



UNIVERSIDADE FEDERAL RURAL DO SEMI-ÁRIDO
PRÓ-REITORIA DE GRADUAÇÃO
CENTRO DE CIÊNCIAS EXATAS E NATURAIS
BACHARELADO EM CIÊNCIA E TECNOLOGIA

BIANCA STEPHANIE GUIMARÃES MORAIS

**PROTÓTIPO DE SISTEMA DE AUTOMAÇÃO RESIDENCIAL PARA O AUXÍLIO
DE PESSOAS IDOSAS E COM DIFICULDADES DE LOCOMOÇÃO**

MOSSORÓ/RN

2018

BIANCA STEPHANIE GUIMARÃES MORAIS

PROTÓTIPO DE SISTEMA DE AUTOMAÇÃO RESIDENCIAL PARA O AUXÍLIO DE
PESSOAS IDOSAS E COM DIFICULDADES DE LOCOMOÇÃO

Trabalho de conclusão de curso apresentado à Universidade Federal Rural do Semi-Árido (UFERSA) e ao Centro de Ciências Exatas e Naturais (CCEN), como requisito para a obtenção do título de Bacharel em Ciência e Tecnologia.

Orientador: Prof. Dr. Álvaro Alvares de Carvalho César Sobrinho.

Co-orientador: Prof. Dr. Helder Fernando de Araújo Oliveira.

MOSSORÓ/RN

2018

© Todos os direitos estão reservados a Universidade Federal Rural do Semi-Árido. O conteúdo desta obra é de inteira responsabilidade do (a) autor (a), sendo o mesmo, passível de sanções administrativas ou penais, caso sejam infringidas as leis que regulamentam a Propriedade Intelectual, respectivamente, Patentes: Lei nº 9.279/1996 e Direitos Autorais: Lei nº 9.610/1998. O conteúdo desta obra tomar-se-á de domínio público após a data de defesa e homologação da sua respectiva ata. A mesma poderá servir de base literária para novas pesquisas, desde que a obra e seu (a) respectivo (a) autor (a) sejam devidamente citados e mencionados os seus créditos bibliográficos.

M827p Morais, Bianca Stephanie Guimarães.
 PROTÓTIPO DE SISTEMA DE AUTOMAÇÃO RESIDENCIAL
 PARA O AUXÍLIO DE PESSOAS IDOSAS E COM
 DIFICULDADES DE LOCOMOÇÃO / Bianca Stephanie
 Guimarães Morais. - 2018.
 62 f. : il.

 Orientador: Álvaro Alvares de Carvalho César
 Sobrinho.

 Coorientador: Helder Fernando de Araújo
 Oliveira.

 Monografia (graduação) - Universidade Federal
 Rural do Semi-árido, Curso de Ciência e
 Tecnologia, 2018.

 1. Automação. 2. Dificuldade de locomoção. 3.
 Idoso. 4. Arduino. 5. Android. I. Sobrinho,
 Álvaro Alvares de Carvalho César, orient. II.
 Oliveira, Helder Fernando de Araújo, co-orient.
 III. Título.

O serviço de Geração Automática de Ficha Catalográfica para Trabalhos de Conclusão de Curso (TCC's) foi desenvolvido pelo Instituto de Ciências Matemáticas e de Computação da Universidade de São Paulo (USP) e gentilmente cedido para o Sistema de Bibliotecas da Universidade Federal Rural do Semi-Árido (SISBI-UFERSA), sendo customizado pela Superintendência de Tecnologia da Informação e Comunicação (SUTIC) sob orientação dos bibliotecários da instituição para ser adaptado às necessidades dos alunos dos Cursos de Graduação e Programas de Pós-Graduação da Universidade.

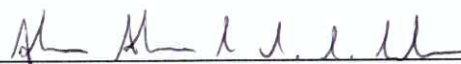
BIANCA STEPHANIE GUIMARÃES MORAIS

**PROTÓTIPO DE SISTEMA DE AUTOMAÇÃO RESIDENCIAL PARA O AUXÍLIO
DE PESSOAS IDOSAS E COM DIFICULDADES DE LOCOMOÇÃO**

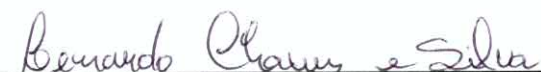
Monografia apresentada a Universidade
Federal Rural do Semi-Árido como
requisito para obtenção do título de
Bacharel em Ciência e Tecnologia.

Defendida em: 12 / 09 / 2018.

BANCA EXAMINADORA


Prof. Dr. Álvaro Alvares de Carvalho César Sobrinho (UFERSA)
Presidente


Prof. Dr. Helder Fernando de Araújo Oliveira (UFERSA)
Membro Examinador


Prof. Dr. Lenardo Chaves e Silva (UFERSA)
Membro Examinador


Prof. Me. Robson Locatelli Macedo (UFERSA)
Membro Examinador

*Dedico esse trabalho aos meus pais,
Antonio Wilton de Moraes Junior, Naiana
Narechy e ao meu irmão Arthur Wilton.*

*Dedico aos meus avôs, Nilton Rosado (In
Memoriam) e Manoel Victor (In
Memoriam), sei que onde estiverem,
estarão muito felizes por mim.*

AGRADECIMENTOS

Agradeço primeiramente a Deus por ter me concedido força e fé para chegar até onde cheguei hoje e por todas as bênçãos derramadas em minha vida.

Agradeço aos meus pais, Antônio Wilton de Moraes Junior e Naiana Narrechy Guimarães Moraes, por todo o amor, ensinamento, inspiração e apoio que me deram desde sempre. Agradeço por serem pais excelentes que nunca me deixaram faltar em nada e sempre fizeram o possível para que eu pudesse alcançar todos os meus sonhos. Vocês são minha vida e todas as minhas conquistas e vitórias são por vocês.

Agradeço ao meu namorado, Matheus Leopoldo de Oliveira Miranda, por toda a ajuda, apoio, companheirismo, compreensão e conhecimento que me ajudou indireta e diretamente neste trabalho e na nossa vida, minhas conquistas e vitória também são por você.

Agradeço as minhas avós, Maria de Lourdes e Ivanilde Carlos, por todo o carinho, amor e apoio que tens me dado desde sempre, minhas vitórias também são por vocês.

Agradeço as minhas amigas, Karine, Fernanda, Alana, Lyla, Larissa, Mayla, Meire e a todas as outras não citadas aqui, pelo companheirismo e pela torcida por mim, vocês estão no meu coração.

Agradeço também a Natan Oliveira, pelo apoio e ajuda no primeiro protótipo deste aplicativo, seus ensinamentos me ajudaram muito a criar o segundo protótipo.

Agradeço aos meus orientadores Prof. Dr. Álvaro Alvares de Carvalho César Sobrinho e Prof. Dr. Helder Fernando de Araújo Oliveira por todo o companheirismo, compreensão, apoio, conhecimento e otimismo. Vocês foram excelentes orientadores, muito obrigado!

Agradeço a Universidade Federal Rural do Semi-Árido por todo o equipamento para que eu pudesse obter todo o conhecimento recebido até hoje.

Enfim, obrigado a todos, que direta e indiretamente contribuíram para a conclusão deste trabalho.

RESUMO

Este trabalho aborda a automação residencial, android e acessibilidade, focando no público pessoas idosas e com dificuldades de locomoção, como um meio de ajudar nas necessidades diárias e minimizar obstáculos como ligar luzes, ar condicionado, televisão, ventilador, entre outros. Neste meio é desenvolvido a criação de um protótipo de aplicativo Android focando na mobilidade, na praticidade, inclusão tecnológica e adaptação digital. O objetivo com este trabalho é desenvolver e implantar o uso de automação no auxílio de pessoas idosas e com dificuldades de locomoção, identificando suas necessidades utilizando técnicas de elicitación de requisitos. Diante disso implementar um protótipo para teste envolvendo *bluetooth*, como meio de comunicação entre plataformas Arduino e Android. Portanto foi desenvolvido um aplicativo para facilitar o controle do ambiente, como foco na solução tecnológica. Foram utilizados os métodos de revisão literária e o desenvolvimento de diagramas em linguagem de modelagem unificada. Os resultados foram satisfatórios, considerando que o protótipo funcionou de forma simples, mas provando suas funções iniciais, sendo apenas utilizado para verificação de desempenho. Pode-se concluir que este trabalho serve como base para futuras pesquisas e base de melhoria no desenvolvimento de aplicativos voltados para as necessidades de idosos e/ou pessoas com dificuldade de locomoção.

Palavras-chave: Automação. Dificuldade de locomoção. Idoso. Arduino. Android. *Bluetooth*. Aplicativo.

ABSTRACT

This work addresses residential automation, android and accessibility, focusing on all the elderly and with difficulties of locomotion, air conditioning, television, fan, among others. This is a product developed in prototype application for Android, in practice, in the area of technology and digital adaptation. The objective is to develop and deploy the use of automation in the aid of elderly people with locomotion difficulties, as well as the use of requirements elicitation techniques. Faced with this, implement a prototype for the bluetooth test, as a means of communication between Arduino and Android platforms. Therefore an application for the control of the environment was developed, as the focus on the technological solution. Literature review methods and the development of unified modeling language diagrams were used. The results were satisfactory, considering that the prototype worked in a simple way, but providing its initial functions, being verified in a non-formal way. This does not exist for work needs prior to the prospect needs and needs in the development of the application to the needs of elderly and / or people with locomotion.

Keywords: Automation. Difficulty of locomotion. Old man. Arduino. Android. *Bluetooth*. App.

LISTA DE FIGURAS

FIGURA 1 - PLACA ARDUINO MEGA 2650.....	21
FIGURA 2 - PRINTSCREEN DO MODELO VISUAL DO SISTEMA EXTENDEDHAND.	26
FIGURA 3 - DIAGRAMA DE CASO DE USO COMPLETO DO APLICATIVO	32
FIGURA 4 - DIAGRAMA DE PACOTES COMPLETO DO APLICATIVO.	34
FIGURA 5 - PACOTE MODELO	35
FIGURA 6 - PACOTE CONTROLE	36
FIGURA 7 - PACOTE BANCO DE DADOS DAO	36
FIGURA 8 - PACOTE VISÃO	37
FIGURA 9 - DIAGRAMA DE CLASSES	38
FIGURA 10 - DIAGRAMA DE OBJETOS DO CASO DE USO CADASTRO DE COMPONENTE	39
FIGURA 11 - DIAGRAMA DE OBJETOS DO CASO DE USO AGENDAMENTO DE COMPONENTE.	40
FIGURA 12 - DIAGRAMA DE SEQUÊNCIA DO CASO DE USO CADASTRO DE COMPONENTE.	41
FIGURA 13 - DIAGRAMA DE SEQUÊNCIA DO CASO DE USO AGENDAMENTO DE COMPONENTE.	42
FIGURA 14 - DIAGRAMA DE ATIVIDADE DO CASO DE USO CADASTRO DE COMPONENTE.	43
FIGURA 15 - DIAGRAMA DE ATIVIDADE DO CASO DE USO AGENDAMENTO DE COMPONENTE.	44
FIGURA 16 - DIAGRAMA DE ESTADO DA CLASSE AGENDAMENTO DE COMPONENTE.	45
FIGURA 17 - DIAGRAMA DE IMPLANTAÇÃO DO APLICATIVO.....	46
FIGURA 18 - A) INTERFACE GRÁFICA DO USUÁRIO INICIAL DO APLICATIVO. B) INTERFACE GRÁFICA DO USUÁRIO MENU PRINCIPAL.....	47
FIGURA 19 – A) MENU DE NAVEGAÇÃO DA INTERFACE GRÁFICA DO USUÁRIO. B) INTERFACE GRÁFICA DO USUÁRIO PARA INSERIR COMPONENTE.....	48
FIGURA 20 - A) INTERFACE GRÁFICA DO USUÁRIO COM A LISTA DE COMPONENTES SALVOS. B) INTERFACE GRÁFICA DO USUÁRIO PARA EXCLUSÃO DE COMPONENTES	48
FIGURA 21 – A) INTERFACE GRÁFICA DO USUÁRIO PARA INSERÇÃO DE NOVOS COMPONENTES. B) INTERFACE GRÁFICA DO USUÁRIO DO MENU DE CONFIGURAÇÕES.	49
FIGURA 22 - INTERFACE GRÁFICA DO PROTÓTIPO NO PROGRAMA ANDROID STUDIO	51
FIGURA 23 - INTERFACE GRÁFICA DO PROTÓTIPO EM UM SMARTPHONE. ...	52
FIGURA 24 - PROTÓTIPO DO APLICATIVO INICIALMENTE COM O LED DESLIGADO	53
FIGURA 25 - PROTÓTIPO DO ARDUINO E APLICATIVO EM FUNCIONAMENTO, COM O LED LIGADO.	53
FIGURA 26 - CIRCUITO ARDUINO NA PLACA PROTOBOARD COM O MÓDULO BLUETOOTH.....	54

