metros, com o intuito de evitar valores pouco confiáveis, e a cada um segundo a rotina amostra o sinal e o envia para a porta serial, onde a entrada do módulo Bluetooth está conectada.

```
1  #define pino_trigger 4
2  #define pino_echo 5
3
4  long duration;
5  int distance;
6
7  void setup()
8  {
9    Serial.begin(9600);
10   pinMode(pino_trigger, OUTPUT);
11   pinMode(pino_echo, INPUT);
12 }
13
14 void loop()
15 {
16   digitalWrite(pino_trigger, LOW);
17   delayMicroseconds(2);
18   digitalWrite(pino_trigger, HIGH);
19   delayMicroseconds(10);
```

```
20    digitalWrite(pino_trigger, LOW);
21    duration = pulseIn(pino_echo, HIGH);
22    // Calculating the distance
23    distance= duration*0.034/2;
24
25    if(distance < 400)
26    {
27       Serial.println(distance);
28       delay(100);
29    }
30 }
```

Código C.3: Rotina executada na placa Arduíno

Anexo I Especificação técnica HC-05

HC-05

-Bluetooth to Serial Port Module

Overview

HC-05 module is an easy to use Bluetooth SPP (Serial Port Protocol) module, designed for transparent wireless serial connection setup.
Serial port Bluetooth module is fully qualified Bluetooth V2.0+EDR (Enhanced Data Rate) 3Mbps Modulation with complete 2.4GHz radio transceiver and baseband. It uses CSR Bluecore 04-External single chip Bluetooth system with CMOS technology and with AFH(Adaptive Frequency Hopping Feature). It has the footprint as small as 12.7mmx27mm. Hope it will simplify your overall design/development cycle.

Specifications

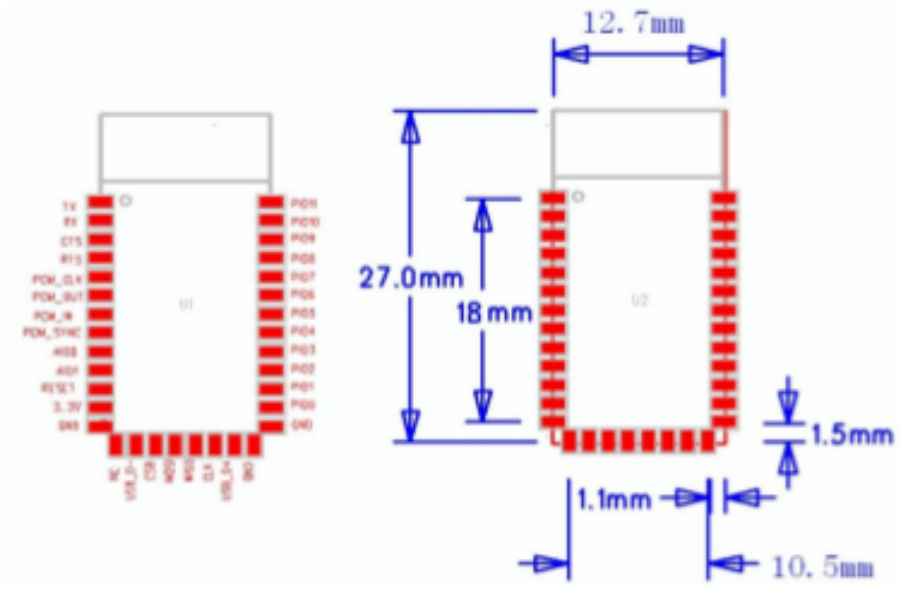
Hardware features

- Typical -80dBm sensitivity
- Up to +4dBm RF transmit power
- Low Power 1.8V Operation ,1.8 to 3.6V I/O
- PIO control
- UART interface with programmable baud rate
- With integrated antenna
- With edge connector

Software features

- Default Baud rate: 38400, Data bits:8, Stop bit:1,Parity:No parity, Data control: has. Supported baud rate: 9600,19200,38400,57600,115200,230400,460800.
- Given a rising pulse in PIO0, device will be disconnected.
- Status instruction port PIO1: low-disconnected, high-connected;
- PIO10 and PIO11 can be connected to red and blue led separately. When master and slave are paired, red and blue led blinks 1time/2s in interval, while disconnected only blue led blinks 2times/s.
- Auto-connect to the last device on power as default.
- Permit pairing device to connect as default.
- Auto-pairing PINCODE:"0000" as default
- Auto-reconnect in 30 min when disconnected as a result of beyond the range of connection.

