

UNIVERSIDADE FEDERAL FLUMINENSE
CAMPUS UNIVERSITÁRIO DE RIO DAS OSTRAS
INSTITUTO DE CIÊNCIA E TECNOLOGIA
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

DIANA GAMA SILVA FRAGA

MicroBlind: Aplicativo para auxiliar deficientes visuais no uso do micro-ondas

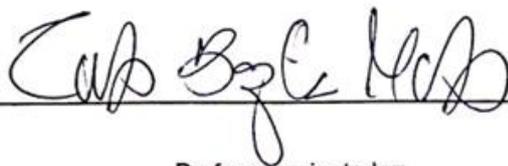
Rio das Ostras - RJ

2019

Diana Gama Silva Fraga

MicroBlind: Aplicativo para auxiliar deficientes visuais no uso do micro-ondas

Monografia apresentada ao curso de Bacharelado em Ciência da Computação da Universidade Federal Fluminense – Rio das Ostras, como pré-requisito para a obtenção do Grau de Bacharel.

A handwritten signature in black ink, appearing to read 'Carlos Bazilio Martins', is positioned above a horizontal line.

Professor orientador:

CARLOS BAZILIO MARTINS

Rio de Janeiro

2019

F811m Fraga, Diana Gama Silva
MicroBlind: Aplicativo para auxiliar deficientes visuais
no uso do micro-ondas / Diana Gama Silva Fraga ; Carlos Ba-
zilio Marins, orientador. 2019.

Trabalho de Conclusão de Curso (Graduação em Computa-
ção)-Universidade Federal Fluminense, Rio das Ostras,
2019.

1. Deficiência Visual. 2. Acessibilidade. 3. Aplicativo. 4.
Android. I. Marins, Carlos Bazilio, orientador. II. Universidade
Federal Fluminense. Rio das Ostras, 2019. III. Título.

CDD -

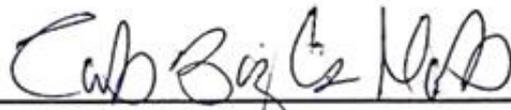
Bibliotecário responsável: Maria Dalva Pereira de Souza - CRB7/7044

MicroBlind: Aplicativo para auxiliar deficientes visuais no uso do micro-ondas

Trabalho de conclusão de curso apresentado ao curso Bacharelado em Ciência da Computação, como requisito parcial para conclusão do curso.

Aprovado em ... 11 de julho ... de 2019.

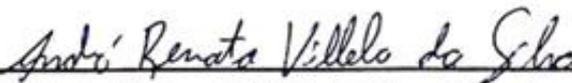
BANCA EXAMINADORA



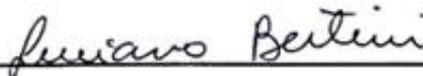
Prof. Carlos Bazilio Martins – Orientador
Universidade Federal Fluminense



Prof. Leandro Soares de Sousa – Avaliador
Universidade Federal Fluminense



Prof. André Renato Villela da Silva - Avaliador
Universidade Federal Fluminense



Prof. Luciano Bertini- Avaliador
Universidade Federal Fluminense

AGRADECIMENTOS

Agradeço primeiramente a Deus, pois sem Ele nada disso seria possível. À minha família, pelo apoio e incentivo de sempre. Aos amigos que fizeram a graduação mais leve e divertida, mesmo com todas as dificuldades. Agradeço também ao professor e orientador Carlos Bazilio, pela confiança, por acreditar na ideia e dar todo o suporte necessário para a realização desse projeto. Ao professor Luciano, pela disponibilidade de sempre, pelo tempo e materiais cedidos, através da FAPERJ, para parte importante do trabalho. Sou muito grata por isso. Obrigada.

Obrigada também, à Universidade Federal Fluminense, e todos os professores que participaram e contribuíram para minha formação profissional e pessoal. Ao IFF, e aos ensinamentos aprendidos no curso de Automação Industrial, por contribuírem com a elaboração da ideia do presente projeto.

“Aprender é a única coisa que a mente nunca se cansa, nunca tem medo e nunca se arrepende”.

Leonardo Da Vinci

Resumo

Os deficientes visuais enfrentam dificuldades para utilizar eletrodomésticos com independência, pois em sua maioria, estes apresentam apenas sinais visuais, teclados lisos e sem relevo, não permitindo o uso de outro sentido além da visão na sua utilização e muitas vezes sendo necessária a ajuda de uma pessoa que enxerga para realizar tal atividade. Diante disso, este trabalho propõe o desenvolvimento de um aplicativo para *smartphone* baseado em Android, a fim de auxiliar pessoas portadoras de deficiência visual no uso do micro-ondas. A ideia é usar a tecnologia como ferramenta para gerar acessibilidade e contribuir com a inclusão social, diminuindo, assim, as barreiras encontradas por esse público no uso do micro-ondas. A fim de atingir o objetivo, foi necessário realizar a adaptação do micro-ondas para receber comandos via *bluetooth*.

Palavras-chave: Android, acessibilidade, deficiente visual, micro-ondas, *bluetooth*.

ABSTRACT

The visually impaired face difficulties to use home appliances with independence, since in the majority, these present only visual signs, flat keyboards without relief, not allowing the use of another sense beyond the vision in its use and often being necessary the help of a person who sees to perform such activity. Therefore, this project proposes the development of an application for smartphone based on Android, in order to assist visually impaired people in the use of the microwave. The idea is to use technology as a tool to generate accessibility and contribute to social inclusion, thus reducing the barriers encountered by this public in the use of the microwave. In order to reach the goal, it was necessary to adapt the microwave to receive the commands through bluetooth.

Keywords: Android, accessibility, visually impaired, microwave, bluetooth.

Lista de figuras

Figura 1: População brasileira em 2010 por tipo de deficiência	15
Figura 2: Exemplo de teclado de um micro-ondas	16
Figura 3: Interface do aplicativo BlindTool	28
Figura 4: Interface do aplicativo Be My Eyes	29
Figura 5: Interface do aplicativo CPqD Alcance	30
Figura 6: Interface do aplicativo Ubook	31
Figura 7: Interface do aplicativo Áudio Acordes	32
Figura 8: Modelos de placas de Arduíno	34
Figura 9: Módulo Bluetooth HC-06	34
Figura 10: Funcionamento do Relé	35
Figura 11: Módulo Relé de 8 canais	36
Figura 12: Tela do software Arduíno IDE	37
Figura 13: Estrutura do projeto	39
Figura 14: Micro-ondas Philco PME25	41
Figura 15: Placa de Conrole do micro-ondas	42
Figura 16: Fios soldados na placa de controle	42
Figura 17: Pinos da placa de controle conectados com os relés	43
Figura 18: Circuito Arduíno + Módulo Bluetooth	43
Figura 19: Divisor de tensão	44
Figura 20: Circuito completo	44
Figura 21: Constantes do programa	45
Figura 22: Método setup	46
Figura 23: Método loop	47
Figura 24: Permissões do aplicativo	47
Figura 25: Classe MainActivity	48
Figura 26: Método que verifica se o bluetooth está ligado - checkBTState()	49
Figura 27: Método responsável por repassar o endereço bluetooth para as outras Activities	49
Figura 28: Teclado mapeado	50
Figura 29: Função executada quando a tecla é acionada	50
Figura 30: Método sendData()	51
Figura 31: Criação da intent RecognizerIntent	52
Figura 32: Conversão do número por extenso	52

Figura 33: Exemplo de um delay antes de chamar o método sendData	53
Figura 34: Etapas para ligar o bluetooth no smartphone	54
Figura 35: Etapas para a ativação do Talkback no smartphone	54
Figura 36: Tela inicial do aplicativo	55
Figura 37: Lista de dispositivos bluetooth	56
Figura 38: Teclado virtual do aplicativo	56
Figura 39: Tela do comando de voz do aplicativo	57
Figura 40: Reconhecimento de voz do aplicativo	58
Figura 41: Exemplos do comando de voz	58

Lista de Tabelas

Tabela 1: Participantes do teste de usabilidade	62
Tabela 2: Perguntas feitas nos testes	63
Tabela 3: Resposta do usuário 1 - Luisa.....	64
Tabela 4: Resposta do usuário 2 - Elizabeth.....	64
Tabela 5: Resposta do usuário 3 - Dominique	65

Sumário

1	INTRODUÇÃO	13
1.1	Motivação e Justificativa	13
1.2	Objetivos	18
1.2.1.	Objetivo geral	18
1.2.2.	Objetivos específicos	18
1.3	Organização do texto	18
2	REFERENCIAL TEÓRICO	20
2.1	Deficiência Visual	20
2.2	Acessibilidade	21
2.2.1	Tecnologia Assistiva	21
2.3	Android	24
2.3.1	Acessibilidade no Android	25
3	TECNOLOGIA MÓVEL DISPONÍVEL PARA DEFICIENTES VISUAIS	27
4	TECNOLOGIAS UTILIZADAS	32
4.1	Circuito no micro-ondas	33
4.1.1	Arduíno	33
4.1.2	Módulo Bluetooth	34
4.1.3	Módulo Relé	35
4.1.4	Arduíno IDE	36
4.2	Aplicativo	37
4.2.1	Java	37
4.2.2	Android Studio	38
5	MONTAGEM DO CIRCUITO NO MICRO-ONDAS	41
5.1	Código do Arduíno	45
6	DESENVOLVIMENTO DO APLICATIVO E TESTES	50
6.1	Android Manifest	50
6.2	Classes	50
6.3	Configurações iniciais de utilização	56
6.4	Interface do aplicativo e seu funcionamento	58
6.5	Testes	62
7	CONCLUSÃO E TRABALHOS FUTUROS	66
8	REFERÊNCIAS	68

1 INTRODUÇÃO

1.1 Motivação e Justificativa

Com o avanço tecnológico, diversos produtos entram diariamente no cotidiano das pessoas. Esses produtos contam cada vez mais com designs modernos e com a popularização das telas *touchscreen* (telas sensíveis ao toque). Atualmente, as telas e teclados sensíveis ao toque estão presentes na maior parte de eletrônicos e eletrodomésticos. Como consequência, acabam impossibilitando o uso dessas tecnologias por uma parte da população: os deficientes visuais.

Embora a Lei Brasileira de Inclusão da Pessoa com Deficiência (LEI nº 13.146, Art. 74) [1] garanta o direito de acesso dos deficientes à produtos e recursos, com auxílio de tecnologias assistivas, a fim de maximizarem sua autonomia e qualidade de vida, não é o que se percebe na prática. Uma pesquisa realizada pela Fundação Procon de São Paulo, na qual 265 grandes empresas foram contatadas, mas apenas 35 concordaram em participar, apresentou resultados alarmantes sobre os produtos comercializados no quesito acessibilidade:

- 45,71% (16 empresas) declararam que seus produtos não possuem nenhuma adaptação;
- 25,71% (9 empresas) informaram que somente alguns produtos sofreram adaptação;
- 20% (7 empresas) acreditam que seus produtos não necessitam de nenhuma adaptação para o acesso dos portadores de necessidades especiais;
- 8,57% (3 empresas) informaram que todos os seus produtos são adaptados.

Das empresas que informaram possuir produtos adaptados:

- 66,67% das adaptações são na embalagem;
- 6,67% nas instruções de uso;
- 6,67% no formato do produto;
- 20% promoveram outras adaptações.

Das deficiências visadas por essas adaptações as empresas apontaram:

- 71,43% deficiência visual;
- 14,29 % física;
- 7,14% auditiva;
- 7,14% outras.

Segundo o Procon (Fundação de Proteção e Defesa do Consumidor), algumas empresas acreditam que seus produtos não necessitam de qualquer adaptação: o que pode significar ou que não foram demandados ou não perceberam a necessidade. A maioria das empresas procede alterações apenas nas embalagens dos produtos e não nos produtos em si. (FUNDAÇÃO PROCON, 2008)

[2]

De acordo com a Organização Mundial da Saúde, existem 39 milhões de pessoas cegas no mundo e outras 246 milhões têm baixa visão [3]. No Brasil, segundo o último Censo do IBGE realizado em 2010 [6], 8,3% da população brasileira apresenta pelo menos um tipo de deficiência severa. Sendo 3,46 % com deficiência visual. Esses números crescem cada vez mais ao passar dos anos. A estimativa apresentada pelo World Report on Disability (2010) [4], é que a cada 5 segundos, 1 pessoa se torna cega. Estima-se que, até 2020, o número de pessoas com deficiência visual poderá dobrar no mundo.

Figura 1: População brasileira em 2010 por tipo de deficiência



Fonte: Censo IBGE (2010)

Deste modo, verifica-se que o número de pessoas com deficiência no Brasil é relevante, e que a deficiência com maior ocorrência no país é a deficiência visual. Os deficientes visuais possuem necessidades de acessibilidade nos serviços, ambientes e tecnologias, para conseguirem realizar suas tarefas cotidianas e viverem com independência.

Levando em consideração o ambiente da cozinha, percebe-se que a maioria dos eletrodomésticos possuem sinalizadores visuais, como luzes coloridas, para

indicar se a comida está pronta. E alguns até utilizam sinais sonoros para indicar que algo ocorreu. Com relação ao micro-ondas, apenas isso não é o suficiente para o eletrodoméstico ser acessível aos deficientes visuais. Para um cego utilizar o micro-ondas, não basta ir à loja, comprar o eletrodoméstico, chegar em casa e ler o manual. Até porque, os manuais não são acessíveis para esse público. Eles não constam com versões em Braille ou em áudio, fazendo-se necessário o auxílio de uma outra pessoa ou de alguma ferramenta de leitura de textos.