LISTA DE TABELAS

TABELA 1 - APLICATIVOS CORRELATOS	28
TABELA 2 - REQUISITOS DO SISTEMA	62

LISTA DE SIGLAS E ABREVIATURAS

API	<i>Application Programming Interface</i>
APPS	<i>Aplicativos</i>
DAO	<i>Data Access Object</i>
ICSP	<i>In Circuit Serial Programming</i>
IDE	<i>Integrated Development Environment</i>
IoT	<i>Internet of Things</i>
IP	<i>Internet Protocol</i>
LED	<i>Light Emitting Diode</i>
MVC	<i>Model-View-Control</i>
OA	<i>Open Handset Alliance</i>
PWM	<i>Pulse Width Modulation</i>
SDK	<i>Software development kit</i>
SQLite	<i>Structured Query Language</i>
UARTS	<i>Universal Synchronous/Asynchronous Receiver/Transmitter</i>
UML	<i>Unified Modeling Language</i>

SUMÁRIO

1 INTRODUÇÃO	12
1.1 PROBLEMÁTICA	13
1.2 JUSTIFICATIVA	14
1.3 OBJETIVOS	15
1.4 METODOLOGIA DE PESQUISA	15
1.5 ORGANIZAÇÃO DO DOCUMENTO	16
2 REFERENCIAL TEÓRICO.....	17
2.1 ANDROID	17
2.2 JAVA	18
2.3 PLATAFORMA DE COMUNICAÇÃO.....	19
2.4 ARDUINO	20
2.5 AUTOMAÇÃO RESIDENCIAL	21
3 TRABALHOS RELACIONADOS	23
3.1 AUTOMAÇÃO RESIDENCIAL E SUAS USABILIDADES	23
3.2 SISTEMAS E ACESSIBILIDADE	25
3.3 APLICATIVOS E ACESSIBILIDADE	26
3.4 APLICAÇÕES CORRELATAS	28
4 SISTEMA DE AUTOMAÇÃO	30
4.1 DESCOBERTA E ELICITAÇÃO DE REQUISITOS	30
4.2 PROJETO DO SISTEMA COM UML	31
4.2.1 DIAGRAMA DE CASO DE USO	32
4.2.2 DIAGRAMA DE PACOTES	34
4.2.3 DIAGRAMA DE CLASSES.....	35
4.2.4 DIAGRAMA DE OBJETOS	39
4.2.5 DIAGRAMA DE SEQUÊNCIA	40
4.2.6 DIAGRAMA DE ATIVIDADE	42
4.2.7 DIAGRAMA DE ESTADO	44
4.2.8 DIAGRAMA DE IMPLANTAÇÃO	45
4.3 INTERFACES GRÁFICAS DO USUÁRIO	46
4.4 PROTÓTIPO DESENVOLVIDO	49
5 CONCLUSÕES	55
REFERÊNCIAS.....	56
APÊNDICE A – TABELA DE REQUISITOS DO SISTEMA E DESCRIÇÃO.....	59

1 INTRODUÇÃO

Um sistema de automação utiliza processos automáticos que comandam e/ou controlam mecanismos para seu funcionamento. Sendo assim, segundo Assis *et al* (2016) a automação ou *Automatus* (latim) envolve técnicas computadorizadas ou mecânicas que reduzem ou até cessa o uso de mão-de-obra, significando também mover-se por si. Esse é o caso da automação residencial: controlar pequena e/ou media residência com o uso de um controle remoto que está em mãos, no caso deste trabalho, no *smartphone*. A automação residencial já é uma realidade na indústria há muitos anos, dada a necessidade de automatizar tarefas nas fabricas, diminuir trabalho humano, utilizar-se de trabalhos que não pudessem ter a mão de obra humana, permitindo ter controle, funcionalizar locais e objetos de forma prática e eficiente. Este é o objetivo da automação, facilitar rotinas e otimizar recursos, proporcionando flexibilidade apenas programando na palma da mão.

Com o crescimento acelerado das funcionalidades tecnológicas como a mobilidade, a localização em tempo real, os armazenamentos em nuvem, a completude de necessidades e a interatividade, em um único aparelho, fabricantes, operadoras e desenvolvedores de *smartphones* tentam acompanhar o avanço tecnológico, implementando novas ideias para manter-se competitivo no mercado.

Segundo Nichele (2010), ter a capacidade de ligar ou desligar equipamentos a partir de cadeira de rodas ou da cama é de grande impacto para portadores de deficiência, servindo também para auxiliar pessoas idosas. Abrir uma porta, ligar uma luz ou um ventilador, podem ser simples atividades para a maior parte das pessoas, mas para a parcela portadora de necessidades especiais físicas, estas podem ser limitações corriqueiras, principalmente quando não tem condições que tornem mais simples. Atualmente, a concepção de condição adequada para um local acessível a deficientes poderia ser rampas, apoios, barras e adaptação em locais específicos, sendo estes citados, comuns. Entretanto, podem não ser suficientes para atender as necessidades dos mesmos. Portanto, para ampliar os campos da acessibilidade que são inclusão tecnológica e adaptação digital, deve-se incorporar a automação como a evolução tecnológica que pode contribuir para ampliar o conforto e diminuir os obstáculos cotidianos dessas pessoas.

Conforme o Instituto Brasileiro de Geografia e Estatística (IBGE, 2010), 46 milhões de brasileiros apresentam algum tipo de incapacidade, ou seja, de 24% da população total, apresentaram algum tipo de incapacidade ou deficiência.

Sendo assim, pesquisas sobre como ampliar o campo da acessibilidade tem se expandido a medida que a necessidade por reduzir obstáculos para pessoas idosas e deficientes, vem sendo ampliada. Focando no contexto tecnológico como uma solução, os equipamentos devem reunir controle e processo, tornando atividades diárias mais simples e ágeis, ou seja, dotar ambientes com soluções tecnológicas que permitam independência e que tarefas cotidianas possam ser feitas de forma descomplicada. Pequenos gestos cotidianos como controlar iluminação e acionar aparelhos a partir de um aplicativo, podem ser exemplos de como a automação pode ser expandida a todos os públicos, pois a flexibilidade é uma realidade comum da evolução tecnológica.

Neste contexto, um exemplo de plataforma que pode ser utilizada na implementação deste tipo de automação é o Arduino. Com o Arduino, é possível construir diversas aplicações que trazem a tecnologia para o meio residencial, em virtude de características como, por exemplo, baixo custo, baixo consumo de energia, tamanho reduzido e facilidade de aprendizado e utilização. A partir de um *software* desenvolvido para a plataforma Android, pode-se disponibilizar um controlador de simples acesso, sendo necessário apenas um *smartphone* ou *tablet* com acesso à Internet.

1.1 PROBLEMÁTICA

Nesse contexto, emerge a problemática associada ao projeto e desenvolvimento de um protótipo de sistema de baixo custo, móvel e facilmente acessível, além da implantação do uso da tecnologia e automação para identificar e adaptar às necessidades de pessoas idosas e com dificuldades de locomoção como ligar luzes, objetos eletrônicos e eletrodomésticos, a fim de simplificar o dia-a-dia de pessoas idosas e com dificuldade de locomoção.

Além disso, outro “desafio” neste trabalho é a implementação de um ambiente automatizado por meio da integração da plataforma Arduino com dispositivos Android.

1.2 JUSTIFICATIVA

Com o crescimento acelerado de tecnologias, tornar a vida mais simples, fácil e segura tem sido o foco de uma boa parte da população nos últimos anos. A automação residencial é um desses focos, proporcionando conforto e auxílio em certos afazeres. Acender as luzes de casa e ligar um aparelho eletrônico estando em um local afastado são exemplos de tarefas. Este tipo de ação torna mais simples e cômodo a vida para os usuários idosos ou com dificuldade de locomoção, suprimindo algumas limitações com facilidade, quando se está em casa sem sair da cama ou sentado sem precisar mover a cadeira de rodas. Portanto existindo aplicativos que tentam adaptar e melhorar a vida dos usuários como eWeLink, IFTTT e Smart Life, que inserem a automação residencial nos seus lares, estes possuem objetivos semelhantes ao tema deste trabalho, mas neste caso, o foco é adaptar a necessidades específicas dos usuários tematizados.

Apesar do alto custo de vários projetos de automação, nada impede de procurar soluções mais acessíveis. Este tipo de solução pode ser implementada em pequenas e médias residências para uma melhor adaptação das necessidades das pessoas idosas e com dificuldade de locomoção e, de certa forma, para facilitar a vida dentro de casa.

O uso de celulares com Android também está em crescente expansão, pois segundo Lima (2018), “hoje, são mais de 2 bilhões de aparelhos ativos com Android e ele está em mais de 80% dos novos *smartphones* vendidos”, tanto pelo custo mediano, quanto pelo seu bom desempenho, sendo também uma plataforma aberta para a qual existe fácil implementação e criação de aplicativos. Esta plataforma é um meio interessante para a criação de um aplicativo para auxiliar na automação, especialmente por atingir uma boa parcela da população, sendo também simples e prático, em relação aos seus códigos de fácil acesso, que são disponibilizados em livros e Internet de simples a mais complexos tipos, e por fim, grande disponibilidade de aplicativos.

O mercado de smartphones está em constante crescimento. Segundo o IBGE (2015), em uma pesquisa realizada para verificar quantas pessoas acima de 10 anos possuem acesso a telefone móvel/celular, foi identificado que o total de 78% possuía acesso para uso pessoal. Isso deixa claro a elevada procura por

smartphones/celulares, que fornecem a diversas funções que viabilizam a praticidade pessoal.