Hardware



Anexo II Especificação técnica HC-SR04

Ultrasonic Ranging Module HC - SR04

Product features:

Ultrasonic ranging module HC - SR04 provides 2cm - 400cm non-contact measurement function, the ranging accuracy can reach to 3mm. The modules includes ultrasonic transmitters, receiver and control circuit. The basic principle of work:
(1) Using IO trigger for at least 10us high level signal,
(2) The Module automatically sends eight 40 kHz and detect whether there is a pulse signal back.
(3) IF the signal back, through high level , time of high output IO duration is the time from sending ultrasonic to returning.
Test distance = (high level timexvelocity of sound (340M/S) / 2,

Wire connecting direct as following:

- 5V Supply
- Trigger Pulse Input
- Echo Pulse Output
- 0V Ground

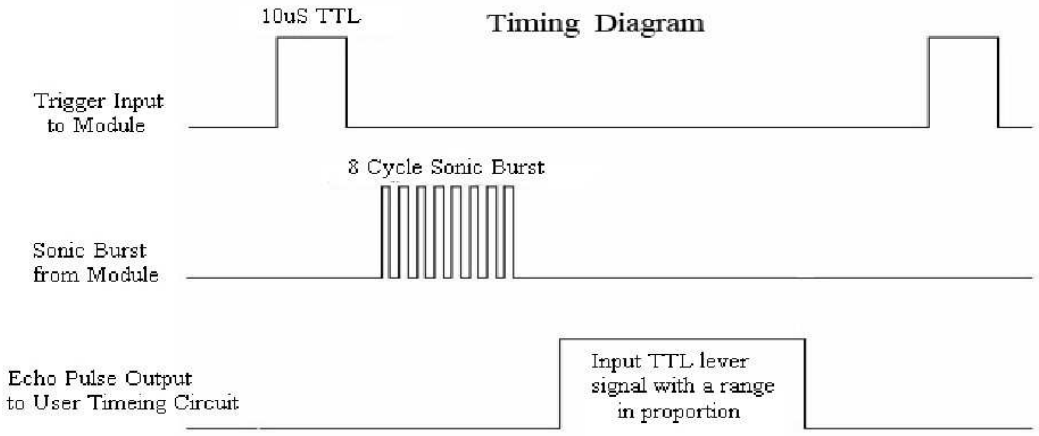
Electric Parameter

Working Voltage	**DC 5 V**
Working Current	**15mA**
Working Frequency	**40Hz**
Max Range	**4m**
Min Range	**2cm**
MeasuringAngle	**15 degree**
Trigger Input Signal	**10uS TTL pulse**
Echo Output Signal	**Input TTL lever signal and the range in proportion**
Dimension	**45*20*15mm**

Vcc Trig Echo GND

Timing diagram

The Timing diagram is shown below. You only need to supply a short 10uS pulse to the trigger input to start the ranging, and then the module will send out an 8 cycle burst of ultrasound at 40 kHz and raise its echo. The Echo is a distance object that is pulse width and the range in proportion .You can calculate the range through the time interval between sending trigger signal and receiving echo signal. Formula: uS / 58 = centimeters or uS / 148 =inch; or: the range = high level time * velocity (340M/S) / 2; we suggest to use over 60ms measurement cycle, in order to prevent trigger signal to the echo signal.



Attention:

- The module is not suggested to connect directly to electric, if connected electric, the GND terminal should be connected the module first, otherwise, it will affect the normal work of the module.
- When tested objects, the range of area is not less than 0.5 square meters and the plane requests as smooth as possible, otherwise ,it will affect the results of measuring.