Outra barreira que os deficientes visuais enfrentam com o micro-ondas, é o teclado extremamente liso e sem relevo algum indicando o que significam àquelas teclas. Os teclados seguem a tendência do avanço tecnológico, com *designs* cada vez mais modernos e com menos teclas, por consequência, constam cada vez menos com acessibilidade para os deficientes visuais.

Figura 2: Exemplo de teclado de um micro-ondas



Fonte: Google Inc.

Em relação à adaptação do eletrodoméstico, o relevo em Braille seria de grande valia para gerar acessibilidade. Porém, o teclado em Braille poderia deixar de fora uma parte do público alvo: pessoas que acabaram de ficar cegas e ainda não foram alfabetizadas nesse sistema.

O Sistema Braille¹ é extremamente importante e relevante para os deficientes visuais, pois proporciona maior independência. Porém, segundo a Fundação Dorina

Nowill para Cegos, apenas 10% dos deficientes visuais são alfabetizados em Braile no Brasil. Normalmente, o conhecimento da linguagem é comum quando a pessoa nasce cega e é educada desde cedo nesse sistema de escrita e leitura. Porém, na maioria dos casos, a cegueira é a consequência de alguma patologia ou trauma que acontece na vida adulta, como em decorrência da diabetes, por exemplo, o que compromete severamente a sensibilidade dos membros e dos sentidos, como o tato, dificultando ou impossibilitando o reconhecimento dos pontos do Braile. Nesses casos, outros recursos de acessibilidade tornam-se importantes, como o recurso do áudio, por exemplo.

É de extrema importância fazer com que os deficientes não se sintam estranhos na sociedade, mas sim sejam incluídos. Para que isso ocorra, certa adaptação no mundo se faz necessária. Diante de um mundo moderno, consciente sobre tais questões, a tecnologia se apresenta como uma poderosa ferramenta para criar soluções de acessibilidade para os deficientes em diversas tarefas e, deveria ser uma obrigação dos agentes que a possuem, aplicá-la de maneira a atender as pessoas com necessidades especiais. Isto posto, este trabalho tem como foco desenvolver um aplicativo para *smartphones* Android, a fim de auxiliar os deficientes visuais no ambiente doméstico, visando o uso do micro-ondas. Para atingir o objetivo, foi necessária a adaptação do micro-ondas para que este pudesse receber comandos via *bluetooth* através do aplicativo desenvolvido. A adaptação foi feita interferindo no circuito dentro do micro-ondas, com a ajuda de componentes como: arduino, módulo *bluetooth* e alguns relés, que serão explicados no decorrer do trabalho.

¹O Braile é um sistema de escrita tátil utilizado por pessoas cegas ou com baixa visão. É tradicionalmente escrito em papel em relevo

1.2 Objetivos

Os objetivos deste trabalho são divididos entre objetivo geral e objetivos específicos.

1.2.1. Objetivo geral

O objetivo geral desse projeto é a criação e o desenvolvimento de um aplicativo baseado em Android para *smartphones*, com o intuito de auxiliar os deficientes visuais na utilização do micro-ondas. A ideia é criar uma ferramenta de acessibilidade para proporcionar uma maior independência para esse público.

1.2.2. Objetivos específicos

- Apresentar os conceitos de deficiência visual e acessibilidade;
- Conceituar e explicar de forma breve o sistema operacional Android e apresentar o que consta no mesmo no quesito acessibilidade para deficientes visuais;
- Listar alguns aplicativos para Android já existentes que proporcionam acessibilidade para o deficiente visual na realização de alguma tarefa;
- Apresentar e explicar a adaptação feita no micro-ondas;
- Apresentar as etapas de desenvolvimento do aplicativo, as tecnologias utilizadas e seu funcionamento.

1.3 Organização do texto

O presente trabalho está organizado em capítulos e será desenvolvido da seguinte maneira:

- **Capítulo 1:** apresenta uma introdução do trabalho, a motivação e justificativa da escolha do tema e os objetivos do projeto;
- **Capítulo 2:** este capítulo explica alguns conceitos como deficiência visual, acessibilidade, tecnologia assistiva, Android e apresenta o que existe de acessibilidade no Android para deficientes visuais;
- **Capítulo 3:** estuda algumas tecnologias assistivas (em forma de aplicativos) já existentes na plataforma Android, com o objetivo de ajudar o deficiente visual na realização de alguma tarefa;
- **Capítulo 4:** neste capítulo é descrita a proposta e modelagem do aplicativo como ferramenta de auxílio aos deficientes visuais no uso do micro-ondas, apresentando as tecnologias utilizadas;
- **Capítulo 5:** neste capítulo é apresentada a montagem do circuito no micro-ondas;
- **Capítulo 6:** este capítulo expõe o desenvolvimento do aplicativo e seus testes e resultados;
- **Capítulo 7:** por fim, no último capítulo serão apresentadas as conclusões deste trabalho e os trabalhos futuros propostos.

2 REFERENCIAL TEÓRICO

2.1 Deficiência Visual

De acordo com a Convenção sobre os Direitos das Pessoas com Deficiência adotada pela Organização das Nações Unidas, pessoas com deficiência são aquelas que têm impedimentos de natureza física, intelectual ou sensorial, os quais, em interação com diversas barreiras, podem obstruir sua participação plena e efetiva na sociedade com as demais pessoas (ONU, 2006) [5]. No Brasil, a deficiência mais comum que atinge a população é a deficiência visual. (CENSO IBGE, 2010) [6]

A deficiência visual pode ser definida como perda total ou parcial, congênita ou adquirida, da visão. Como o nível de acuidade visual pode variar, as deficiências visuais são classificadas em dois grupos: cegueira ou visão subnormal. (MORCELLI, 2014) [7].

Segundo a Fundação Dorina Nowill para Cegos [8], cegueira e visão subnormal podem ser definidas como:

- Cegueira: há perda total da visão ou pouquíssima capacidade de enxergar, o que leva a pessoa a necessitar do Sistema Braille como meio de leitura e escrita.
- Baixa visão ou visão subnormal: caracteriza-se pelo comprometimento do funcionamento visual dos olhos, mesmo após tratamento ou correção. As pessoas com baixa visão podem ler textos impressos ampliados ou com uso de recursos óticos especiais.

Entre os dois extremos da capacidade visual estão situadas patologias como miopia, estrabismo, astigmatismo, ambliopia, hipermetropia, que não constituem necessariamente deficiência visual, mas que na infância devem ser identificadas e tratadas o mais rápido possível, pois podem interferir no processo de desenvolvimento e na aprendizagem. (GIL, 2000) [9]

2.2 Acessibilidade

O termo acessibilidade significa incluir a pessoa com deficiência na participação de atividades como o uso de produtos, serviços e informações. (GOVERNO DO BRASIL, 2017) [10]

A Secretaria Especial dos Direitos da Pessoa com Deficiência explica que acessibilidade é um atributo essencial do ambiente que garante a melhoria da qualidade de vida das pessoas. Deve estar presente nos espaços, no meio físico, no transporte, na informação e comunicação, bem como em outros serviços e instalações abertos ao público ou de uso público, tanto na cidade como no campo.

Apesar de ser um tema ainda pouco difundido, a implementação da acessibilidade é fundamental, considerando que ela gera resultados sociais positivos e contribui para o desenvolvimento inclusivo e sustentável, a fim de possibilitar à pessoa com deficiência viver de forma independente e participar plenamente de todos os aspectos da vida. [11]

2.2.1 Tecnologia Assistiva

Tecnologia assistiva é um termo ainda novo, utilizado para identificar recursos e serviços que contribuem para proporcionar ou ampliar habilidades funcionais de pessoas com deficiência e, conseqüentemente, promover vida independente e inclusão social.

A Fundação Dorina Nowill define recursos e serviços da seguinte maneira:

- Recursos: todo e qualquer item, equipamento ou parte dele, produto ou sistema fabricado em série ou sob medida, utilizado para aumentar, manter ou melhorar as capacidades funcionais das pessoas com deficiência.

- Serviços: aqueles que auxiliam diretamente uma pessoa com deficiência a selecionar, comprar ou usar os recursos acima definidos.

Os recursos de tecnologia assistiva podem variar de uma simples bengala a um complexo sistema computadorizado. Estão incluídos brinquedos e roupas adaptadas, computadores, *softwares* e *hardwares* especiais, que contemplam questões de acessibilidade, dispositivos para adequação da postura sentada, recursos para mobilidade manual e elétrica, equipamentos de comunicação alternativa, e muitos outros.

Já os serviços são aqueles prestados profissionalmente à pessoa com deficiência visando selecionar, obter ou usar um instrumento de tecnologia assistiva.

Segundo José Tonolli e Rita Bersch (2017), a tecnologia assistiva pode ser classificada em doze categorias, conforme consta abaixo:

1. Auxílios para a vida diária: Materiais e produtos para auxílio em tarefas rotineiras tais como comer, cozinhar, vestir-se, tomar banho e executar necessidades pessoais.
2. CAA - Comunicação Aumentativa e Alternativa: Destinada a atender pessoas sem fala ou escrita funcional ou em defasagem entre sua necessidade comunicativa e sua habilidade em falar e/ou escrever.
3. Recursos de acessibilidade ao computador: Conjunto de *hardware* e *software* especialmente idealizado para tornar o computador acessível a pessoas com privações sensoriais (visuais e auditivas), intelectuais e motoras. Inclui dispositivos de entrada (*mouses*, teclados e acionadores diferenciados) e dispositivos de saída (sons, imagens, informações táteis).

4. Sistemas de controle de ambiente: Através de um controle remoto as pessoas com limitações motoras, podem ligar, desligar e ajustar aparelhos eletroeletrônicos como a luz, o som, televisores, ventiladores, executar a abertura e fechamento de portas e janelas, receber e fazer chamadas telefônicas, acionar sistemas de segurança, entre outros, localizados em seu quarto, sala, escritório, casa e arredores.
5. Projetos arquitetônicos para acessibilidade: Projetos de edificações e urbanismo que garantem acesso, funcionalidade e mobilidade a todas as pessoas, independentemente de sua condição física e sensorial. Como por exemplo rampas, elevadores, adaptações em banheiros, entre outras.
6. Órteses e próteses: Órtese é um apoio ou dispositivo externo aplicado ao corpo para modificar os aspectos funcionais ou estruturais do sistema neuro músculo-esquelético para obtenção de alguma vantagem mecânica ou ortopédica. Já próteses, são peças artificiais que substituem partes ausentes do corpo.
7. Adequação Postural: Um projeto de adequação postural diz respeito à seleção de recursos que garantam posturas alinhadas, estáveis, confortáveis e com boa distribuição do peso corporal.
8. Auxílios de mobilidade: A mobilidade pode ser auxiliada por bengalas, muletas, andadores, carrinhos, cadeiras de rodas manuais ou elétricas, e qualquer outro veículo, equipamento ou estratégia utilizada na melhoria da mobilidade pessoal.
9. Auxílios para cegos ou com visão subnormal: Auxílios para grupos específicos que inclui lupas e lentes, Braille, equipamentos com síntese de voz, *softwares* ampliadores de tela, material gráfico com texturas e relevos, entre outros.

10. Auxílios para pessoas com surdez ou com déficit auditivo: Aparelhos para surdez, celular com mensagens escritas e chamadas por vibração, *software* que favorece a comunicação ao telefone celular transformando em voz o texto digitado no celular e em texto a mensagem falada, livros, textos e dicionários digitais em língua de sinais, sistema de legendas.
11. Mobilidade em veículos: Acessórios que possibilitam uma pessoa com deficiência física dirigir um automóvel, facilitadores de embarque e desembarque como elevadores para cadeiras de rodas (utilizados nos carros particulares ou de transporte coletivo), rampas para cadeiras de rodas, serviços de autoescola para pessoas com deficiência.
12. Esporte e Lazer: Recursos que favorecem a prática de esporte e participação em atividades de lazer.

Podemos considerar que o presente trabalho se enquadra em algumas dessas categorias listadas acima. Como por exemplo: a categoria 1, por ser um auxílio na vida diária do deficiente, o ajudando a cozinhar; na categoria 4, por ser um sistema de controle de ambiente, já que o aplicativo funcionará como uma espécie de controle remoto para o micro-ondas; e na categoria 9, por ser direcionado para o público com deficiência visual especificamente.

2.3 Android

Android é um sistema operacional desenvolvido pela Android Inc., que posteriormente foi adquirida pelo Google. Atualmente, os dispositivos Android incluem *smartphones*, *tablets*, *e-readers*, robôs, motores à jato, satélites da NASA, equipamentos com sistemas inteligentes, e muitos outros.

Os aplicativos Android são desenvolvidos em Java, uma das linguagens de programação mais usadas no mundo. Uma das maiores vantagens do sistema operacional Android é que ele apresenta o código-fonte aberto e gratuito, o que estimula a rápida inovação da plataforma. (DEITEL, WALD *et al.*, 2016) [12]

2.3.1 Acessibilidade no Android

Um dos desafios enfrentados pelos deficientes visuais com os *smartphones* Android são as telas *touchscreen* (telas sensíveis ao toque). Para vencer esse desafio e permitir o uso do *smartphone* por esse público, o sistema Android possui um recurso de acessibilidade que já vem instalado nos *smartphones* atuais. Esse recurso utiliza um serviço TTS (*Text-To-Speech* - conversor de texto para fala) para fazer a leitura dos dados da tela do *smartphone*. Esse recurso é chamado de Talkback.