1.3 OBJETIVOS

Este trabalho tem como objetivo principal desenvolver e implantar o uso de automação no auxílio de pessoas idosas e com dificuldades de locomoção, a partir de um protótipo com Arduíno e um aplicativo na plataforma Android, favorecendo a redução de alguns obstáculos cotidianos. Seus objetivos específicos são:

- Identificar necessidades de pessoas com dificuldades de locomoção, quando em suas residências, para eliciação de requisitos funcionais e não funcionais;
- Projetar aplicativo utilizando a linguagem de modelagem unificada (UML);
- Prototipar interfaces gráficas do usuário do sistema atendendo aos requisitos elicitados;
- Utilizar a tecnologia de comunicação sem fio para integrar a plataforma Arduino ao aplicativo Android.

1.4 METODOLOGIA DE PESQUISA

Como descrito anteriormente, o objetivo é criar um aplicativo de automação para usuários idosos e/ou com dificuldade de locomoção, com base nas pesquisas e análises de suas necessidades principais. Logo, elaboram-se as seguintes questões de pesquisa a seguir (QP):

- **QP1:** Quais as principais necessidades a serem atendidas com o aplicativo?
- **QP2:** Quais funcionalidades/opções, devem ser implementadas no aplicativo para que seja útil as demandas dos usuários em questão?

As questões abordadas serão utilizadas para influenciar, analisar e obter resultados deste trabalho. A terminologia Deficiência, Aplicativo e Automação foram utilizadas no processo de análise de palavras-chaves, levando em consideração seus sinônimos e abrangência de assunto. Alguns não focaram somente no tema necessitado, estando incluso uma ou mais categorias associadas, pois a abrangência das palavras-chaves inclui assuntos relevantes que ampliam os resultados deste documento.

Apenas estudos relacionados a automação residencial para ajudar pessoas idosas e com dificuldade de locomoção serão segmentados. As bases de dados utilizadas são a IEEE¹, PubMed², Scielo³, *Web of Science*⁴, ACM⁵ e Scopus⁶, fundamentais como base deste trabalho. Pesquisas no *Google Acadêmico* também foram de grande ajuda para inserção de artigos relevantes no tema abordado, a fim de ampliar a base de dados e enriquecer o trabalho.

1.5 ORGANIZAÇÃO DO DOCUMENTO

Este documento está estruturado em 5 capítulos além da introdução. No capítulo 2 é explanado uma introdução sobre Android, Java, Plataformas de Comunicação, Arduino, automação residencial, dentre outros assuntos que serão fundamentais para a explanação e o desenvolvimento desse trabalho. No capítulo 3 serão abordados trabalhos relacionados a Automação Residencial para pessoas idosas e com dificuldade de locomoção. No capítulo 4 é apresentada a análise literária de aplicativos relacionados ao trabalho, que serve como base para a aplicação proposta. No capítulo 5 são apresentados os resultados obtidos com o desenvolvimento do presente trabalho, assim como sua finalidade e funcionalidade, de acordo com a pesquisa. Finalizando com uma breve discursão acerca de tais resultados obtidos. No capítulo 6 é apresentado a conclusão sobre o trabalho, de acordo com a pesquisa, seu resultado final, limitações e trabalhos futuros.

¹ <https://www.ieee.org/>

² <https://www.ncbi.nlm.nih.gov/pubmed/>

³ <http://www.scielo.org/php/index.php>

⁴ <http://wok.mimas.ac.uk/>

⁵ <https://www.acm.org/>

⁶ <https://www.scopus.com/home.uri>

2 REFERENCIAL TEÓRICO

Com a busca de melhorias e conforto, a automação torna-se útil tanto para alguém que busca despreocupação e comodidade em seu ambiente familiar, como a ampliação de autonomia para pessoas com necessidades especiais. A independência é um benefício que não pode ser mensurado, e é algo que muitos buscam. Neste capítulo, são apresentados conceitos relacionados com o desenvolvimento deste trabalho, dentre os quais estão Android, Java, plataformas de comunicação, Arduino e automação residencial.

2.1 ANDROID

Android é um sistema operacional baseado em plataforma Linux, projetado para uso em *smartphones*, *tablets*, *TVs* e diversos outros aparelhos digitais. Serve para o desenvolvimento de aplicativos móveis, tendo uma interface visual rica e desenvolvimento flexível. A linguagem de programação Java é utilizada durante o desenvolvimento de aplicativos para Android (LECHETA, 2013). O Android foi desenvolvido pela OAH (*Open Handset Alliance*) um grupo liderado pela Google, que domina uma boa parte do mercado de telefonia celulares (LECHETA, 2013).

Cada plataforma tem um código identificador chamado *API Level* (*Application Programming Interface* – Interface de programação de aplicações). Esse é um número identificador valioso para as aplicações Android, uma vez que, ao desenvolver um aplicativo com a API, sabe-se quais *smartphones* estão aptos a executar a aplicação. (LECHETA, 2013).

Aplicativos móveis, também conhecidos como *Apps*, são programas projetados e desenvolvidos para serem dimensionadas e usadas em dispositivos eletrônicos móveis com capacidade de armazenamento e processamento de dados. Podem possuir conexão com Internet, sendo esses do tipo: *smartphones*, *tablets*, dentre outros. Apps estes que podem ser encontrados na loja virtual da Google – Android para *download*: *Play Store* (SILVA; PIRES; CARVALHO, 2015).

A plataforma para desenvolvimento do aplicativo utilizado nesse projeto será a Android, onde a criação desse aplicativo virá por meio da IDE ou ambiente de desenvolvimento integrado, *Android Studio*. Este programa pode criar aplicativos para todos os tipos de dispositivos Android, pois possui ferramentas para a criação de

aplicativos exclusivos de alta qualidade. Pode ser utilizado em computadores/*notebooks* com sistemas operacionais *Windows*, *Mac* e *Linux*.

Essa plataforma é conciliada com o *Android SDK*, que é usado para desenvolver as aplicações *Android*, juntamente com um emulador para teste de aplicativo, ou até mesmo teste em um *smartphone* com configurações compatíveis (LECHETA, 2013).

Como o Android é baseado em linguagem Java, no tópico seguinte há uma explanação sobre este assunto.

2.2 JAVA

Java é uma linguagem de programação e plataforma computacional lançada pela *Sun Microsystems* em 1995. Portanto, é uma linguagem extensamente usada no mundo, a fim de atender às necessidades de programação. É amplamente utilizada para a implementação de aplicativos e sistemas de *software* baseados na Internet e para dispositivos que possam se conectar pela mesma (DEITEL, 2017).

A linguagem de programação Java é bastante abrangente, usada em vários programas e sistemas operacionais, e em qualquer aparelho que o aceite (*i.e.*, possua uma máquina virtual Java). Neste quesito, a grande maioria de dispositivos é englobada, sendo assim:

Didaticamente, podemos imaginar o Java como um computador, uma máquina virtual. Não uma máquina qualquer, podemos considerá-la um Rolls Royce (alguém perguntaria: "Por que não Ferrari?". Bem, digamos que Java ainda não é um primor em velocidade). E, como todo carro de luxo que se preze, vem cheio de recursos. Temos suporte a impressora, telas gráficas, teclado, mouse, rede, arquivos, e muitas outras coisas. (FRÓES, 2008).

Java é composta por um conjunto de especificações, que inclui linguagem de programação, formato do arquivo, API e máquina virtual. Assim, emulando seu funcionamento em vários aparelhos existentes, que vão de microcomputadores à relógios inteligentes.

2.3 PLATAFORMA DE COMUNICAÇÃO

Wireless é uma tecnologia de comunicação que não faz uso de cabos e geralmente usa através de frequências de rádio, infravermelhos, entre outros.

“Tecnologia presente em muitos dos dispositivos atuais o *wireless* traz consigo grande mobilidade em relação ao seu predecessor as redes cabeadas. O padrão mais utilizado por redes *wireless* é o 802.11, padrão este definido pelo IEEE (*Institute of Electrical and Electronics Engineers*), instituto responsável por estabelecer padrões para computadores e dispositivos” (PINTO; GOMES, 2011, p.210).

Neste trabalho, será integrado o *bluetooth*, uma tecnologia de comunicação sem fio, juntamente com um *smartphone*. Como descrito anteriormente, *smartphone* é um dispositivo programável que propõe mobilidade, conectividade e aplicabilidade para seus usuários, tendo funcionalidades variadas como agenda, jogos, câmera, dispositivo de música, comunicação via chamadas telefônicas e via Internet, dentre vários outros recursos (RODRIGUES, 2009).

O *bluetooth* é considerado um protocolo padrão de comunicação sem fio, com baixo consumo de energia e baixo alcance. Seu propósito de criação foi eliminar as conexões físicas entre dispositivos de diferentes fabricantes. Ele oferece transferência de arquivos e transferência de dados entre dispositivos. Sua vantagem principal sobre outros dispositivos *wireless* é sua facilidade de conexão, conseguindo ter conexão com até oito aparelhos ao mesmo tempo sem necessidade de grande proximidade e alinhamento. Sua comunicação se dá por ondas eletromagnéticas e seu raio de alcance pode chegar até 100 metros (SILVA *et al.*, 2012).

Constituindo um padrão de comunicação baseado em *microchips* transmissores de baixo custo e usando comunicação via rádio, segundo Silva *et al.* (2012):

“Este protocolo tem como características base a faixa de operação ISM (*Industrial, Scientific and Medical*) centrada em 2.4 GHz e com largura de banda de um MHz, modulação por chaveamento de frequência gaussiana (GFSK, em inglês), comunicação de dados *full-duplex*, interface aérea com base técnica de espalhamento espectral por saltos em frequência (FHSS, em inglês) e transferência de informações com restrições de tempo. Valendo ressaltar que o *Bluetooth* inicialmente foi desenvolvido por um consórcio de empresas denominado SIG (*Special Interest Group*) e pouco tempo depois foi adotado pelo IEEE dando origem ao padrão 802.15 para redes PAN (*Personal Area Network*). ”

Sua arquitetura subdivide-se em: *piconet* e *scatternet*. Na *piconet* pode ser considerada uma rede com até 8 dispositivos conectados entre si. Já a associação entre duas ou mais *piconets* formam a *scatternet* que é uma rede maior (SILVA, *et al.*, 2012).

2.4 ARDUINO

Arduino é um pequeno computador programável no qual é possível processar entradas e saídas entre dispositivos/componentes conectados a ele. É chamado de plataforma de computação física ou embarcada, sendo isso um sistema que interage com o ambiente por meio de *hardware* e *software* (MCROBERTS, 2011).

“Assim, foi criada uma placa composta por um microcontrolador Atmel, circuitos de entrada/saída e que pode ser facilmente conectada à um computador e programada via IDE (*Integrated Development Environment*, ou Ambiente de Desenvolvimento Integrado) utilizando uma linguagem baseada em C/C++, sem a necessidade de equipamentos extras além de um cabo USB” (THOMSEN, 2014).

O objetivo do Arduino em sua criação era ser um dispositivo barato, funcional e prático de programar, sendo acessível a estudantes e projetistas amadores. Esta plataforma foi adotada pelo conceito de *hardware* livre, o qual significa que qualquer pessoa pode montar, modificar e personalizar o Arduino, sempre pelo mesmo *hardware* básico (THOMSEN, 2014).

Na Figura 1 está um exemplo de placa Arduino, modelo Mega, sua placa microcontroladora é baseada no *datasheet* ATmega2560, que possui 54 pinos de entrada/saída, sendo 15 usados como saída PWM, 16 entradas analógicas, 4 UARTS (portas seriais de hardware), oscilador de cristal de 16 MHz, conexão USB, conector de alimentação, conector ICSP (protocolo de comunicação) e um botão *reset*.

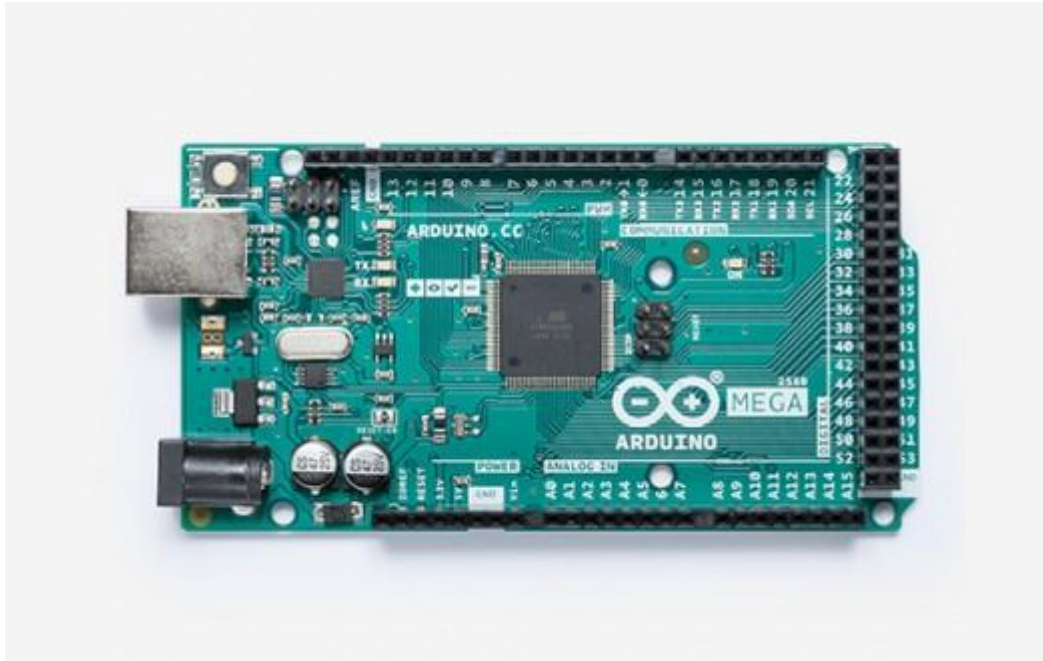


Figura 1 - Placa Arduino Mega 2560

FONTE: <https://store.arduino.cc/usa/arduino-mega-2560-rev3>

2.5 AUTOMAÇÃO RESIDENCIAL

Automação é um processo operacional estabelecido em determinado ambiente, no contexto deste trabalho, uma residência, que é controlado e executado por meio de dispositivos, sendo eles eletrônicos ou mecânicos, facilitando ou até substituindo o trabalho humano (ROCKENBACH, 2004).

A automação residencial representa a utilidade das tecnologias no ambiente doméstico, englobando em seu objetivo proporcionar mais conforto, praticidade, produtividade, eficiência, rentabilidade, e inúmeras outras vantagens. Grande parte das instalações da automação numa residência visa ter controle, seja por controle remoto e até mesmo via Internet, que vem se tornando mais popular devido aos serviços de comunicação, canais abertos, IPs (*Internet Protocol*), dentre outros recursos (WORTMEYER; FREITAS; CARDOSO, 2005).

O objetivo com a automação residencial é integrar tecnologias de fácil acesso à informação, Internet e segurança. Além de integrar com rede de dados, voz, imagens e multimídia. É obtido através de um projeto único que envolve infra-estrutura, dispositivos e *software* de controle, contendo apenas os equipamentos necessários e a implementação correta (WORTMEYER; FREITAS; CARDOSO, 2005).

Tornando-se estudada amplamente, a automação residencial tem metas a serem implementadas para satisfação do usuário. Sendo assim “A meta de um projeto de automação residencial é garantir ao usuário a possibilidade de controle e de acesso aos equipamentos instalados em sua residência, de dentro ou de fora da mesma” (WORTMEYER; FREITAS; CARDOSO, 2005). Por isso os autores procuram diversificar o olhar sobre essa inovação tecnológica, explicando de uma forma natural e de melhor entendimento, na seguinte citação:

“Um ambiente inteligente é aquele que otimiza certas funções inerentes à operação e administração de uma residência. É como se ela tivesse vida própria, com cérebro e sentidos. Uma casa inteligente permite que se tenha acesso a todos os sistemas em qualquer ponto da casa, que se controle as luzes, realize-se o agendamento de tarefas, assista-se a programas de vídeo em qualquer cômodo, visualize-se as crianças brincando enquanto se assiste à TV ou navegue-se pela Internet, não havendo a necessidade de se comprar um conjunto de equipamentos para todos os ambientes” (WORTMEYER; FREITAS; CARDOSO, 2005).

3 TRABALHOS RELACIONADOS

Neste capítulo são abordados alguns trabalhos relacionados ao tema explanado e apresentado neste documento. A seguir serão apresentados alguns estudos sobre Automação Residencial e Aplicativos para pessoas idosas e/ou com dificuldade de locomoção.

3.1 AUTOMAÇÃO RESIDENCIAL E SUAS USABILIDADES

No estudo de Singh *et al.* (2018) é apresentado um conceito de automação usando sensores e “A Internet das Coisas (IoT)” que, para os autores, é a interconexão entre vários dispositivos tecnológicos usados diariamente com acesso à Internet, permitindo comunicação entre eles.

Os autores abordam como o uso de sensores conectados a um Arduíno pode ser uma alternativa para medir a energia consumida em uma casa. Os dados de consumo são salvos em um banco de dados, onde é feito o cálculo do consumo, e, logo após, enviada uma notificação ao usuário. Os sensores detectam as mudanças em tempo real em todas as variáveis no local desejado. O microcontrolador recebe esses dados, analisa e assimila e, em seguida, aciona os reles conectados aos aparelhos, automatizando-os sem qualquer intervenção do usuário.

Segundo Silva (2014), a automação tem uma falta de padronização nos equipamentos e isso afeta diretamente no custo da implementação dos projetos. A Autora apresenta uma solução para automação residencial se baseando em redes sem fio e acessível em termos de custo. Ainda em sua monografia a autora busca automatizar uma residência com um sistema composto por sensores que incluem: movimento, gás, luminosidade e temperatura. Tudo controlado por um aplicativo Android juntamente com protocolos *Wi-Fi* e *ZigBee* para transmitir os dados. Ainda em seu estudo Silva (2014) aborda o quanto a automação residencial cresceu nas últimas décadas e o que a mesma vem buscando. Para a autora, os principais objetivos da automação residencial seriam: proporcionar conforto, auxílio e redução de consumo energético. Segundo a autora a falta de padronização no processo de automação é um problema e ressalta que, até o fechamento de seu estudo não havia encontrado uma solução que fosse conveniente e que atendesse de forma global todas as necessidades do mercado, onde o problema de falta de soluções possa acarretar

um aumento significativo de custo para as propostas dos produtos no mercado. Seu principal objetivo de seu estudo é promover automação residencial com tecnologia de baixo custo, usando o *Wi-Fi* por sua disponibilidade abrangente em dispositivos móveis e por ser a principal conexão com Internet atualmente. O *ZigBee* foi escolhido por sua eficiência energética e pelos sensores e circuitos que são alimentados por baterias. Trabalhou as camadas físicas dos protocolos, frequência do *Wi-Fi*, camadas de aplicação, dentre outros.

Segundo Nichele (2010), o crescente número de idosos, a necessidade e o desejo de ser independente das pessoas idosas e deficientes, os problemas enfrentados devido sua idade ou limitações físicas, são fatores contribuintes para a evolução da automação residencial. Afinal, é um meio de proporcionar conforto, auxílio e independência aos usuários idosos e deficientes físicos.

Nichele (2010) mostra a diversidade tecnológica que envolve a Automação Residencial e as aplicabilidades nela agregadas, com foco em idosos e deficientes físicos, descrevendo a funcionalidade da automação e sua implementação com a Internet, de acordo com a busca por um padrão mais simples de sistema, com adaptação às várias necessidades como: cumprir um ciclo diário de iluminação, ligar e desligar equipamentos e enviar avisos ao usuário. Tem como tema de maior enfoque a “desospitalização”, que para Nichele (2010):

“Implantação maciça das redes de acesso e com monitoramento remoto de grupos de risco, enfermos, portadores de deficiências, etc., reduz-se o custo do sistema de saúde e aumenta a utilização de métodos preventivos trazendo maiores benefícios a todos”.

A medicina aliada a telemetria pode monitorar pessoas em suas casas evitando a ida ao posto médico, servindo também de apoio a medicina preventiva, afirma Nichele (2010). Alguns exemplos como: monitoração de cardíacos em 24h tendo seus dados enviados ao hospital, diabéticos enviando seus resultados de exames diários ao sistema residencial automatizado, dados armazenados em um banco de dados do hospital, com funções de gerenciamento para avisar ao médico caso haja algum problema. Sabendo que o desenvolvimento de um sistema é algo que necessita de entendimento prévio das possibilidades e implicações no uso do mesmo. O autor usa em seu trabalho padrões de protocolo como *X-10*; *Cebus*; *ZigBee*, *Z-Wave*, e etc.

3.2 SISTEMAS E ACESSIBILIDADE.

No artigo de Hashizume *et al.* (2018), é apresentado a *Telewheelchair*, uma cadeira de rodas elétrica equipada com uma função de controle remoto e assistência com funções computacionais. O cuidador pode controlar remotamente a cadeira de rodas por meio de um visor colocado na cabeça, como um capacete. Este visor possui um *display* óptico à frente de um dos dois olhos. Além disso, a cadeira de rodas possui um sistema que evita colisões automaticamente. Para a criação dessa cadeira foi aplicado um estudo em quatro tipos de sistemas e o tempo gasto para a realização dessas tarefas. O trabalho teve como objetivo uma melhora na mobilidade dos usuários, através da combinação de inteligência do usuário e inteligência da máquina.

Os autores criam vários cenários, em um desses cenários a cadeira é totalmente automática, em outro nem todas as funcionalidades precisam ser cobertas, só funções específicas do usuário devem ser automáticas. Por último, o cenário onde o foco é facilitar o uso da cadeira na área da saúde para profissionais de enfermagem.

Os autores deixam claro que cadeiras de rodas são essenciais para idosos e usuários com dificuldades de locomoção. A intenção é reduzir a carga dos cuidadores e trazer independência aos usuários. Usam o reconhecimento do ambiente de forma automática, para que o usuário não precise ser supervisionado constantemente. É deixado claro que as funções tecnológicas da cadeira não são de entendimento simples. Portanto, há a proposta de combinar a operação humana com a operação automática usando o que eles chamam de Inteligência Artificial. Este realiza reconhecimento de objetos e ajuda os usuários a usar a cadeira de rodas.

No artigo de Asai *et al.* (2016), foi abordado a criação do *ExtendedHand*, que usa um sistema com técnica de realidade espacial aumentada, onde a mão é um componente. Esse sistema foca no bem-estar do usuário cadeirante, já que sua comunicação é muitas vezes limitada, pois os alcances são estreitos e o movimento exacerbado da cadeira é cansativo. Os autores propõem um sistema de cadeira de rodas em que um projetor móvel alimentado por bateria é montado. O usuário manipula a mão virtual projetada como uma extensão da sua própria mão, usando um painel de toque instalado em sua cadeira de rodas em um dos braços. O painel reconhece os gestos da mão do usuário, especificando que o reconhecimento de gestos seja independente de condições de luz ambiente.