O TalkBack é o leitor de tela do Google incluído em dispositivos Android. Ele oferece *feedback* falado para que o usuário possa usar seu dispositivo sem olhar para a tela. (ANDROID ACCESSIBILITY, 2018) [13]

Este aplicativo de ajuda parece cumprir o objetivo para muitos usuários, mas pode não funcionar tão bem para os outros. O Google TalkBack tem muitos recursos úteis para leitura de texto da tela, mas a facilidade com que ele funciona pode depender, em parte, do dispositivo e da versão do sistema Android. Segundo uma *review* feita pelo *site* Common Sense Media, em um Samsung Galaxy Tab executando o Android 7.1.1, havia muitos atrasos e alguns casos em que o texto não era lido automaticamente. Além disso, os padrões de voz e fala *text-to-speech* às vezes não combinavam completamente com o texto, especialmente quando havia ênfase com letras maiúsculas. No entanto, esta ferramenta é gratuita e é um ótimo complemento para os usuários do Android que possuem alguma deficiência visual e necessitam de conversão de texto para fala. (COMMON SENSE MEDIA, 2018)¹ [14]

A vantagem do sistema operacional Android de possuir o código-fonte aberto e gratuito permite que diversas empresas usem o sistema como base para seus produtos. Porém, essa vantagem pode se transformar em desvantagem no quesito tecnologia assistiva. Se os fabricantes julgarem a área de acessibilidade do sistema desnecessária e não considerarem deficientes visuais como possíveis clientes, e assim, mudarem partes do sistema, a tecnologia assistiva pode simplesmente não funcionar ou funcionar de forma limitada para esses aparelhos.

Além do Talkback, podem ser encontrados no Google Play² alguns aplicativos que têm como objetivo gerar acessibilidade para os deficientes visuais e facilitar o desempenho de alguma tarefa. Alguns desses aplicativos serão apresentados no próximo capítulo.

¹Review do App Talkback Traduzida. Fonte: <https://www.commonsemmedia.org/app-reviews/google-talkback>

²Loja online Google Play. Disponível em: <https://play.google.com/store>

3 TECNOLOGIA MÓVEL DISPONÍVEL PARA DEFICIENTES VISUAIS

Uma sociedade igualitária deve ter como princípio a participação efetiva de todos os cidadãos. Por isso, é imprescindível a construção de políticas de inclusão para suprir as diferenças sociais. E a tecnologia tem sido uma grande aliada dos deficientes, ajudando a quebrar obstáculos.

Segundo Campbell, desde a invenção do Código de Braille, nada teve tanto impacto nos programas de educação, reabilitação e emprego quanto o recente desenvolvimento da informática para os cegos. (CAMPBELL, 2001)

A tecnologia não soluciona todos os problemas enfrentados pelos deficientes visuais, mas pode ser usada como uma incrível ferramenta de apoio, proporcionando uma maior independência para esse público. Um exemplo dessas tecnologias são os *smartphones* e seus aplicativos de acessibilidade.

No quesito acessibilidade aos deficientes visuais, o sistema operacional iOS, presente nos *smartphones* da Apple, revolucionou com a criação do *VoiceOver*: o primeiro leitor de tela totalmente compatível com os monitores em Braille. Já o leitor de tela TalkBack do Android, já chega instalado nos dispositivos, mas possui algumas funções a menos do que o assistente da Apple. De acordo com José Francisco de Souza, formado em Letras e professor de Informática no Instituto Benjamin Constant¹ (uma tradicional instituição de ensino para deficientes visuais localizada no bairro da Urca, na cidade e estado do Rio de Janeiro, no Brasil), a Apple é mais eficiente quando se trata de acessibilidade, pois o *VoiceOver* lê todos os eventos da tela, estando disponível para iPhone, iPod, iPad e MacBooks. E para quem tem baixa visão, os recursos do Android não são tão eficientes, mas, dependendo do grau de patologia, ele pode ser usado por essas pessoas. (Techo) [18]

O Android ainda está um pouco atrasado em relação à Apple na vertente de acessibilidade, porém já existem alguns aplicativos disponíveis acessíveis aos deficientes visuais, e estes serão listados na sequência:

¹ Instituto Benjamin Constant é uma tradicional instituição de ensino para deficientes visuais localizada no bairro da Urca, na cidade e estado do Rio de Janeiro, no Brasil.

- **BlindTool**

O aplicativo tem como objetivo reconhecer objetos. Funciona da seguinte maneira: o usuário deve apontar o celular para seu entorno até senti-lo vibrar. Isso significa que o aplicativo detectou um objeto reconhecível e pode verbalizar qual é. A desvantagem do aplicativo é que, além de estar disponível apenas na língua inglesa, o sistema ainda está na sua versão beta e está sujeito a erros de funcionamento.

Figura 3: Interface do aplicativo BlindTool



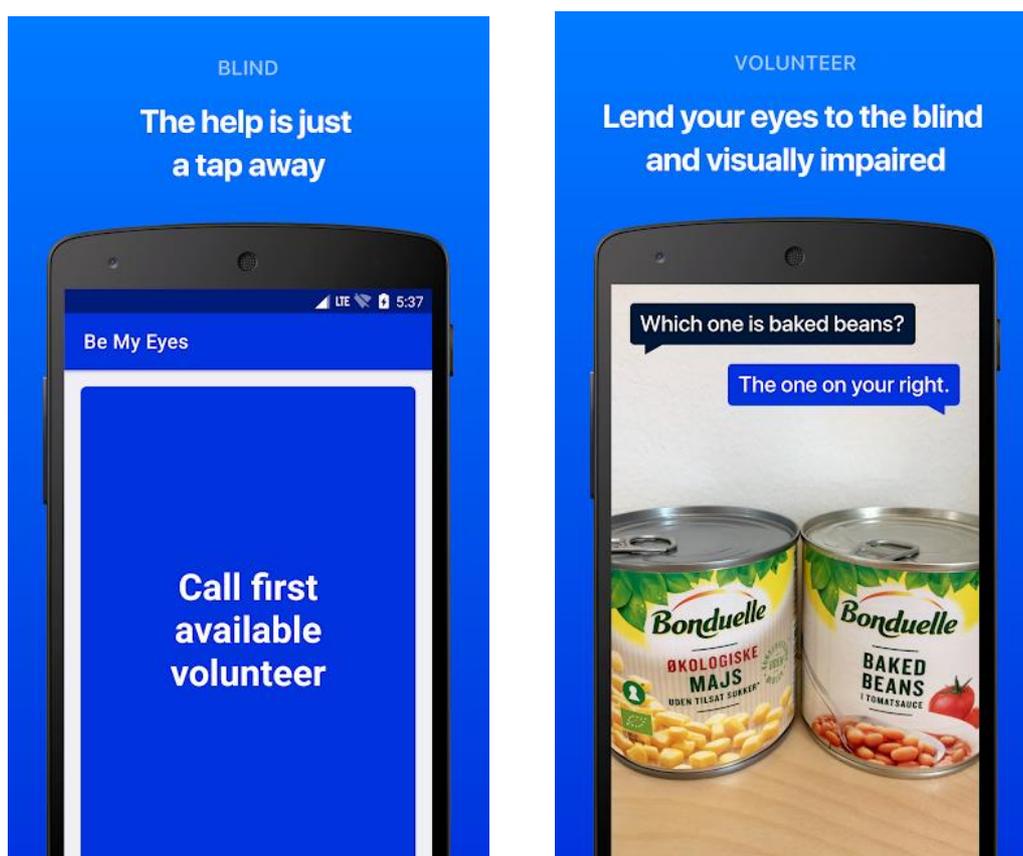
Fonte: Google Play

- **Be My Eyes**

É um aplicativo colaborativo que funciona da seguinte forma: usuários cegos ou com deficiência visual podem solicitar ajuda de um voluntário com visão,

que será notificado no telefone. Assim que o primeiro voluntário aceitar a solicitação de ajuda, uma conexão de áudio e vídeo ao vivo será configurada entre as duas partes. Através da conexão de vídeo da câmera traseira do usuário cego, o voluntário pode auxiliar o usuário na sua tarefa - desde saber o prazo de validade no leite, até se certificar de que suas roupas combinam.

Figura 4: Interface do aplicativo Be My Eyes



Fonte: Google Play

- **CPqD Alcance**

Este aplicativo possui um modelo de interação inclusivo com narração automática das telas em Português, que permite ao deficiente visual utilizar diferen-

tes funções básicas como realizar e receber ligações telefônicas, enviar e receber mensagens de texto (SMS), consultar data e hora, e até funções mais avançadas, tais como despertador com lembrete de voz e acesso à câmera fotográfica e galeria de fotos de seu *smartphone touchscreen* com autonomia, privacidade e simplicidade, melhorando a sua qualidade de vida e se apropriando das facilidades de um mundo cada vez mais digital e conectado. Após instalado, o aplicativo se torna a interface padrão do celular, como é ilustrado na figura 5.

Figura 5: Interface do aplicativo CPqD Alcance



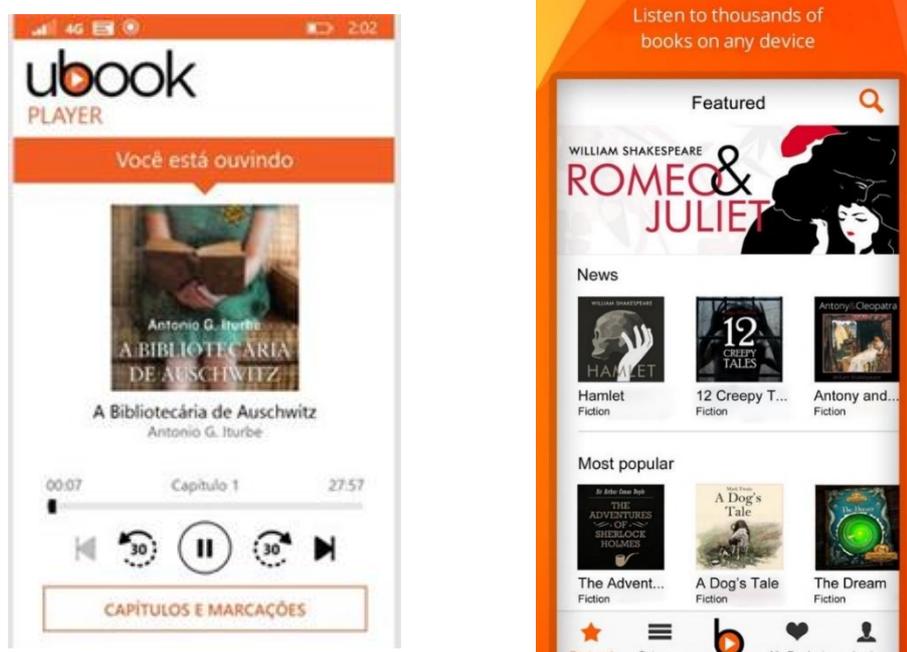
Fonte: Google Play

- **Ubook**

O Ubook é um software de audioteca e conta com um enorme acervo de livros, contando com mais de dez mil títulos cadastrados em seu catálogo. O sistema está disponível para dispositivos Android e IOS, mas também pode ser utilizado diretamente pelo seu *website*. Apesar de ser uma alternativa interessante

para deficientes visuais, o sistema não foi desenvolvido diretamente para esse público. Como consequência, não apresenta suporte de conversão texto para fala no catálogo de livros, o que pode dificultar para o deficiente a tarefa de escolher o texto a ser lido, fazendo-se necessário o uso em conjunto com um leitor de telas.

Figura 6: Interface do aplicativo Ubook



Fonte: Google Play

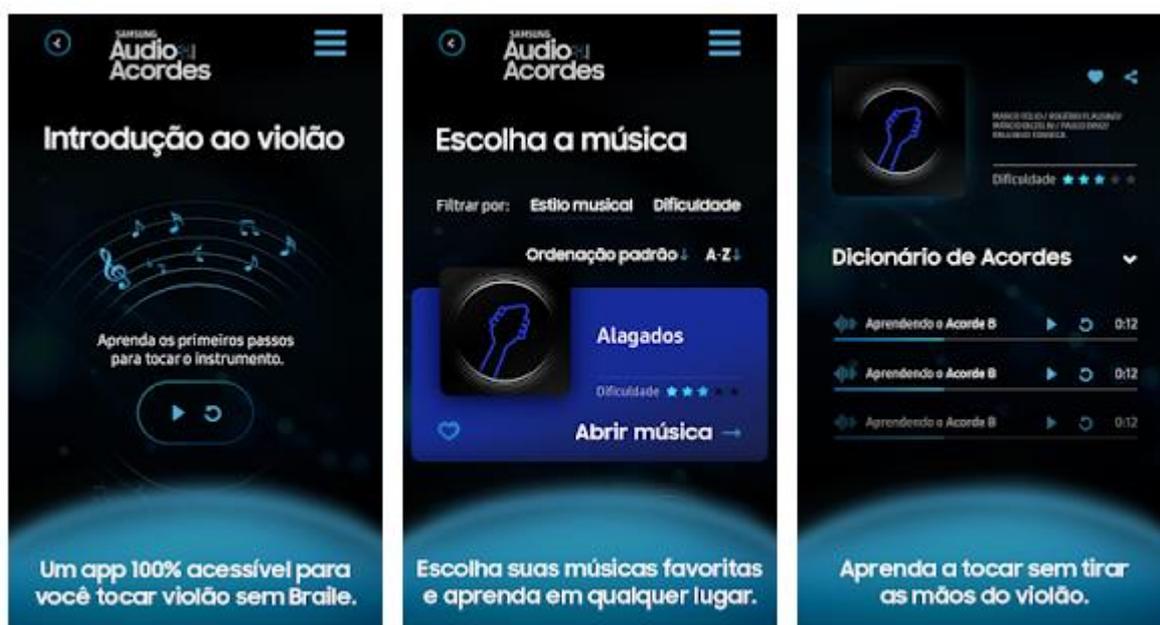
- **Áudio Acordes**

O projeto Áudio Acordes consiste no primeiro sistema de voz que permite que pessoas com deficiência visual aprendam a tocar violão, sem a necessidade do Braille. Segundo a empresa Samsung, a música é uma ferramenta poderosa para a socialização e para a melhoria da autoestima de pessoas com alguma deficiência visual¹. No entanto, apenas 10% das pessoas cegas no Brasil conhecem o Braille, criando assim uma barreira para eles na hora de aprenderem instrumentos musicais

e se conectarem com o mundo ao seu redor. O *site* e o aplicativo **Áudio Acordes** usam uma metodologia que a Samsung desenvolveu para ensinar pessoas com deficiências visuais a tocarem violão usando cifras audíveis, exigindo apenas um computador ou um dispositivo móvel com conexão à internet para funcionar.

O sistema está disponível para celulares Samsung a partir do sistema Android 4.4, em forma de aplicativo, e no formato de *site*, com algumas funções a menos.

Figura 7: Interface do aplicativo **Áudio Acordes**



Fonte: Samsung Brasil

4 TECNOLOGIAS UTILIZADAS

Esse capítulo descreve todas as tecnologias utilizadas como suporte na aplicação. Como por exemplo, a linguagem Java, o ambiente de desenvolvimento Android Studio, Arduíno IDE, Arduíno, *Bluetooth*.

4.1 Circuito no micro-ondas

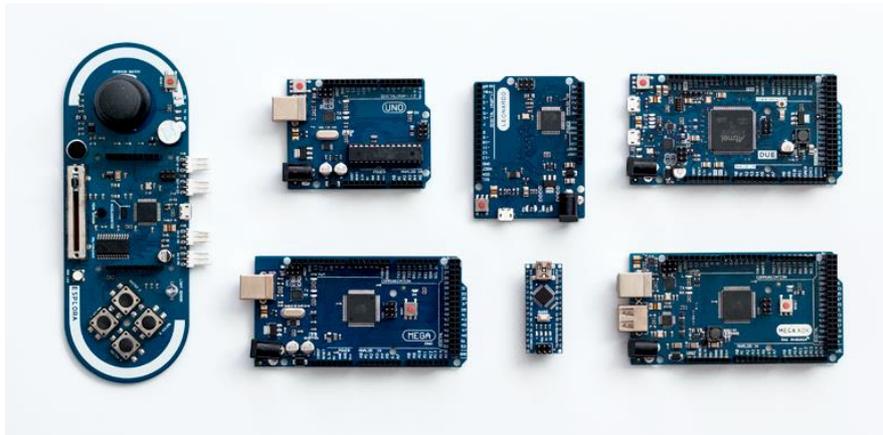
Nesta seção serão apresentadas as tecnologias utilizadas no desenvolvimento do circuito no micro-ondas.

4.1.1 Arduíno

O Arduino foi criado em 2005 por um grupo de 5 pesquisadores: Massimo Banzi, David Cuartielles, Tom Igoe, Gianluca Martino e David Mellis. O objetivo era elaborar um dispositivo que fosse ao mesmo tempo barato, funcional e fácil de programar, acessível aos estudantes e projetistas amadores. Além disso, foi adotado o conceito de *hardware* livre, o que significa que qualquer um pode montar, modificar, melhorar e personalizar o Arduino, partindo do mesmo *hardware* básico.

Uma placa Arduíno é composta, basicamente, por um controlador Atmel AVR de 8 *bits*, uma interface serial ou USB e alguns pinos digitais e analógicos. A parte de *software* é desenvolvida por meio de linguagem C/C++ e pode ser programada via Arduíno - IDE (*Integrated Development Environment*, ou Ambiente de Desenvolvimento Integrado). [15]

Figura 8: Modelos de placas de Arduino



Fonte: Google Inc.

4.1.2 Módulo Bluetooth

O módulo Bluetooth HC-06 é usado para comunicação *wireless* (sem fio) entre o Arduino e algum outro dispositivo com *bluetooth*, como por exemplo um telefone celular, um computador ou *tablet*. As informações recebidas pelo módulo são repassadas ao Arduino (ou outro microcontrolador) via comunicação serial. Ele dispõe de apenas 4 pinos de entrada e saída de dados, sendo dois deles para alimentação do sistema, e os outros dois para transmissão e recepção de dados via serial.

O alcance do módulo segue o padrão da comunicação *bluetooth*, que é de aproximadamente 10 metros. Esse módulo funciona apenas em modo *slave* (escravo), ou seja, ele permite que outros dispositivos se conectem a ele, mas não permite que ele próprio se conecte a outros dispositivos *bluetooth*. [16]

Figura 9: Módulo Bluetooth HC-06

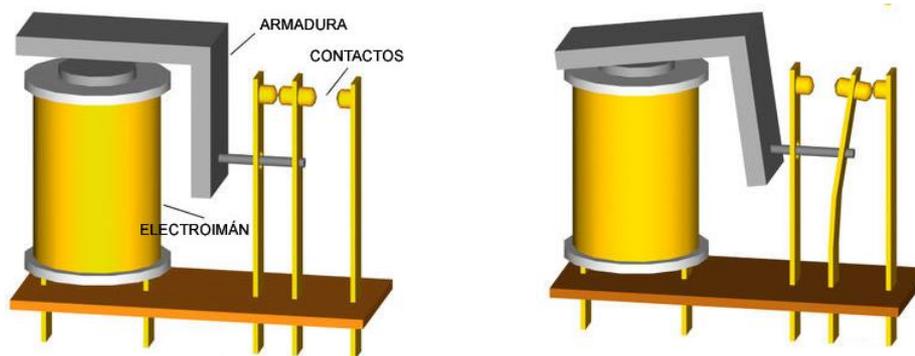


Fonte: Buildbot (2015)

4.1.3 Módulo Relé

Um relé é um interruptor eletromecânico. A essência do funcionamento é a seguinte: quando a bobina, que é um eletroímã, é alimentada, ela cria um campo magnético que atrai uma peça metálica. Ao atrair a peça metálica, dois contatos que antes não se tocavam, agora fazem contato. E um outro contato, que antes encostava, agora fica separado. Portanto, as chaves são comutadas. A ideia de funcionamento pode ser vista graficamente na figura 10. [17]

Figura 10: Funcionamento do Relé



Fonte: Wikipedia (2015)

Para este projeto, foi usado um módulo relé de 8 canais, mais quatro relés independentes para fazer os vários acionamentos do *display* do micro-ondas.

Figura 11: Módulo Relé de 8 canais

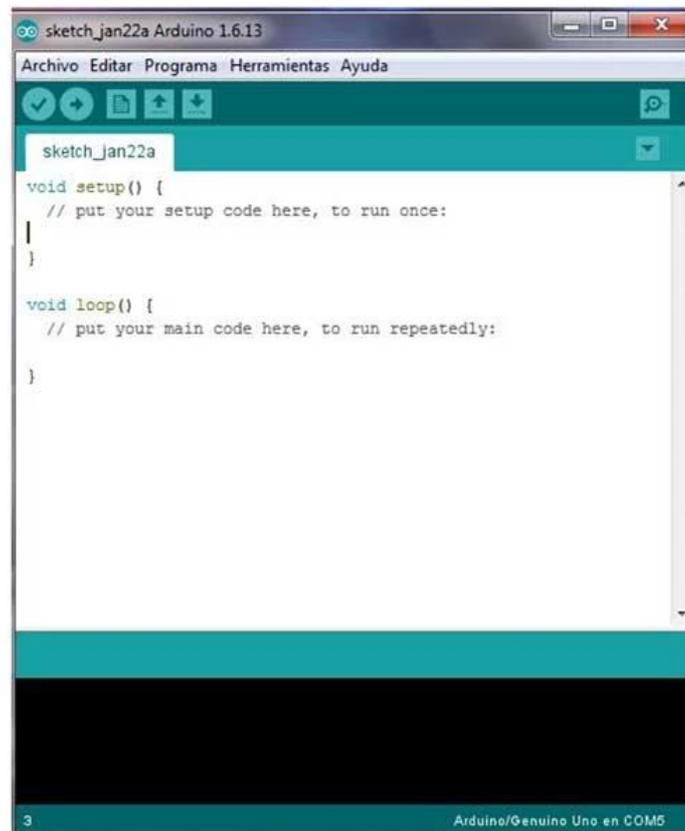


Fonte: Google Inc.

4.1.4 Arduíno IDE

O ambiente de desenvolvimento integrado (IDE) do Arduino é um aplicativo de plataforma cruzada (para Windows, mac OS, Linux) que é escrito na linguagem de programação Java. Ele é usado para escrever e carregar programas em placas compatíveis com Arduino. Os programas são escritos, normalmente, nas linguagens C ou C++.

Figura 12: Tela do software Arduino IDE



Fonte: Arduino IDE

4.2 Aplicativo

Nesta seção serão apresentadas as tecnologias utilizadas no desenvolvimento do aplicativo.

4.2.1 Java

A linguagem Java foi desenvolvida pela empresa Sun Microsystems, na década de 90. A ideia principal do Java era que os aparelhos eletrônicos se comunicassem entre si. E por isso, teria a necessidade de a linguagem não ficar presa aos

sistemas operacionais e nem da plataforma que estava sendo utilizada. Assim, surgiu uma das principais características da linguagem: independência de plataforma. A independência de plataforma possibilita o programa ser executado em diferentes plataformas e sistemas operacionais, através de um emulador conhecido como a Máquina Virtual Java ou JVM (*Java Virtual Machine*), que ajuda rodar os sistemas baseados em Java.

Outra importante característica do Java é utilizar o paradigma de orientação a objetos, no qual seus elementos são tratados como objetos que são compostos por atributos e métodos definidos a partir de classes. Em suma, uma classe é uma estrutura que abstrai um conjunto de objetos com características similares. Ela define o comportamento de seus objetos - através de métodos - e os estados possíveis destes objetos - através de atributos.

4.2.2 Android Studio

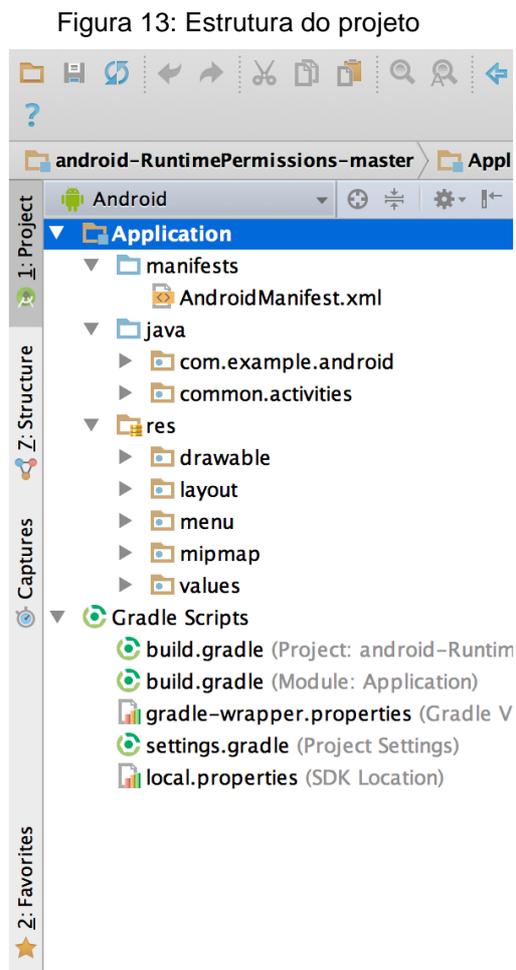
Android Studio é um ambiente de desenvolvimento integrado (IDE) para desenvolver para a plataforma Android. Foi lançado em 2013 na conferência Google I/O, e a partir desta data, foi adotado como plataforma oficial da Google para desenvolver aplicações Android. O Android Studio oferece diversos recursos para aumentar a produtividade na criação de aplicativos Android, como por exemplo:

- **Um sistema de compilação flexível baseado no Gradle:** De forma simplificada, podemos dizer que Gradle é um sistema para automatizar o processo de compilação da aplicação, que, por padrão, estrutura todo projeto no Android Studio para utilizá-lo. O Gradle automaticamente pega todos os arquivos fonte (.java ou .xml), aplica as ferramentas apropriadas e agrupa todos eles em um único arquivo comprimido, o APK. Utilizar o Gradle internamente ao projeto simplifica o uso de bibliotecas externas o que possibilita mais agilidade no processo de importação de ferramentas.
- **Um emulador rápido com inúmeros recursos** que simula um dispositivo e exibe-o no computador de desenvolvimento. Com ele, é possí-

vel criar protótipos, desenvolver e testar aplicativos do Android sem usar um dispositivo de hardware.

- **Integração com GitHub** para ajudar a criar recursos comuns dos aplicativos e importar exemplos de código. O que possibilita um sistema de controle de versão para acompanhar as mudanças feitas no código base. Além de registrar quem efetuou a mudança e permitir a restauração do código removido ou modificado.

A estrutura de um projeto no Android Studio pode ser vista na ilustração da figura 13.



Fonte: Android Studio

Cada módulo de aplicativo contém as pastas: manifests, java e res.