Os autores reiteram a importância da cadeira de rodas na vida do usuário e de como ela exige espaço para movimentação, sendo esse espaço, às vezes, insuficiente para alcançar o alvo desejado com as mãos. Sabendo disso, ao montar o projetor na cadeira de rodas, reconhecem que existe um problema onde a região projetada a frente da cadeira é limitada e fixa, estudando e solucionando todos os possíveis problemas do usuário, sempre de acordo com a preocupação do conforto e segurança. Abordam todo o seu projeto por meio da IoT (Internet das Coisas). Na Figura 2 é mostrado o aparelhamento do *ExtendedHand*, onde se tem o espelho de *pan-tilt*, próprio para mover-se e espelhar a imagem da mão onde o usuário desejar, o projetor da imagem da mão e o *tablet* acoplado a cadeira de rodas para que o usuário possa escolher o que a mão vai fazer ou para qual lugar ela será apontada.

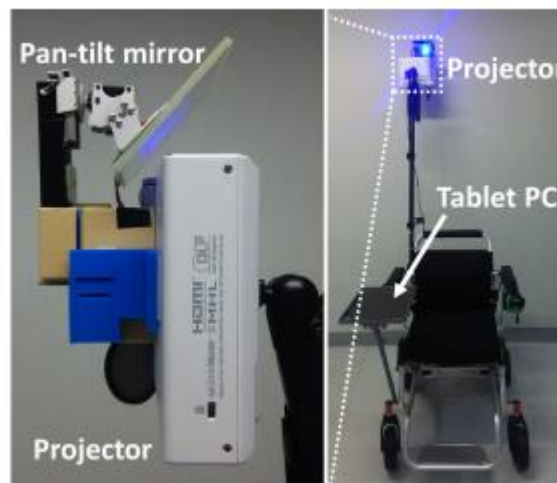


Figura 2 - Printscreen do modelo visual do sistema *ExtendedHand*.

FONTE: (Yuki Asai, *et al.*, 2016)

3.3 APLICATIVOS E ACESSIBILIDADE

Segundo Condé *et al.* (2014), no estudo de repositório de aplicativos móveis para pessoas idosas (tradução livre), o uso de tecnologia não é fácil para idosos. Os dispositivos e as tecnologias nem sempre estão adaptadas às suas necessidades, e em suma, são caras as que têm a adaptação. O contexto do projeto deixa claro carência de aplicativos específicos para pessoas idosas, com melhor usabilidade. A facilidade de uso e aprendizado do usuário, bem como, o desempenho do aplicativo, são fatores que precisam ser explorados. Após a análise estudos, os autores concluem que é possível perceber que os usuários precisam de mais tempo para

concluir as tarefas desejadas em seus dispositivos tendo problemas com: tamanho de tela, menu, interface, teclado virtual, funcionalidades como arrastar e soltar, dentre outros. Idealmente, aplicativos desse tipo carecem de lembretes para compromissos, informações pessoais de fácil acesso, menus padronizados e personalizados, recursos visuais como luz de fundo adaptável, texto grande e em negrito, esquema de cores para melhor visibilidade, grandes botões, etc.

A metodologia do estudo de Condé *et al.* (2014) foi iterativa, tendo vários ciclos com estágios, uma vez identificado o problema os estágios avançavam. Para seus resultados, foi utilizada revisão sistemática de literatura. Para o desenvolvimento tecnológico, foi utilizado o SCRUM, sendo esse um *framework* de desenvolvimento ágil que fornece ao usuário o processo, regras, artefatos necessários para aumentar a produtividade no desenvolvimento do aplicativo. Definido os critérios para pesquisa, foi feito questionários com idosos onde eles atribuíam notas de 1 a 10 para classificar as funcionalidades, levando em conta recursos relativos ao tato, auditivos e de memória. Portanto, considerado um projeto orientado a facilitar a vida de pessoas mais velhas, usando tecnologia e aplicativos móveis, um repositório, frisando que o usuário final deve ser sempre a prioridade no desenvolvimento de aplicativos, a adaptação e atenção as pessoas mais velhas nesse sentido são fundamentais.

Segundo Siebra *et al.* (2017), no artigo sobre acessibilidade em aplicativos e usabilidade para pessoas prejudicadas no contexto móvel, obteve sua observação e análise a partir de coleta de informações. A análise obteve voluntários de vários tipos de comprometimentos, como, por exemplo, o visual, o auditivo e o motor. Ao longo do experimento, os usuários foram observados e o processo foi registrado em vídeo. A pesquisa foi obtida com critérios de perfil de voluntário, uso dos dispositivos móveis e suas aplicações, observação do uso do voluntário com o seu *smartphone* e aplicativos pré-definidos e, por fim, uma entrevista após a interação.

No resultado do estudo de Siebra *et al.* (2017), foi apontado que usabilidade é um requisito não funcional fundamental para mapear os requisitos funcionais tangíveis. Como é focado em pessoas com dificuldades de locomoção, dentre essas diretrizes, o estudo aqui citado dividiu as deficiências em várias vertentes, de acordo com as iniciais. Foi concluído que para usuários, independentemente do tipo de deficiência, o aplicativo deve fornecer *layouts* adaptáveis sem perda de informação, com recursos de customização, sinalização de mudanças na interface com sons,

imagens e sensores tácteis (exemplo: vibração). Os autores concluíram que o aplicativo também deve orientar e fornecer *feedbacks* aos usuários sobre o que está acontecendo, fornecer interações por toque, oferecer instruções para uso, ter navegação linear e intuitiva, assistência para encontrar conteúdo, orientar usuários através do sistema, fornecer cobertura de erros e uso de sugestões sobre suas correções. Portanto, o artigo contempla informações sobre vários tipos de deficiência e as descreve de forma contextualizada, fornecendo diretrizes para análise de aplicativos para essas deficiências. O desenvolvimento de aplicativos móveis para esse grupo de usuários é complexo, sendo assim apenas diretrizes para ajuda de elicitação de requisitos para desenvolvimentos futuros.

3.4 APLICAÇÕES CORRELATAS

Nessa seção é apresentado uma breve descrição sobre aplicativos correlatos ao desenvolvimento deste trabalho.

Tabela 1 - Aplicativos correlatos

Aplicativo	Descrição
OpenHAB (https://community.openhab.org)	É uma plataforma de automação residencial de código aberto baseada em Java, que integra e combina uma vasta gama de diferentes sistemas de casa inteligente e tecnologias em uma única solução. Todos os dispositivos conectados neste aplicativo são possíveis de automação. Também contém várias regras diversificadas e abrangentes para o uso do usuário.
IFTTT (ifttt.com/discover)	Disponível para Android e iOS, o aplicativo oferece suporte para mais de 300 redes sociais, armazenamento em nuvem, música e outros. O usuário pode customizar ações de automação, tendo a opção de consultar regras criadas por outras pessoas, realizar ações pontuais e criar regras a partir de textos escritos pelo usuário.
eWeLink (http://ewelink.coolkit.cc/?p=1258)	E-WeLink é um <i>software</i> livre universal que transforma sua casa em um local inteligente, apenas com o uso de um controle remoto. Obtendo a opção de controle remoto de produtos de diferentes tipos e marcas de <i>hardware</i> . E-WeLink

	suporta dezenas de dispositivos inteligentes, incluindo tomada inteligente, lâmpada <i>LED</i> , ventilador, aquecedor, purificador de ar, etc. É um centro de controle inteligente que pode gerenciar todos os seus aparelhos através de um aplicativo
Android <i>Accessibility Suite</i> (https://play.google.com/store/apps/details?id=com.google.android.marvin.talkback&hl=pt_BR)	É uma coleção de serviços para acessibilidade em que os usuários utilizam seus dispositivos com auxílio de um interruptor, para ligar e desligar, sem necessidade do uso dos olhos. Inclui: <i>talkback</i> , acesso com interruptor, selecionar para ouvir em voz alta.
<i>Smart Life - Smart Living</i> (https://play.google.com/store/apps/details?id=com.tuya.smartlife&hl=pt_BR)	Aplicativo para transformar a casa em um local inteligente, controlando remotamente aparelhos domésticos de qualquer lugar, adicionando e monitorando vários aparelhos ao mesmo tempo, função de comando de voz, funcionamento com base em sensores e envio de alertas para garantia de segurança.

4 SISTEMA DE AUTOMAÇÃO

Neste capítulo é apresentado o projeto e a prototipação para a criação do aplicativo desenvolvido de acordo com a descoberta e a elicitación de requisitos, modelagem UML, prototipação de interfaces gráficas do usuário e prototipação com Arduino e Android.

4.1 DESCOBERTA E ELICITAÇÃO DE REQUISITOS

Os requisitos a seguir foram descobertos por meio da revisão de literatura dos trabalhos relacionados dispostos nesse documento, por análise de aplicativos existentes e cenários de uso. O aplicativo proposto deve atender aos seguintes requisitos:

- O aplicativo deve permitir o cadastro, a edição e a exclusão de usuários no banco de dados SQLite;
- O cadastro do usuário deve conter login e senha. Caso seja usado impressão digital, o cadastro deve conter uma impressão digital;
- O aplicativo deve permitir o cadastro, a exclusão e a edição de componentes, por fornecimento de listagem de dispositivos pareados por *bluetooth*;
- O aplicativo deve armazenar dados sobre o cadastro, a exclusão e a edição de componentes no banco de dados SQLite;
- O aplicativo disponibilizará a função *talkback*⁷, alerta vibratório, alerta luminoso e agendamento de tarefas;
- O aplicativo disponibilizará de recursos visuais claros, interface simples e informações claras para facilitação do seu uso.

Os requisitos e suas descrições estão descritos como requisitos de sistema no final deste documento (Apêndice A). O aplicativo foi documentado no padrão de modelo da IEEE⁸, com base nas adaptações propostas por Sommerville (2011).

O aplicativo desenvolvido neste trabalho, foi intitulado de AutBility, influenciado por automação e acessibilidade (*accessibility*). Contempla as funcionalidades base apresentadas por suas aplicações correlatas e complementa em sua funcionalidade

⁷ “adiciona feedback falado, por áudio e vibração ao seu dispositivo.”

(https://play.google.com/store/apps/details?id=com.google.android.marvin.talkback&hl=pt_BR)

⁸ <https://www.ieee.org/>

com alertas vibratório e luminoso, a interface clara e intuitiva, onde o usuário pode modificar alguns elementos da parte visual em consenso com seu conforto, como, por exemplo, mudar cores da interface do aplicativo e aumentar a fonte para melhor visibilidade.

4.2 PROJETO DO SISTEMA COM UML

A linguagem de modelagem unificada (ou *Unified Modeling Language - UML*) é considerada uma linguagem “unificadora” de notações que utiliza diferentes técnicas de modelagem. É uma linguagem visual que modela sistemas orientados a objetos, ou seja, define elementos gráficos que são utilizados em modelagem de sistemas, representando o paradigma da orientação a objetos em conceitos e representação, dispondo de elementos gráficos para construir diagramas que reproduzem as visões de um sistema (BEZERRA, 2015).

A UML pode ser usada em qualquer modelagem, não importando a linguagem de programação e processo de desenvolvimento do sistema podendo ser utilizada em qualquer sistema de *software* com diversas abordagens de desenvolvimento. É extensiva, então ela é adaptável às características específicas de qualquer projeto de desenvolvimento.