- **manifestos:** contém o arquivo AndroidManifest.xml. O arquivo de manifesto apresenta informações essenciais sobre o aplicativo ao sistema Android, necessárias para o sistema antes que ele possa executar o código do aplicativo. Como por exemplo: nomear o pacote Java do aplicativo que serve como identificador único exclusivo para o aplicativo, declarar as permissões que o aplicativo deve ter para acessar partes protegidas da API e interagir com outros aplicativos, entre outras;
- **java:** contém os arquivos de código-fonte do projeto, incluindo o código de teste;
- **recursos:** contém todos os recursos que não são código, como configurações de elementos visuais, layouts XML, imagens, sons, etc.

5 MONTAGEM DO CIRCUITO NO MICRO-ONDAS

O micro-ondas utilizado para o projeto é da marca Philco, modelo PME25, como pode ser visto na figura 14.

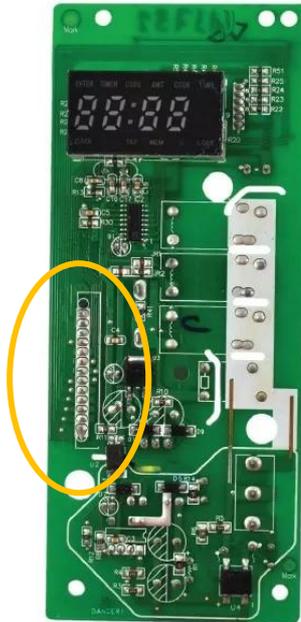
Figura 14: Micro-ondas Philco PME25



Fonte: Google Inc.

A placa de controle do micro-ondas foi adaptada para receber os comandos do *smartphone* via *bluetooth*. Os pinos em destaque na imagem, são responsáveis por controlar o display. Cada tecla do teclado do micro-ondas é uma combinação de dois pinos. Para este projeto, foram mapeadas apenas as combinações das teclas essenciais para o funcionamento do eletrodoméstico: o teclado numérico e as teclas ligar, pausar e cancelar.

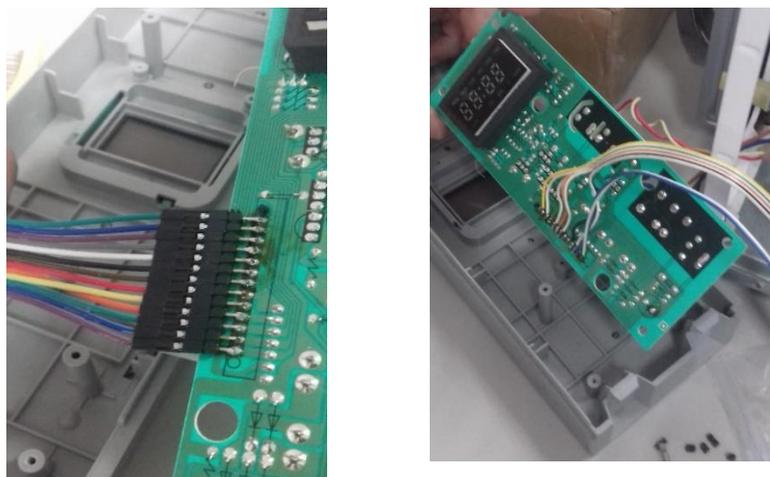
Figura 15: Placa de Controle do micro-ondas



Fonte: Google Inc.

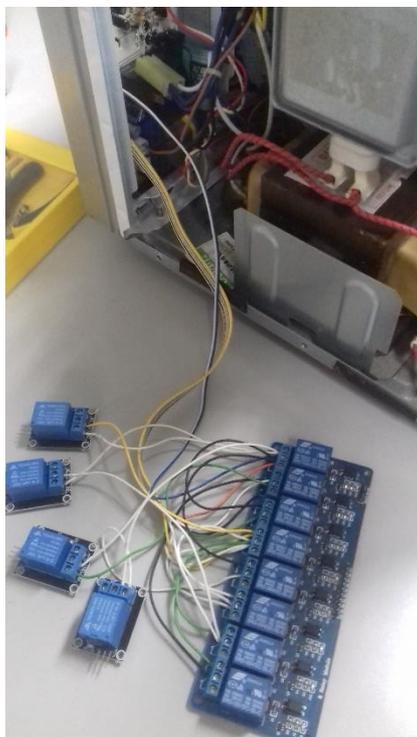
Primeiramente, foram soldados na placa os fios nos pinos controladores do display. Cada combinação de pinos foi conectada a um relé. O relé funciona como um interruptor liga/desliga, quando ativo, é como se a tecla tivesse sido pressionada. O controle dos relés é feito pelo Arduino, que executa os comandos de acordo com os dados recebidos pelo módulo *bluetooth*. As ilustrações dos circuitos podem ser vistas nas figuras 17, 18, 19 e 20.

Figura 16: Fios soldados na placa de controle



Fonte: Autor do texto

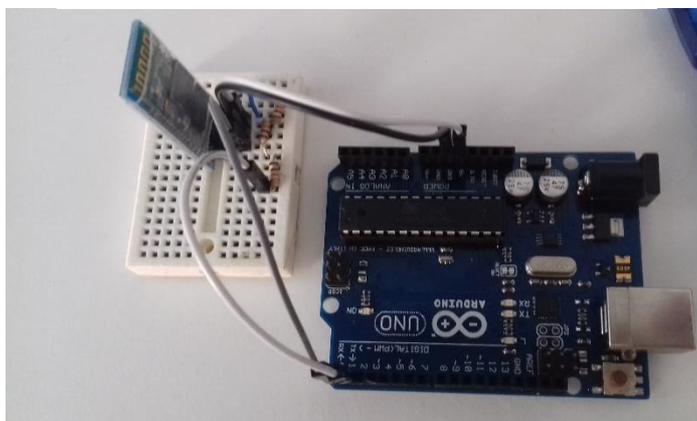
Figura 17: Pinos da placa de controle conectados com os relés



Fonte: Autor do texto

O circuito Arduino + Módulo Bluetooth pode ser visto a seguir.

Figura 18: Circuito Arduino + Módulo Bluetooth

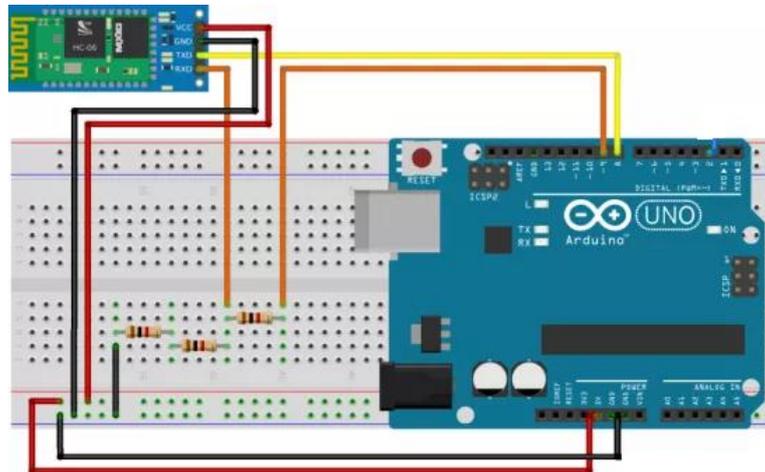


Fonte: Autor do texto

Como o módulo *bluetooth* HC-06 utilizado neste projeto trabalha com um nível de sinal de 3,3V para a comunicação serial e o Arduino produz uma saída de 5V,

foi necessário um divisor de tensão no pino RX do módulo *bluetooth* para não danificar o aparelho. O divisor de tensão foi feito com três resistores de 1k ohm, gerando a saída de 3,3V no pino RX do módulo. A figura 19 ilustra o divisor de tensão feito entre o Arduíno e o módulo Bluetooth.

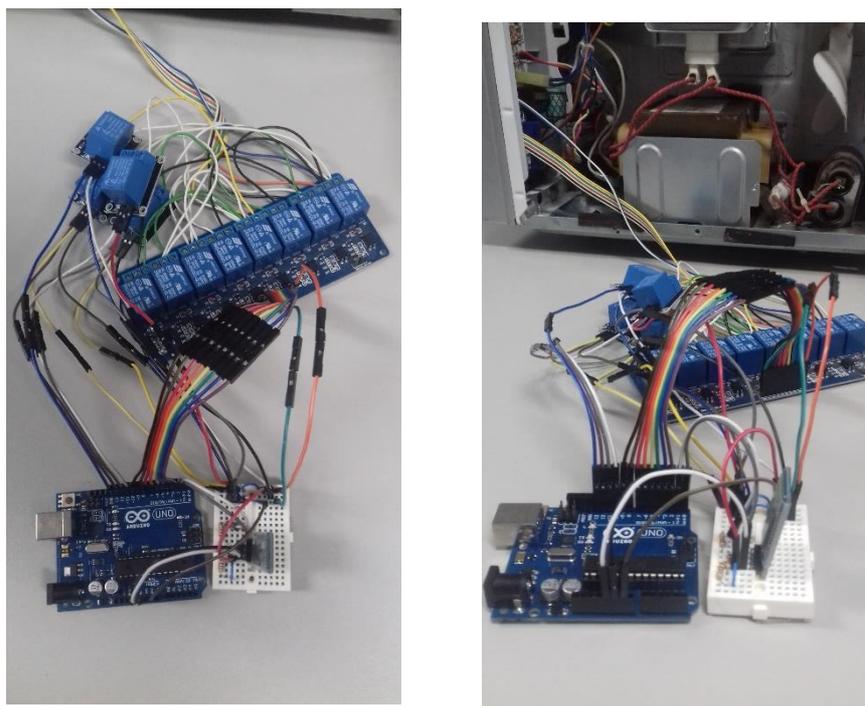
Figura 19: Divisor de tensão



Fonte: Google Inc.

A saída dos relés foram conectadas às saídas digitais do Arduíno, a fim de receberem os comandos de acordo com o que foi programado.

Figura 20: Circuito completo



Fonte: Autor do texto

5.1 Código do Arduino

O código do Arduino pode ser visto nas figuras 21, 22 e 23, respectivamente. Cada constante definida representa um relé e a saída correspondente no Arduino. Por exemplo, o relé D0 está conectado à saída número 2 do Arduino, e assim sucessivamente.

Figura 21: Constantes do programa

```
#define D0 2
#define D1 3
#define D2 4
#define D3 5
#define D4 6
#define D5 7
#define D6 8
#define D7 9
#define D8 10
#define D9 11
#define LIGAR 12
#define CANCELAR 13
#define DELAY_TECLA 500
```

Fonte: Autor do texto

O Setup é uma seção obrigatória de um programa para Arduino. Qualquer código que estiver dentro do setup é executado uma única vez no início do programa. É onde ficam as configurações iniciais.

Neste código (figura 22), todos os pinos do Arduino conectados com os relés foram definidos como saída (OUTPUT). Todos os relés iniciam desativados (LOW). Como o módulo Relé funciona com lógica invertida, os oito relés do módulo foram

iniciados com nível alto (HIGH). A melhor opção seria utilizar relés que funcionassem na mesma lógica ou até mesmo alguns equipamentos menores, porém, para esse projeto, foram utilizados os componentes disponíveis na Universidade.

Figura 22: Método setup

```
void setup() {
  // put your setup code here, to run once:
  Serial.begin(115200);
  pinMode(D0, OUTPUT);
  pinMode(D1, OUTPUT);
  pinMode(D2, OUTPUT);
  pinMode(D3, OUTPUT);
  pinMode(D4, OUTPUT);
  pinMode(D5, OUTPUT);
  pinMode(D6, OUTPUT);
  pinMode(D7, OUTPUT);
  pinMode(D8, OUTPUT);
  pinMode(D9, OUTPUT);
  pinMode(LIGAR, OUTPUT);
  pinMode(CANCELAR, OUTPUT);

  digitalWrite(D0, HIGH);
  digitalWrite(D1, HIGH);
  digitalWrite(D2, HIGH);
  digitalWrite(D3, HIGH);
  digitalWrite(D4, HIGH);
  digitalWrite(D5, HIGH);
  digitalWrite(D6, HIGH);
  digitalWrite(D7, HIGH);
  digitalWrite(D8, LOW);
  digitalWrite(D9, LOW);
  digitalWrite(LIGAR, LOW);
  digitalWrite(CANCELAR, LOW);
}
```

Fonte: Autor do texto

Assim como a seção Setup, o Loop também é obrigatório em um programa para Arduino. A função loop () será executada para sempre, ou até que seja feito *upload* de um novo código, reiniciando o processo. Ela também pode ser reiniciada resetando o Arduino (através do botão de *reset*, por exemplo).

O programa verifica se existe algum dispositivo *bluetooth* conectado. Enquanto houver conexão, o Arduino ficará recebendo caracteres, via *bluetooth*, do aplicativo no *smartphone*. Cada caractere representa uma tecla do micro-ondas (teclado

numérico, ligar/pausar e cancelar) e irá acionar o relé correspondente, que como consequência, irá acionar o *display*.

Figura 23: Método loop

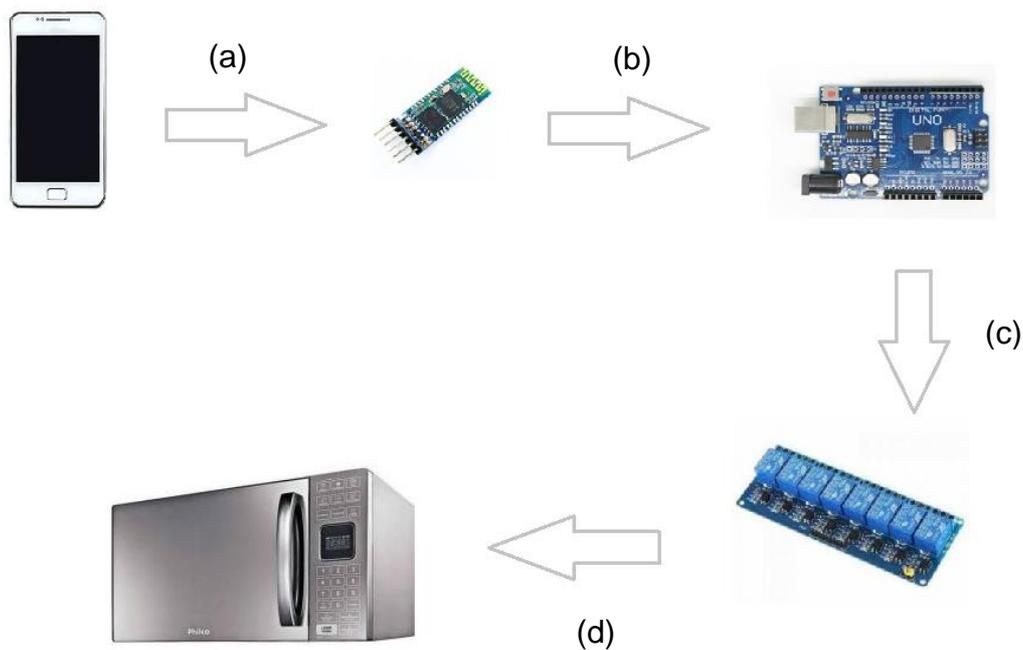
```

void loop() {
  // put your main code here, to run repeatedly:
  if (Serial.available() > 0)
  {
    char data = Serial.read();

    switch (data)
    {
      case '0':
        digitalWrite(D0, LOW);
        delay(DELAY_TECLA);
        digitalWrite(D0, HIGH);
        break;
      case '1':
        digitalWrite(D1, LOW);
        delay(DELAY_TECLA);
        digitalWrite(D1, HIGH);
        break;
      case '2':
        digitalWrite(D2, LOW);
        delay(DELAY_TECLA);
        digitalWrite(D2, HIGH);
        break;
      case '3':
        digitalWrite(D3, LOW);
        delay(DELAY_TECLA);
        digitalWrite(D3, HIGH);
        break;
      case '4':
        digitalWrite(D4, LOW);
        delay(DELAY_TECLA);
        digitalWrite(D4, HIGH);
        break;
      case '5':
        digitalWrite(D5, LOW);
        delay(DELAY_TECLA);
        digitalWrite(D5, HIGH);
        break;
      case '6':
        digitalWrite(D6, LOW);
        delay(DELAY_TECLA);
        digitalWrite(D6, HIGH);
        break;
      case '7':
        digitalWrite(D7, LOW);
        delay(DELAY_TECLA);
        digitalWrite(D7, HIGH);
        break;
      case '8':
        digitalWrite(D8, HIGH);
        delay(DELAY_TECLA);
        digitalWrite(D8, LOW);
        break;
      case '9':
        digitalWrite(D9, HIGH);
        delay(DELAY_TECLA);
        digitalWrite(D9, LOW);
        break;
      case 'L':
        digitalWrite(LIGAR, HIGH);
        delay(DELAY_TECLA);
        digitalWrite(LIGAR, LOW);
        break;
      case 'C':
        digitalWrite(CANCELAR, HIGH);
        delay(DELAY_TECLA);
        digitalWrite(CANCELAR, LOW);
        break;
    }
  }
}

```

Fonte: Autor do texto



Passo a passo do funcionamento:

- a. O *smartphone* se conecta com o módulo *bluetooth* e o aplicativo irá enviar caracteres como comandos. Por exemplo, no caso do teclado numérico, ao selecionar a tecla '1', o comando enviado via *bluetooth* é o caractere '1', este será analisado pelo Arduíno para ativar corretamente a tecla do número 1 no micro-ondas. O mesmo acontece no comando de voz. Os números identificados na fala são enviados como caracteres via *bluetooth* para o Arduíno.
- b. O módulo *bluetooth*, que está conectado ao Arduíno, irá receber os comandos do aplicativo e estes serão identificados pelo Arduíno, através do código programado no mesmo. Ao identificar

o comando, o Arduíno envia um sinal de ativação ao relé conectado na tecla correspondente.

- c. O relé, funcionando como um interruptor, após receber o comando do Arduíno, ativará a tecla correspondente no micro-ondas.
- d. As teclas serão ativadas, fazendo com que o tempo desejado apareça no *display* e o micro-ondas execute seu funcionamento.

6 DESENVOLVIMENTO DO APLICATIVO E TESTES

Este capítulo define todas as etapas do desenvolvimento e os testes realizados.

6.1 Android Manifest

O aplicativo necessita de algumas permissões de acesso para o seu funcionamento. Essas permissões são definidas no arquivo `AndroidManifest.xml`. Este arquivo inclui as definições de nome do aplicativo, ícone utilizado e declarações das *activities*. Neste projeto foram necessárias duas permissões de *bluetooth*, como consta na figura 24.

Figura 24: Permissões do aplicativo

```
<uses-permission android:name="android.permission.BLUETOOTH"/>  
<uses-permission android:name="android.permission.BLUETOOTH_ADMIN"/>
```

Fonte: Autor do texto

- `Android.permission.BLUETOOTH`: permite que a aplicação se conecte com dispositivos *bluetooth* pareados com o *smartphone*.
- `Android.permission.BLUETOOTH_ADMIN`: permite que a aplicação possa descobrir e parear com outros dispositivos *bluetooth*.

6.2 Classes

O código desta aplicação está dividido em classes e cada classe representa uma *Activity* do aplicativo. *Activity* é um componente de aplicativo que fornece uma tela com a qual os usuários podem interagir para fazer algo, como discar um número no telefone, tirar uma foto, enviar um e-mail ou ver um mapa. Nesta seção serão apresentadas as classes do aplicativo e suas principais funcionalidades.

- **MainActivity**: esta classe representa a tela inicial da aplicação. Nesta *Activity* são declarados dois botões que dirigem o usuário para outras áreas do aplicativo: O botão de teclado (direciona o usuário para a área que permite enviar comandos ao micro-ondas através de um teclado virtual); e o botão de voz (direciona o usuário para a área que permite enviar comandos de voz ao micro-ondas).

Cada *Activity* gerencia a interface com o usuário. Ela quem recebe as requisições, as trata e processa. Ao iniciar uma *Activity*, o método `onCreate()` é executado. Este método é responsável, geralmente, por carregar os *layouts* XML (elementos da estrutura visual da interface do usuário) e outras operações de inicialização.

Primeiramente, são carregados o *layout* dos botões nas variáveis `mTecladoButton` e `mVozButton`, respectivamente. Em seguida, são implementados os métodos a serem executados ao clicar nos botões. Os dois botões levam o usuário, antes de tudo, para a *Activity* que realiza a conexão do *smartphone* com o módulo *bluetooth* presente no circuito do micro-ondas: *DeviceListActivity*. Para iniciar esta nova *Activity*, foi necessário o uso da classe *Intent*. Essa classe representa a 'intenção de fazer algo' no aplicativo. Neste caso, a intenção de iniciar uma outra *Activity*, ou seja, ir para outra tela do aplicativo. Uma *intent* descreve a atividade a iniciar e carrega os dados necessários.

As partes mais importantes dessa classe são ilustradas na figura 25.

Figura 25: Classe MainActivity

```
mTecladoButton = findViewById(R.id.button_teclado);
mVozButton = findViewById(R.id.button_voz);

mTecladoButton.setOnClickListener((v) -> {
    Intent i = new Intent( packageContext: MainActivity.this, DeviceListActivity.class);
    startActivity(i);
});

mVozButton.setOnClickListener((v) -> {
    Intent myIntent = new Intent( packageContext: MainActivity.this, DeviceListActivity.class);
    myIntent.putExtra(EXTRA_ACTIVITY, value: "VozClass");
    startActivity(myIntent);
});
```

Fonte: Autor do texto

- **DeviceListActivity**: esta classe é responsável por apresentar ao usuário, uma lista de dispositivos *bluetooth* disponíveis para a conexão, bem como passar para as outras Activities o nome e o endereço *bluetooth* do dispositivo escolhido, para que a conexão seja realizada. Além disso, a classe possui um método responsável por verificar se o *bluetooth* do *smartphone* está ativo. Caso não esteja, o método solicita ao usuário a ativação do mesmo.

Figura 26: Método que verifica se o bluetooth está ligado - checkBTState()

```
//método que verifica se o bluetooth está ligado
private void checkBTState()
{
    // Verifica se o bluetooth está ligado
    mBluetoothAdapter=BluetoothAdapter.getDefaultAdapter();
    if(mBluetoothAdapter==null) {
        Toast.makeText(getApplicationContext(), text: "Device does not support Bluetooth", Toast.LENGTH_SHORT).show();
        finish();
    } else {
        if (!mBluetoothAdapter.isEnabled()) {
            //Solicita a ativação do bluetooth
            Intent enableBluetoothIntent = new Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);
            startActivityForResult(enableBluetoothIntent, requestCode: 1);
        }
    }
}
}
```

Fonte: Autor do texto

Figura 27: Método responsável por repassar o endereço bluetooth para as outras Activities

```
private OnItemClickListener mDeviceClickListener = (av, v, arg2, arg3) → {
    // Pega o endereço MAC do dispositivo (17 últimos chars da view)
    String info = ((TextView) v).getText().toString();
    String address = info.substring(info.length() - 17);

    // Inicia a Activity correspondente, passando o endereço do bluetooth
    String value = getIntent().getStringExtra(EXTRA_ACTIVITY);
    Intent i;

    if(value != null && value.compareTo("VozClass") == 0)
    {
        i = new Intent( packageContext: DeviceListActivity.this, VozClass.class);
        i.putExtra(EXTRA_DEVICE_ADDRESS, address);
        startActivity(i);
    }
    else
    {
        i = new Intent( packageContext: DeviceListActivity.this, TecladoClass.class);
        i.putExtra(EXTRA_DEVICE_ADDRESS, address);
        startActivity(i);
    }
};
```

Fonte: Autor do texto

- **TecladoClass:** esta classe representa a parte do teclado virtual do aplicativo. É nela que os botões do teclado são associados ao *layout* do teclado, através do comando *findViewById*, e, após isso, são implementadas as funções de envio dos comandos via *bluetooth* ao clicar na tecla desejada.

Como já foi dito, neste projeto foram mapeadas apenas as teclas essenciais para o funcionamento do micro-ondas: o teclado numérico, ligar, cancelar e pausar.

Figura 28: Teclado mapeado

```
mUmButton = findViewById(R.id.um);
mDoisButton = findViewById(R.id.dois);
mTresButton = findViewById(R.id.tres);
mQuatroButton = findViewById(R.id.quatro);
mCincoButton = findViewById(R.id.cinco);
mSeisButton = findViewById(R.id.seis);
mSeteButton = findViewById(R.id.sete);
mOitoButton = findViewById(R.id.oito);
mNoveButton = findViewById(R.id.nove);
mZeroButton = findViewById(R.id.zero);
mCancelarButton = findViewById(R.id.cancelar);
mLigarButton = findViewById(R.id.ligar);
mPausarButton = findViewById(R.id.pausar);
```

Fonte: Autor do texto

Cada tecla, quando acionada, envia o caractere correspondente para o Arduíno via *bluetooth*, através da função *sendData()*, como pode ser visto na figura 29.

Figura 29: Função executada quando a tecla é acionada

```
mUmButton.setOnClickListener((v) -> { sendData("1"); });
mDoisButton.setOnClickListener((v) -> { sendData("2"); });
mTresButton.setOnClickListener((v) -> { sendData("3"); });
```

Fonte: Autor do texto

O método `sendData()` recebe o caractere por parâmetro e usa um bloco *try-catch* para enviar o comando via *bluetooth*. O bloco *try-catch* é necessário para tratar exceções, ou seja, erros que podem ocorrer em tempo de execução que causariam o mau funcionamento do aplicativo. Neste caso, no bloco *try* é a parte do código responsável pelo envio dos dados, e no bloco *catch*, o tratamento, caso o erro ocorra.

Figura 30: Método `sendData()`

```
// Método que envia os dados
private void sendData (String message){
    byte[] msgBuffer = message.getBytes();

    try {
        //tenta enviar os dados para o bluetooth
        outputStream.write(msgBuffer);
    } catch (IOException e) {
        //se deu erro é porque o bluetooth perdeu a conexão, exibe a mensagem e finaliza
        Toast.makeText(getApplicationContext(), text: "ERROR - Device not found", Toast.LENGTH_SHORT).show();
        finish();
    }
}
```

Fonte: Autor do texto

- **VozClass:** esta classe permite o envio de comandos via *bluetooth* por voz. Ela possui em termos de *layout*, apenas um botão que, ao clicar, dá início ao reconhecimento de voz.

Para fazer o reconhecimento de voz, a Google fornece ao Android alguns recursos que permitem converter a fala em texto, ou seja, a fala é interpretada como texto e não propriamente compreendida. O recurso utilizado nesse projeto foi a criação da *intent RecognizerIntent*. Ela controla um grande número de argumentos, como por exemplo, o idioma que está sendo falado. O texto é interpretado e o resultado é retornado em forma de uma lista de *strings* e pode ser acessado através do método `OnActivityResult()`.