Segundo Viegas (2009), “UML permite que você ‘desenhe’ uma ‘planta’ do seu sistema.” a UML disponibiliza dos diagramas de classes, casos de uso, objetos, sequencia, colaboração, estado, atividade e componente. Independente de ferramenta de desenvolvimento, fornece visão lógica, facilitando implementação física.

“...é uma linguagem padrão para visualização, especificação, construção e documentação de um aplicativo ou projeto de software, e objetiva aumentar a produtividade, otimizar as etapas que envolvem o desenvolvimento de um sistema, aumentando assim a qualidade do produto a ser implementado.” (VIEGAS, 2009).

Neste capítulo serão abordados e descritos os diagramas de caso de uso, pacotes, classes, sequencia, atividades, estados e implantação.

4.2.1 DIAGRAMA DE CASO DE USO

Os diagramas de caso de uso foram usados no levantamento de requisitos, sendo assim esclarecidos nos cenários de uso. Na Figura 3 ilustra o caso de uso completo do aplicativo AutBility.

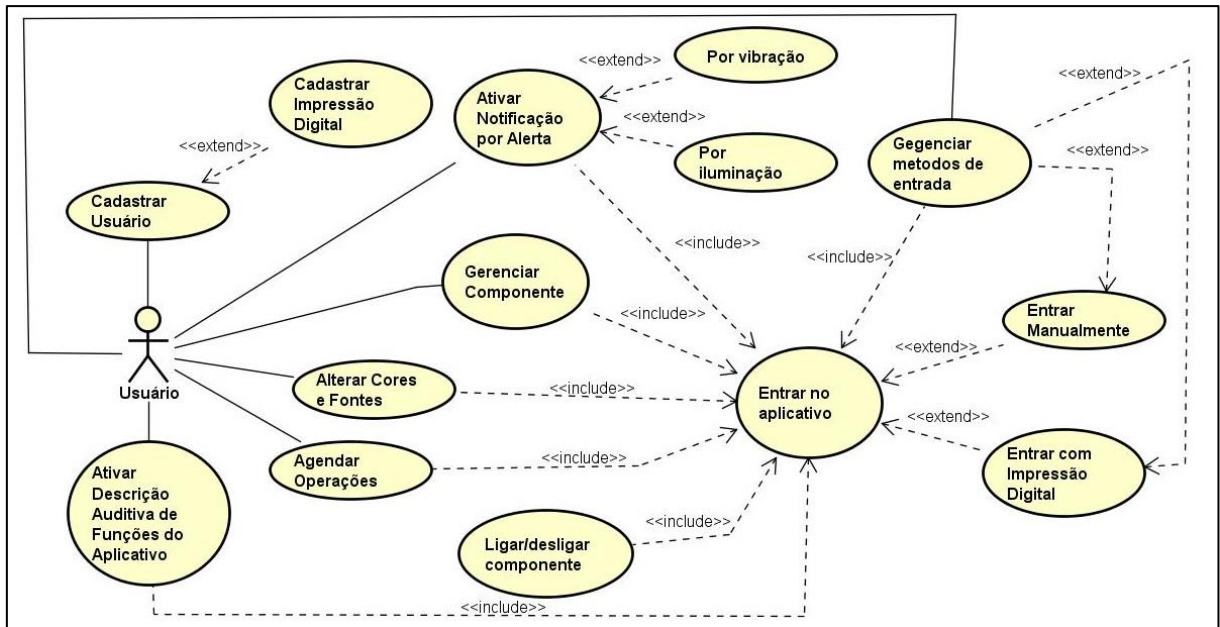


Figura 3 - Diagrama de caso de uso completo do aplicativo

FONTE: (AUTORIA PRÓPRIA).

No cenário de Cadastrar Usuário o aplicativo deve inicializar a tela onde será efetuado o cadastro, onde poderá ser utilizado um *login* e senha da escolha do usuário, ou a opção Cadastrar Impressão Digital, onde o aparelho que tenha o sensor de escaneamento de digital, escaneará a digital e ela será armazenada no banco de dados do aplicativo.

Alguns cenários após o cenário de cadastramento exigem autenticação prévia no aplicativo usando os dados cadastrados anteriormente, essa descrição se encaixa no cenário de Entrar no Aplicativo onde todos os outros cenários, menos os de métodos de entrada, são relacionamento de inclusão com ele. Assim, sempre que qualquer caso incluso nele for executado, ele será executado primeiramente, todos os cenários inclusos ao de autenticação precisam executa-lo primeiramente. Neste cenário as opções: Manualmente (*login* e senha) e Impressão digital são estendidas a ele, pois esses casos podem ser executados, alternativamente, para logar no aplicativo. No cenário Gerenciar Impressão Digital o usuário poderá excluir, editar ou

inserir. Portanto, nesta ordem, remover a impressão já cadastrada no banco de dados, modificar uma já existente no banco de dados por outra de sua preferência ou inserir caso não possua nenhuma digital cadastrada.

No cenário Gerenciar Componente serão disponibilizadas ao usuário as opções de Cadastro, Exclusão e Edição de componentes, que podem ser televisão, ar condicionado, ventilador. Com essas funções será possível inserir um novo componente, retirar componente salvo, modificar informações de componente salvo, essas informações, contidas no banco de dados do aplicativo. Todas estas opções estão dispostas de forma clara e intuitiva no menu principal do aplicativo, restando ao usuário apenas a escolha da opção desejada. O cenário Ativar Notificação por Alerta será disponibilizado opções de Cadastro e Edição. O usuário pode definir um alerta que está estendido a luminoso ou vibratório. Caso haja atividade no componente desejado, por exemplo, quando um sensor de proximidade fosse ativado com uma pessoa se aproximando dele, o usuário receberia um dos alertas escolhidos. Essa escolha pode ser feita no menu Configurações.

No cenário Ligar/Desligar componente, o usuário após o cadastro do componente poderá, no Menu Principal, selecionar a opção Lista de Componente, onde será catalogado todos os componentes salvos pelo usuário. Será exibido a opção Ligar/Desligar, quando o usuário selecionar o componente desejado.

No cenário Alterar Cores e Fontes, o usuário no menu Configurações, pode escolher por uma fonte maior ou menor, por cores que sejam confortáveis a sua visão. Cores de fundo do aplicativo, dos blocos de opções e das fontes. O aplicativo disponibilizará ao usuário, tabelas de cores diversificadas que estejam dentro das configurações suportadas do aparelho. O mesmo vale para fontes, que serão fontes disponíveis pelo aparelho usado pelo usuário.

No cenário Agendar Operações, o usuário poderá agendar, pelo Menu Principal, na opção agendamento, a quantidade de dias e horas que ele deseja que o componente escolhido permaneça ligado ou desligado automaticamente, sem necessidade de interação do usuário sempre que precisar utiliza-lo, caso a usabilidade seja corriqueira.

No cenário Ativar Descrição Auditiva de Funções do Aplicativo, o usuário poderá ativar a função *TalkBack* inclusa no seu aparelho, onde poderá ouvir toda a

descrição e funcionalidade do que está sendo tocado na tela, facilitando assim a usabilidade para pessoas que tenham dificuldades relacionadas a visão.

4.2.2 DIAGRAMA DE PACOTES

Segundo Larman (2007), diagrama de pacotes é frequentemente usado para ilustrar a arquitetura lógica do sistema, incluindo nelas: camadas, subsistemas, pacotes, etc. Fornece uma forma de agrupar elementos, podendo agrupar por exemplo classes, outros pacotes, casos de uso, etc. Pode mostrar dependência (acoplamento) entre pacotes, exibindo o acoplamento, em larga escala, do sistema aos desenvolvedores. As linhas de dependência (setas tracejadas) têm esse uso, onde a seta aponta para o pacote dependente.

Na Figura 4, é ilustrado os quatro pacotes que foram usados, que são Modelo, Visão, Controle e Banco de dados. Estes pacotes seguem o modelo padrão de desenvolvimento Modelo-Visão-Controle (MVC) que tem uma usabilidade ampla na programação. Visa assim separar a lógica de negócio da interface com o usuário, assim os programadores podem ter a facilidade de obter modificações nas partes necessárias, sem que influencie nas outras partes. Segundo Durelli *et al.* (2008) *Model* engloba informações sobre domínio e objetos que são funcionalidades principais do sistema, em *View* representa-se a interface gráfica do usuário, em *Controller* é definido como *View* se comporta e se comunica, de acordo com informações inseridas pelo usuário, atualizando a camada *Model*, sendo de certa forma um canal de comunicação entre essas duas camadas.

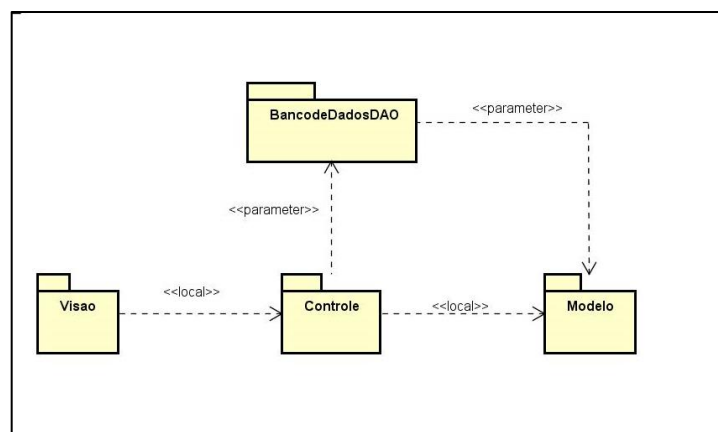


Figura 4 - Diagrama de pacotes completo do aplicativo.

FONTE: (AUTORIA PRÓPRIA).

No padrão MVC, o modelo representa dados do aplicativo. Visão representa os elementos de interface do usuário, as telas de interação do aplicativo, textos e formulários. Controle gerencia a comunicação entre o modelo e a visão. O Banco de dados é onde será armazenado os dados da integração entre controle, modelo e visão. Também foi usado o padrão *Data Access Object* (DAO), que segundo Trindade e Fischer (2015) “[...] pode ser aplicado nas aplicações que façam acesso a qualquer tipo de banco de dados [...]” “[...] e que possa ter que acessar outros tipos quaisquer durante ou após o desenvolvimento da aplicação. ”

4.2.3 DIAGRAMA DE CLASSES

Segundo Bezerra (2015) o diagrama de classes é utilizado em relação a construção de modelos de classe, desde o nível de análise até o nível de especificação. Sendo rico em termos de notação, as classes são representadas por caixas, com o primeiro compartimento sendo o nome; o segundo seus atributos (estes correspondem a informações que um objeto armazena); o terceiro são as operações, que correspondem às ações que um objeto pode realizar.

Na avaliação dos casos de uso, foram criadas as classes, as colaborações e as estruturas internas do aplicativo, que influenciam no funcionamento com o usuário e interno de acordo com a elicitação de requisitos feita previamente. Na Figura 5 é ilustrado as classes do pacote Modelo que foram Usuario, Digital, Agendamento e Aparelho, que tem estereotipo *entity*, pois são entidades onde apenas armazenam dados.