Figura 31: Criação da intent RecognizerIntent

```
Intent intent = new Intent(RecognizerIntent.ACTION_RECOGNIZE_SPEECH);
intent.putExtra(RecognizerIntent.EXTRA_LANGUAGE_MODEL,
    RecognizerIntent.LANGUAGE_MODEL_FREE_FORM);
intent.putExtra(RecognizerIntent.EXTRA_LANGUAGE, Locale.getDefault());
intent.putExtra(RecognizerIntent.EXTRA_PROMPT,
    getString(R.string.speech_prompt));
```

Fonte: Autor do texto

No método onActivityResult() foi feito o tratamento do texto recebido e o envio dos comandos via *bluetooth* do seguinte modo: o formato da fala deve seguir um padrão, por exemplo “x minutos e y segundos”, “x segundos”, “x minutos”. O texto recebido é separado em palavras em um vetor de *strings*. As palavras chaves como ‘minutos’ e ‘segundos’ são identificadas, e os números encontrados antes dessas palavras são salvos separadamente, pois são esses valores que serão enviados como o tempo escolhido via *bluetooth* para o micro-ondas através do método sendData(), já descrito anteriormente.

Alguns ajustes foram necessários no desenvolvimento do comando de voz. Como por exemplo: ao interpretar a fala para o texto, em algumas vezes pode ocorrer do retorno do número falado ser por extenso. Por isso, antes de enviar o comando, este deve ser convertido.

Figura 32: Conversão do número por extenso

```
switch (min)
{
    case "zero" : min = "0"; break;
    case "um" : min = "1"; break;
    case "dois" : min = "2"; break;
    case "três" : min = "3"; break;
    case "quatro" : min = "4"; break;
    case "cinco" : min = "5"; break;
    case "seis" : min = "6"; break;
    case "sete" : min = "7"; break;
    case "oito" : min = "8"; break;
    case "nove" : min = "9"; break;
}
```

Fonte: Autor do texto

Outro ajuste foi a necessidade do uso de *delay* antes do envio de cada caractere. No envio através do teclado isso não foi necessário, pois era enviado um caractere por vez. Já no caso do comando de voz, isso não acontece. Vários caracteres são enviados de uma vez. Por exemplo, para enviar o comando de um minuto para o micro-ondas, devem ser enviados os caracteres '1', '0', '0' e o caractere responsável por ligar o aparelho, 'L'. E isso não pode ser feito ao mesmo tempo, pois o código do Arduíno possui um *delay* entre ligar e desligar um relé para acionar a tecla correspondente no micro-ondas. Caso seja enviado sem esse tempo de espera, alguns comandos serão perdidos no caminho.

Figura 33: Exemplo de um delay antes de chamar o método sendData

```
mHandlerUm.postDelayed(() -> {  
    sendData("0");  
}, delayMillis: 1000);
```

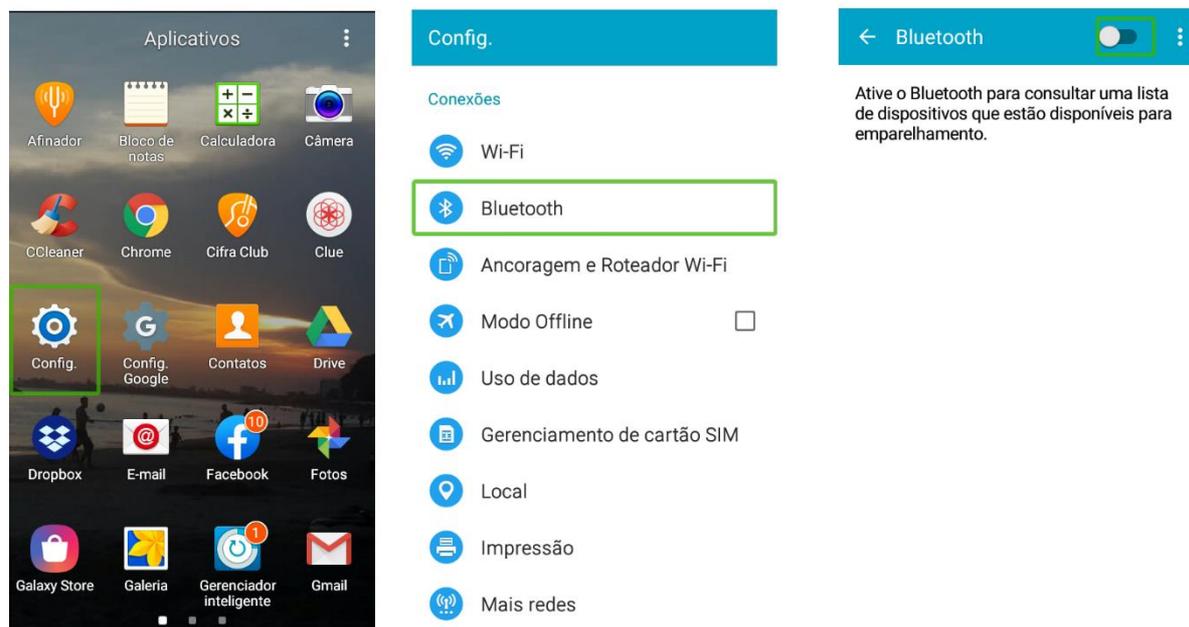
Fonte: Autor do texto

6.3 Configurações iniciais de utilização

O aplicativo proposto necessita que duas condições sejam atendidas, para o seu bom funcionamento. São essas:

- **O *bluetooth* do *smartphone* deve estar ligado** – esta é uma condição importante para o funcionamento do aplicativo. Todos os comandos são enviados através da conexão *bluetooth* com o micro-ondas. A seguir, na figura 34, podem ser vistas as etapas para realizar essa tarefa no *smartphone*.

Figura 34: Etapas para ligar o bluetooth no smartphone



Fonte: Autor do texto

Obs.: A ativação do bluetooth pode variar de acordo com a versão do Android, mas todos seguem basicamente o mesmo padrão

- **O Talkback deve estar ativado:** o Talkback é um serviço nativo do Android, que facilita a usabilidade para pessoas com deficiência visual. Ele traduz automaticamente as informações presentes na tela de texto para áudio, além de inserir sons e vibrações como resposta à cliques. A seguir estão ilustradas as etapas de ativação do Talkback no *smartphone*.

Figura 35: Etapas para a ativação do Talkback no smartphone

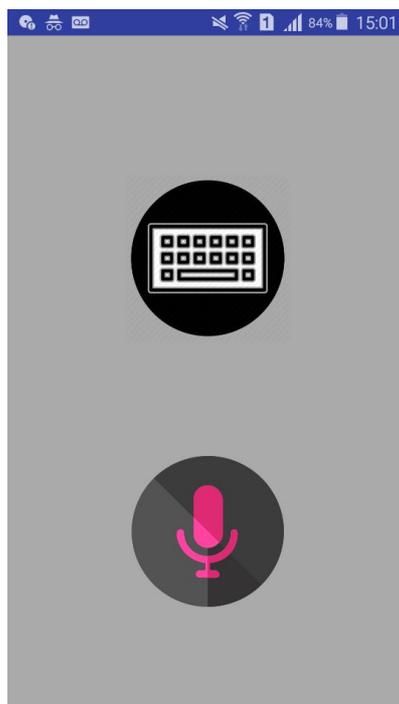


Fonte: Autor do texto

6.4 Interface do aplicativo e seu funcionamento

O aplicativo possui quatro telas ao todo. A tela inicial, como pode ser vista na figura 36, possui dois botões: o botão de teclado, que direciona o usuário para a área do teclado virtual; e o botão de voz, que direciona o usuário para a área do comando de voz.

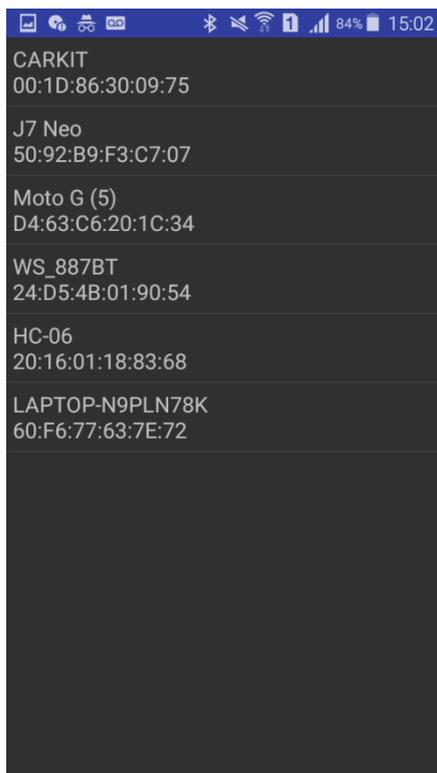
Figura 36: Tela inicial do aplicativo



Fonte: Autor do texto

Ao clicar em um dos botões, o usuário é direcionado para a tela onde será feita a conexão *bluetooth* entre os dispositivos. No momento em que essa tela é exibida, o aplicativo reproduz o seguinte áudio: “Antes de começar, conecte com o *bluetooth* HC-06”. Na tela é exibida uma lista de dispositivos *bluetooth* pareados com o *smartphone*. O usuário deficiente visual encontrará o dispositivo correto com o auxílio do TalkBack que transformará o texto exibido pela lista em áudio.

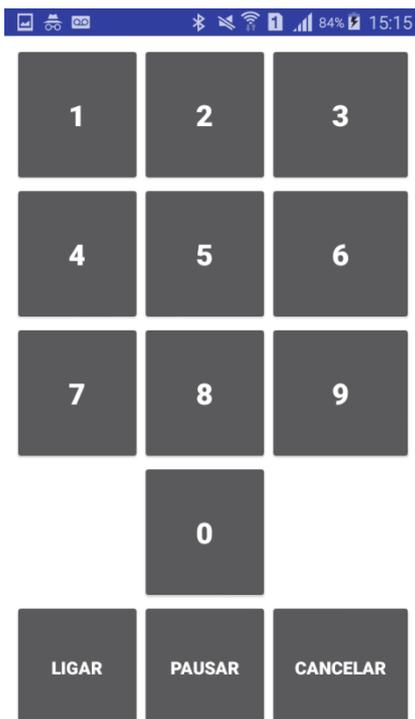
Figura 37: Lista de dispositivos bluetooth



Fonte: Autor do texto

Caso o usuário tenha escolhido utilizar o aplicativo no modo teclado, ele será direcionado para a seguinte tela (figura 38):

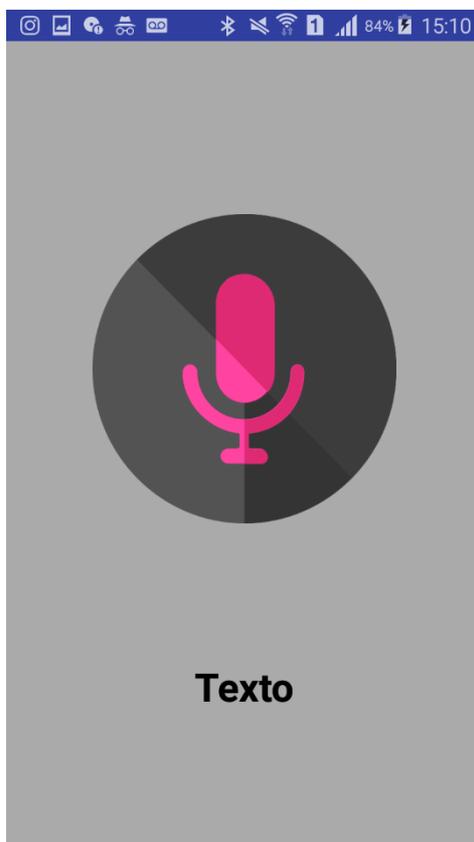
Figura 38: Teclado virtual do aplicativo



Fonte: Autor do texto

Já a tela do comando de voz pode ser vista na figura 39.

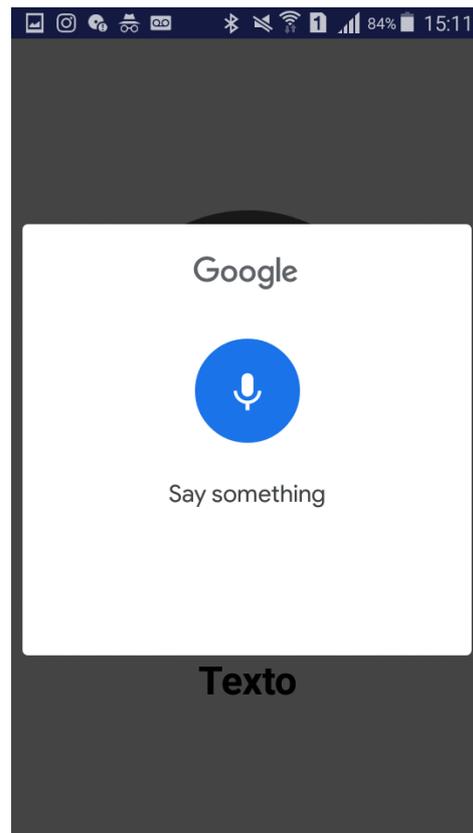
Figura 39: Tela do comando de voz do aplicativo



Fonte: Autor do texto

Ao entrar nessa tela, o aplicativo reproduz um áudio dizendo: “Clique no botão no centro da tela, aguarde um sinal sonoro e fale”. Quando o usuário clicar no botão, ativará a etapa de reconhecimento de voz e emitirá um sinal indicando que o usuário pode começar a falar. O texto retornado do reconhecimento de voz será escrito no lugar da palavra “Texto” indicada abaixo do botão, como pode ser visto na figura 39.