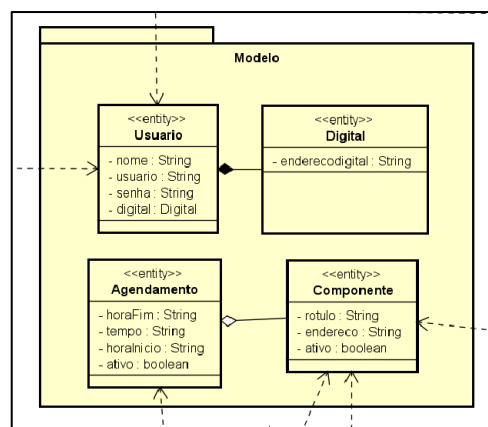


Figura 5 - Pacote Modelo

FONTE: (AUTORIA PRÓPRIA).

Na Figura 6 é ilustrado o pacote Controle, que se encontram as classes Alerta, que tem como sub-classes Vibracao, Iluminacao e DescricaoAuditiva; AlterarInterface, que tem métodos para modificar a interface aplicativo; OperacaoAgendamento que tem seus métodos para o agendamento de componentes; ControleUsuario e ControleComponente.

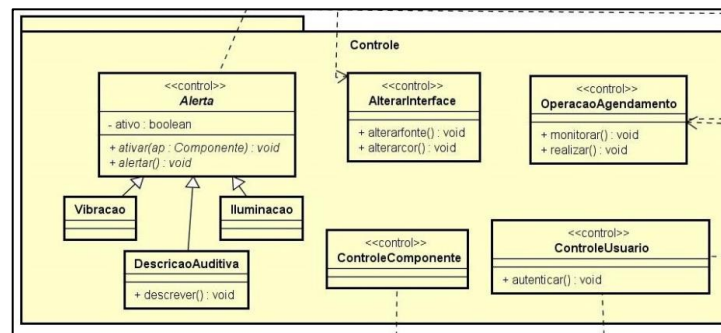


Figura 6 - Pacote Controle

FONTE: (AUTORIA PRÓPRIA).

Na Figura 7 ilustra o pacote BancodeDadosDAO que tem as classes: Conexao, BancoUsuario, BancoComponente e BancoAgendamentos.

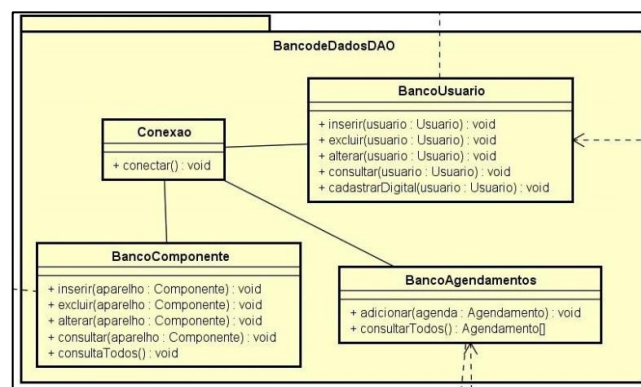


Figura 7 - Pacote BancodeDadosDAO

FONTE: (AUTORIA PRÓPRIA).

Na Figura 8 é ilustrado o pacote Visão que tem as classes: FormCadastUsuario, FormCadastDigital, FormCadastComp, FormCadastAgendamento, EdicaoInterface, InterfaceAjuda, InterfaceSobre e InterfaceAcessibilidade. A classe Usuario gera dependência com BancoUsuario e

ControleUsuario, armazenando e controlando neles suas funcionalidades, que tem relacionamento de composição com a classe Digital.

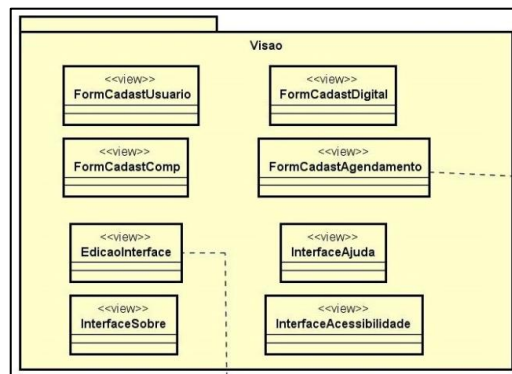


Figura 8 - Pacote Visao

FONTE: (AUTORIA PRÓPRIA).

A classe EdicaoInterface depende de AlterarInterface, pois esta controlará suas mudanças na interface. É usado a classe Conexao com associação entre as classes do pacote BancodeDadosDAO para haver relacionamento entre elas na execução do aplicativo.

A classe Agendamento gera dependência com BancoAgendamento, local de armazenamento de dados de agendamentos, gerando dependência também com o controle OperacaoAgendamento, local onde controla as suas funcionalidades. A visão FormCadastAgendamento, interface de aplicativo, gera agregação com Componente estando como parte dele. Componente tem dependência com BancoComponente, armazenamento de dados de componentes, depende de ControleComponente e Alerta, onde são controladas todas as funcionalidades.

A Figura 9 ilustra a visão geral do diagrama de classes e seus relacionamentos, nas setas de dependência (setas tracejadas), que segundo Bezerra (2015) implica que uma classe depende de serviços que são disponibilizados por outra. Portanto, também pode-se notar as dependências com descrições do tipo local ou *parameter*, que significam segundo Bezerra (2015) que no caso da variável local alguma operação da classe, onde inicia-se seta, tem alguma ação que sua efetivação precisa de uma variável local da classe que se finaliza a seta. E o autor também explica que no caso da variável *parameter* alguma operação da classe, onde inicia-se a seta, possui um parâmetro da classe que se finaliza a seta.

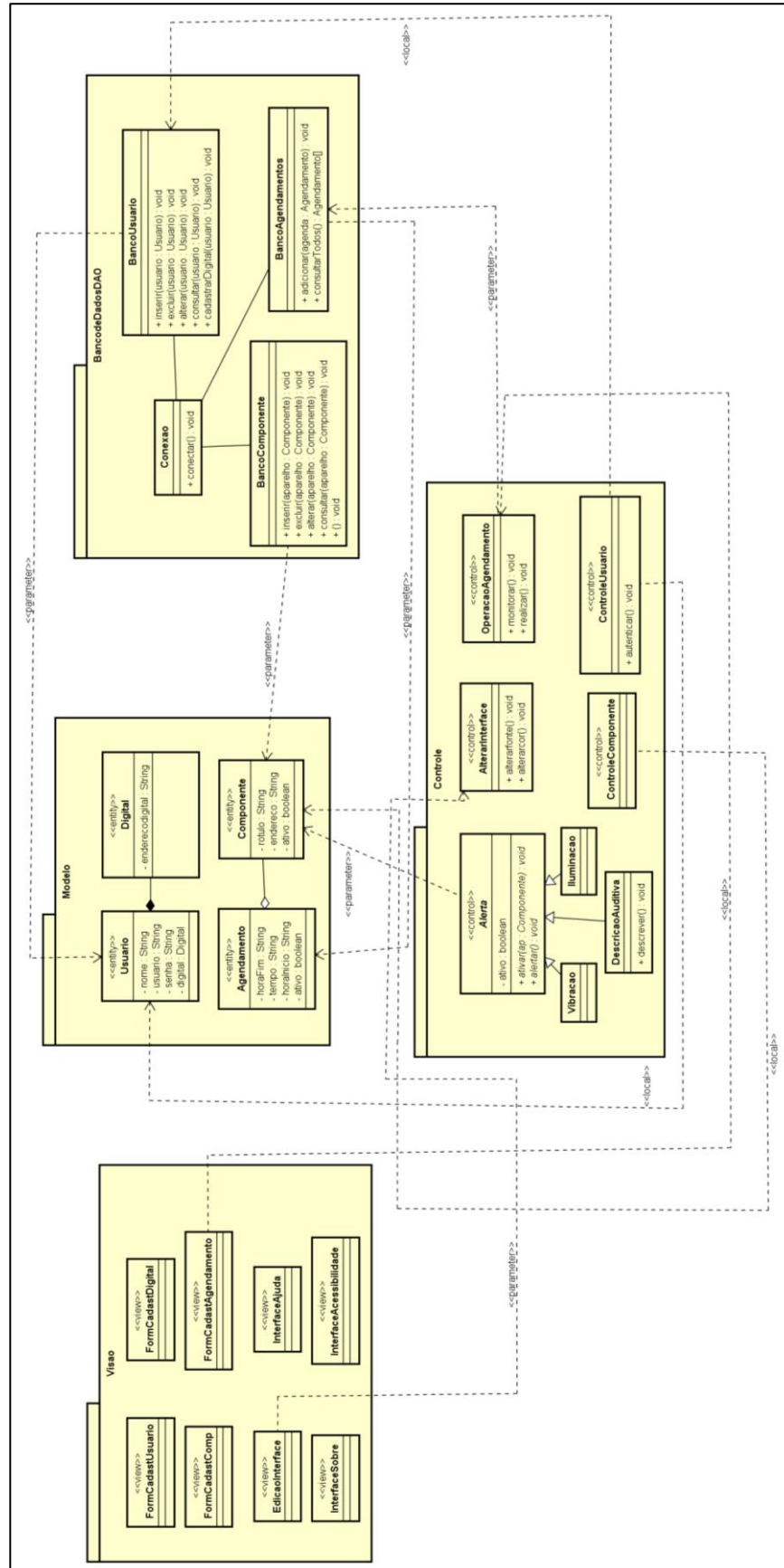


Figura 9 - Diagrama de classes

FONTE: (AUTORIA PRÓPRIA).

4.2.4 DIAGRAMA DE OBJETOS

Para ilustrar como ocorre o relacionamento entre algumas das classes definidas, na Figura 10 é ilustrado o diagrama de objetos da operação cadastro de componente. É iniciado com o controle ControleUsuario, que tem como atributo autenticar o usuário para que ele possa usar todas as funcionalidades do aplicativo. Logo após, a interface de cadastro de componente é acionada pelo usuário em FormCadastComp, onde serão inseridos os atributos da classe Componente, que são o rotulo, endereço e estado atual. Finalizadas as inserções, será acionado o controle ControleComponente que se associa com o banco de dados BancoComponente, inserindo estes dados nele, que se associa com um novo Componente cadastrado tendo seu rotulo, endereço e estado atual.

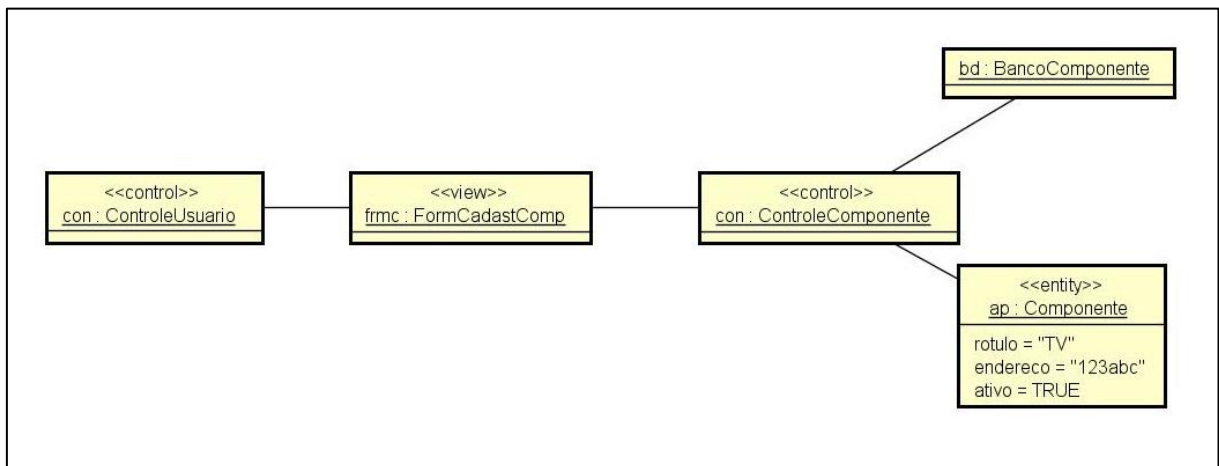


Figura 10 - Diagrama de objetos do caso de uso Cadastro de Componente

FONTE: (AUTORIA PRÓPRIA).