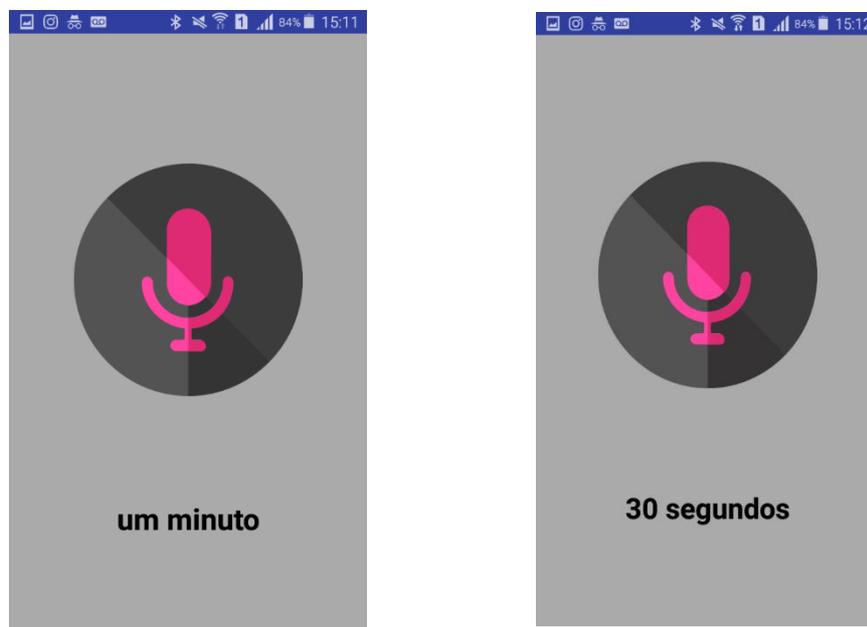
Figura 40: Reconhecimento de voz do aplicativo



Fonte: Autor do texto

Após o usuário terminar de falar, a tela anterior volta a ser exibida, mas agora com o texto interpretado escrito logo abaixo do botão. (figura 41)

Figura 41: Exemplos do comando de voz



Fonte: Autor do texto

6.5 Testes

Teste de usabilidade é uma técnica que avalia um produto ou um serviço, e é de extrema importância para que o desenvolvedor possa levar em consideração as necessidades do usuário final da aplicação. Nesta seção, serão apresentados os resultados dos testes realizados com usuários representativos para o aplicativo.

Os testes foram realizados com três usuários finais, levando em consideração seus níveis de conhecimento tecnológico e idades, como consta na tabela 1:

Tabela 1: Participantes do teste de usabilidade

Usuário	Idade	Nível de conhecimento tecnológico
Luisa	30 anos	Intermediário
Elizabeth	56 anos	Baixo
Dominique	23 anos	Avançado

Fonte: Autor do texto

Com relação aos usuários participantes, a usuária Luisa é a única que possui cegueira total. As outras duas voluntárias apresentam apenas alguns problemas leves de visão que não prejudicariam no uso do aplicativo. Por isso, para uma melhor experiência, essas participantes realizaram os testes de olhos vendados, guiando-se apenas pelas instruções em áudio do aplicativo.

Visando testar a usabilidade da aplicação, foi requisitado aos participantes que realizassem algumas tarefas. Essas tarefas foram divididas em duas partes:

1. **Parte do teclado virtual:** nessa parte, os usuários teriam que selecionar três tempos diferentes, seguidos do comando 'ligar'. Os tempos propostos foram os seguintes:
 - a. Selecionar o tempo de **30 segundos**, seguido do comando 'ligar';
 - b. Selecionar o tempo de **7 minutos**, seguido do comando 'ligar';
 - c. Selecionar o tempo de **2 minutos e 10 segundos**, seguido do comando 'ligar';

2. Parte do comando de voz: Nessa parte, as tarefas propostas foram:

- a. Falar o tempo de **30 segundos**;
- b. Falar o tempo de **2 minutos e 10 segundos**;
- c. Falar o comando **'pausar'**;
- d. Falar o comando **'cancelar'**.

Após a realização dos testes, os usuários responderam cinco perguntas sobre a experiência, para relatarem suas dificuldades e opiniões sobre o aplicativo.

Tabela 2: Perguntas feitas nos testes

Número	Pergunta
1	Sentiu dificuldade em realizar algumas das ações propostas? Se sim, quais? As dificuldades foram em relação ao aplicativo ou ao modo de funcionamento do Talkback (leitor de tela)?
2	Qual dos modos foi mais fácil de usar (Teclado ou Voz)?
3	O que você considera como ponto positivo da aplicação?
4	O que você considera como ponto negativo da aplicação?
5	Sugestões?

Fonte: Autor do texto

As respostas dos usuários estão presentes nas tabelas 3, 4 e 5:

Tabela 3: Resposta do usuário 1 - Luisa

Pergunta	Resposta
1	Não.
2	Os dois modos foram fáceis de usar.
3	O aplicativo se mostrou útil e fácil de usar. O comando de voz é uma excelente alternativa para o controle do micro-ondas.
4	Um ponto negativo é a aplicação só ter versão para Android. Uma versão iOS seria excelente, pois o VoiceOver é muito mais eficiente como leitor de tela do que o Talkback.
5	Gostaria de uma versão do aplicativo para iPhone. Tenho certeza de que me ajudaria muito.

Fonte: Autor do texto

Tabela 4: Resposta do usuário 2 - Elizabeth

Pergunta	Resposta
1	Sim, todas as dificuldades foram em relação ao modo de funcionamento do Talkback, por não ter costume de usar o <i>smartphone</i> deste modo.
2	A parte do teclado foi mais simples de usar.
3	A interface do aplicativo é simples, apresenta poucos botões e poucas funcionalidades, evitando a confusão do usuário.
4	O ponto negativo da aplicação é depender do leitor de tela Talkback, pois este é de difícil manuseio.
5	Usar outro leitor de tela ou utilizar leitor de tela próprio do aplicativo.

Fonte: Autor do texto

Tabela 5: Resposta do usuário 3 - Dominique

Pergunta	Resposta
1	Não.
2	Controle por teclado virtual.
3	O aplicativo é de fácil manuseio, aumentando a acessibilidade de pessoas com necessidades especiais.
4	Não apresenta confirmação do tempo escolhido através do teclado.
5	Incluir o feedback falado confirmando o tempo escolhido no controle pelo teclado virtual.

Fonte: Autor do texto

Os testes permitiram que algumas conclusões fossem tiradas sobre o aplicativo:

- Foi possível perceber, durante os testes, a diferença da facilidade de uso entre um usuário deficiente visual e um usuário vidente. O aplicativo se mostrou muito mais fácil de ser usado por um deficiente visual, já que este já está habituado a usar o *smartphone* com auxílio de leitores de telas.
- A interface é simples e de fácil manuseio, apresenta poucas informações e etapas para realizar uma atividade, o que evita a confusão do usuário;
- O Talkback dificulta um pouco o uso, em sua configuração padrão, não só do aplicativo, mas do *smartphone* em geral. Uma resolução para essa questão, seria uma configuração personalizada, de acordo com a preferência do usuário. O Talkback apresenta uma área de 'ajustes' ou 'configurações', que fica na mesma tela de ativação do aplicativo. Nessa parte, o que poderia ajudar o usuário, seria ativar o modo de navegação por toque único, por exemplo. Já que a configuração padrão é a navegação com dois toques para selecionar algo.

7 CONCLUSÃO E TRABALHOS FUTUROS

Este projeto teve como objetivo o desenvolvimento de um aplicativo para *smartphones* baseado na plataforma Android, a fim de auxiliar pessoas portadoras de deficiência visual no uso do micro-ondas. A ideia foi de usar a tecnologia como ferramenta de acessibilidade. A pesquisa realizada durante o presente trabalho, demonstrou uma carência de acessibilidade nos eletrodomésticos com relação aos deficientes visuais, principalmente no que diz respeito ao micro-ondas. Este, por sua vez, representa uma barreira para os deficientes visuais. Com um teclado liso e sem relevo ou representação em Braille, torna-se uma missão impossível para o deficiente o manuseio do aparelho.

No Brasil, a deficiência com maior incidência é a visual. Mais de trinta milhões de pessoas apresentam a deficiência. E, este público, precisa de adaptações no mundo ao seu redor para viver com maior independência e se sentir incluído na sociedade. Pensando nisso, o tema proposto busca trazer maior independência no ambiente doméstico, realizando uma adaptação no modo de utilizar o micro-ondas. Em termos de adaptar o eletrodoméstico, adesivos em Braille seriam uma opção para permitir que os deficientes visuais ‘enxerguem’ as teclas do aparelho. Porém, não seria suficiente para atender todo o público. Isso se dá pelo fato de que apenas 10% dos deficientes visuais no Brasil são alfabetizados na linguagem Braille.

Partindo do objetivo, foi desenvolvido um aplicativo que permite realizar o comando do micro-ondas através da conexão *bluetooth*. Para isso, foi necessária a adaptação do eletrodoméstico para que este estivesse apto a receber os comandos via *bluetooth*. O aplicativo *MicroBlind* realiza de forma simples e fácil de usar, o controle do micro-ondas e permite que o usuário escolha a forma de utilização: teclado virtual ou comando de voz. Ele apresenta ainda, auxílio de áudios para indicar o que o usuário deve fazer, simplificando ainda mais o seu uso. Sendo assim, o resultado esperado, de gerar acessibilidade com a ajuda da tecnologia, foi alcançado.

Como trabalho futuro, tem-se por meta mapear o restante das teclas do micro-ondas, que não foram contempladas nesse trabalho. Teclas essas, com funções especiais do tipo: pipoca, arroz, descongelar, potência, entre outras. Além disso, pretende-se atender às sugestões dos voluntários que realizaram o teste, para a melhora do aplicativo.

É possível pontuar como característica positiva do trabalho realizado, a possibilidade de que o protótipo possa ser de fato aplicado no mundo real e seja utiliza-

do como uma tecnologia assistiva para os deficientes visuais, devido o projeto apresentar a vantagem de poder ser continuamente melhorado e representar uma adaptação simples e de baixo custo.

É possível concluir, com o presente trabalho que é necessário considerar os deficientes como um mercado consumidor, e realizar ajustes e adaptações em produtos no geral, para atender qualquer tipo de carência.

8 REFERÊNCIAS

- [1] P. da República, “Lei brasileira de inclusão da pessoa com deficiência,” 2015. Disponível em: https://www.planalto.gov.br/ccivil_03/ato2015-2018/2015/lei/l13146.htm.
- [2] P. de Proteção e Defesa do Consumidor PROCON, “Relatório de acessibilidade a produtos,” 2008. Disponível em: <http://www.procon.sp.gov.br/pdf/relatorio-acessibilidade-produtos-2008.pdf>.
- [3] O. M. da Saúde, “Estatísticas da deficiência visual,” 2012. Disponível em: <https://www.fundacaodorina.org.br/a-fundacao/deficiencia-visual/estatisticas-da-deficiencia-visual/>.
- [4] W. H. Organization, “World report on disability,” 2011. Disponível em: https://www.who.int/disabilities/world_report/2011/report.pdf.
- [5] F. D. N. para Cegos, “Convenção da onu sobre direitos das pessoas com deficiência,” 2018. Disponível em: <https://www.fundacaodorina.org.br/a-fundacao/deficiencia-visual/convencao-da-onu-sobre-direitos-das-pessoas-com-deficiencia/>.
- [6] I. B. de Geografia e Estatística IBGE, “Censo ibge 2010 pessoas com deficiência,” 2010. Disponível em: <http://www.pessoacomdeficiencia.gov.br/app/sites/default/files/publicacoes/cartilha-censo-2010-pessoas-com-deficiencia-reduzido.pdf>.
- [7] R. D. Morcelli and R. D. Seabra, “Inclusão digital e deficiência visual: análise do uso de ferramentas de comunicação pela internet,” *Informática na educação: teoria & prática*, vol. 17, no. 1, 2014.
- [8] F. D. N. para Cegos, “O que é deficiência?,” 2018. Disponível em: <https://www.fundacaodorina.org.br/a-fundacao/deficiencia-visual/o-que-e-deficiencia/>.

- [9] P. do MEC Marta Gil, “Deficiência visual,” 2000. Disponível em: <http://portal.mec.gov.br/seed/arquivos/pdf/deficienciavisual.pdf>.
- [10] G. do Brasil, “Acessibilidade,” Dezembro 2017. Disponível em: <http://www.brasil.gov.br/acessibilidade>.
- [11] S. E. dos Direitos da Pessoa com Deficiência, “Acessibilidade.” Disponível em: <http://www.pessoacomdeficiencia.gov.br/app/acessibilidade-0>.
- [12] P. Deitek and W. A. Deitel, H., *Android 6 Para Programadores - Uma Abordagem Baseada em Aplicativos*. No. 3, Bookman, 2016.
- [13] S. G. A. Accessibility., “Primeiros passos no Android com o talkback,” 2018. Disponível em: <https://support.google.com/accessibility/Android/answer/6283677?hl=pt-BR>.
- [14] C. S. Media, “Google talkback review.” Disponível em: <https://www.commonsemmedia.org/app-reviews/google-talkback>.
- [15] B. F. Flop, “Arduíno,” 2014. Disponível em: <https://www.filipeflop.com/blog/o-que-e-arduino/>.
- [16] Buildbot, “Módulo bluetooth,” 2015. Disponível em: <http://buildbot.com.br/blog/configuracao-do-modulo-bluetooth-hc-06-com-arduino/>.
- [17] Wikipedia, “Relé,” 2015. Disponível em: <https://pt.wikipedia.org/wiki/Rel%C3%A9>.
- [18] TechTudo, “Cinco tecnologias para ajudar pessoas com deficiência visual.” Disponível em: <https://www.techtudo.com.br/listas/noticia/2015/04/cinco-tecnologias-para-ajudar-pessoas-com-deficiencia-visual.html>