Na Figura 11, o diagrama de objetos da operação agendamento de componentes é ilustrado. A operação é iniciada com o controle ControleUsuario, que tem como atributo autenticar o usuário para que ele possa usar todas as funcionalidades do aplicativo. Posteriormente, a interface do cadastro de agendamento é acionada em FormCadastAgendamento, onde será iniciado o BancoComponente para listagem de todos os dispositivos salvos, o usuário escolherá um componente, neste exemplo o Ar Condicionado, onde já virá com endereço e estado atual gravados nele. Logo após a operação Agendamento é iniciada, onde será inserido o horário de início, o horário de término e o tempo em que este componente ficará ativo. Posteriormente, esses dados serão inseridos no banco de dados

BancoAgendamento, que iniciará o controle operação Agendamento, realizando o agendamento e monitorando seu funcionamento, sendo esses seus métodos de classe.

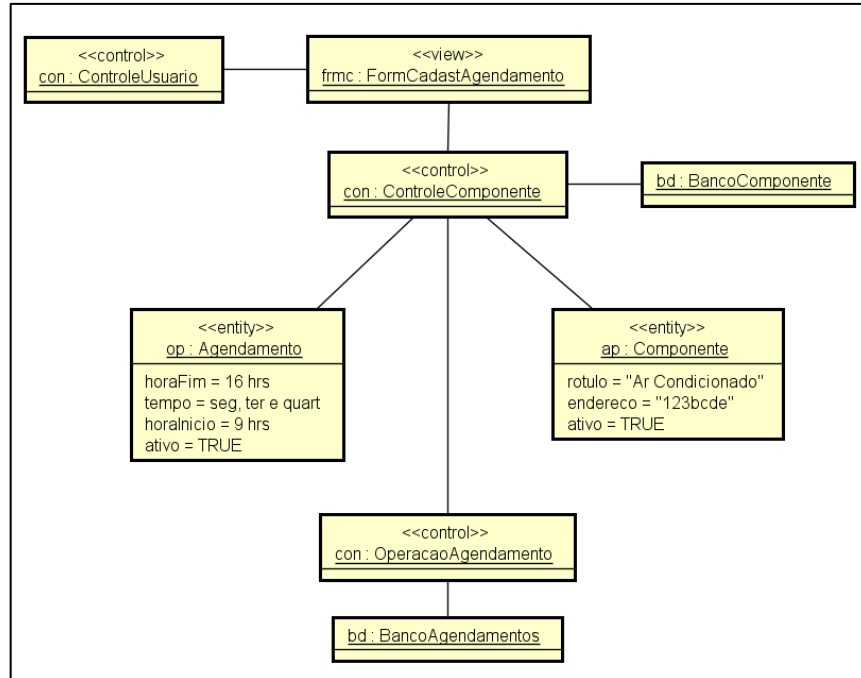


Figura 11 - Diagrama de objetos do caso de uso Agendamento de Componente.

FONTE: (AUTORIA PRÓPRIA).

4.2.5 DIAGRAMA DE SEQUÊNCIA

Segundo Bezerra (2015), um diagrama de sequência expõe as interações em ordem temporal aos acontecimentos, dispondo de recursos gráficos para isto.

Os diagramas das Figuras 12 e 13 foram elaborados de acordo com algumas partes do caso de uso e diagrama de classes. Mais especificamente, o cadastro de componente e o agendamento de tarefas para um componente.

- Cenário – Cadastro de Componente

Na Figura 12, o diagrama de sequência da atividade cadastro de componente é ilustrado. As classes são FormCadastComp, ControleComponente, Componente e BancoComponente. Estes objetos são fundamentais na comunicação do usuário (representado por um ator), aplicativo e banco de dados, que armazenará os dados necessários.

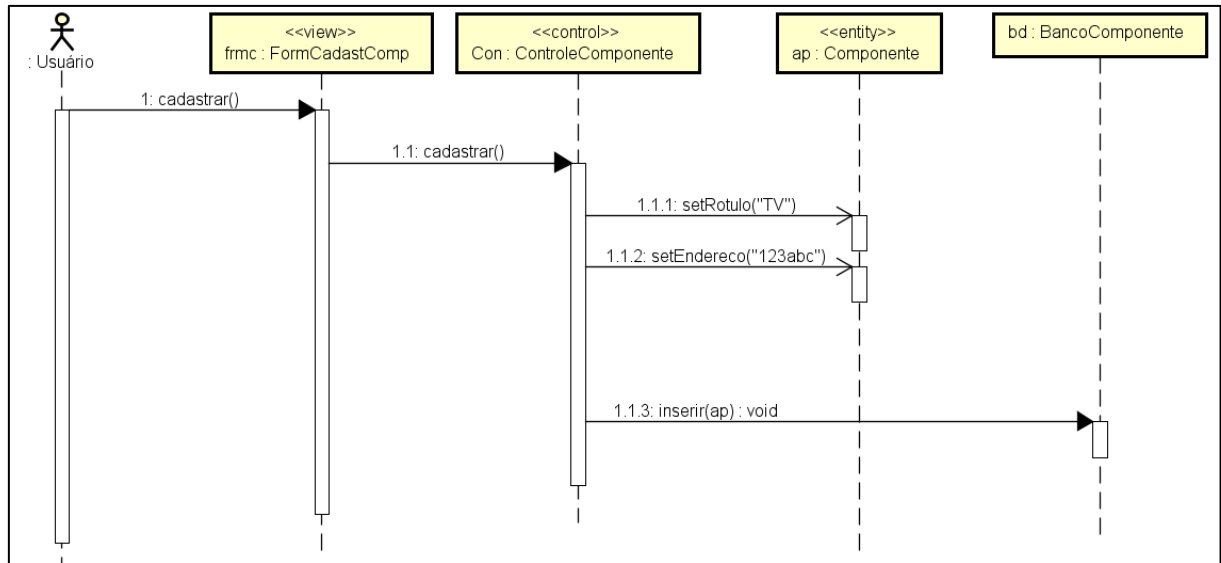


Figura 12 - Diagrama de sequência do caso de uso Cadastro de Componente.

FONTE: (AUTORIA PRÓPRIA)

A sequência do cadastro de componentes inicia-se com o ator Usuário requisitando uma ação síncrona, cadastrar(), ao objeto da classe de visão FormCadastComp, que requisita uma ação síncrona, cadastrar(), ao objeto da classe controle, ControleComponente, que chama o método setRotulo("TV") e setendereco("123abc") ao objeto de classe entidade Componente, usando Televisão como exemplo, que realiza uma ação síncrona, inserir(ap), que envia o método de armazenamento no banco de dados do componente, através da comunicação com o objeto da classe BancoComponente.

- Cenário – Agendamento de Componentes

Na Figura 13 é descrito o diagrama de sequência da operação Agendamento de Componentes, o ator é o Usuário e as classes são FormCadastAgendamento, ControleComponente, BancoComponente, OperacaoAgendamento, Agendamento e BancoAgendamento.

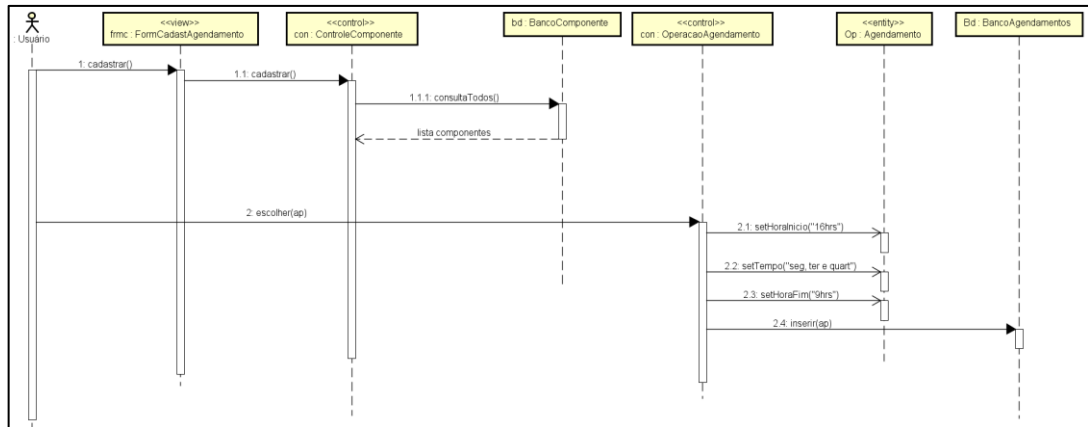


Figura 13 - Diagrama de sequência do caso de uso Agendamento de Componente.

FONTE: (AUTORIA PRÓPRIA).

A sequência para o agendamento inicia com o ator Usuário requisitando uma ação síncrona, cadastrar(), ao objeto da classe de visão FormCadastrAgendamento, que requisita outra ação síncrona, cadastrar(), ao objeto da classe de controle ControleComponente, realizando a ação síncrona, ConsultaTodos(), ao objeto da classe BancoComponente, listando todos os aparelhos e enviando de volta a mensagem, listar componentes, ao ControleComponente, realiza uma ação síncrona, escolher(ap), usando Ar Condicionado como exemplo, ao objeto da classe controle OperacaoAgendamento, que realizará a ação assíncrona que chama o método, setHorainicio("16hrs"), logo após, setTempo("seg, ter e quart") e, por fim, setHoraFim(9hrs). Concluídos esses métodos, será realizada a ação assíncrona, inserir(ap), ao objeto de classe banco de dados BancoAgendamento.

4.2.6 DIAGRAMA DE ATIVIDADE

Diagrama de atividade é definido por Bezerra (2015) como sendo “[...] um tipo de especial de diagrama de estados, em que são representados os estados de uma atividade em vez dos estados de um objeto [...]” “[...] diagramas de atividade são orientados a fluxos de controle”.

Nesta seção serão apresentados os diagramas de atividade do aplicativo nas funções Cadastro de Componente e Agendamento de Componente.

- Cadastro de Componente

O diagrama de atividade, Cadastro de Componente, é ilustrado na Figura 14. A ação inicial é precedida pela ação escolher acesso, onde o usuário vai escolher o método de entrada, uma estrutura de ramificação seleciona a escolha do usuário. Caso a escolha seja digital, a ação procedida será solicitar a digital, logo após a ação validar digital é acionada. Caso a escolha seja manualmente, a ação procedida será solicita usuário/senha e ação após será válida usuário/senha. Uma estrutura de ramificação mostra que em qualquer das duas escolhas, se validados o usuário, seguirá com a ação escolher a opção cadastro, e assim a ação lista todos os componentes ativos, por *bluetooth*. A próxima ação seleciona um ou mais componentes para cadastrar, e a ação posterior é confirmar cadastro, depois a ação registra na base de dados, onde os dados serão inseridos no banco de dados de componentes, encerrando no estado final.

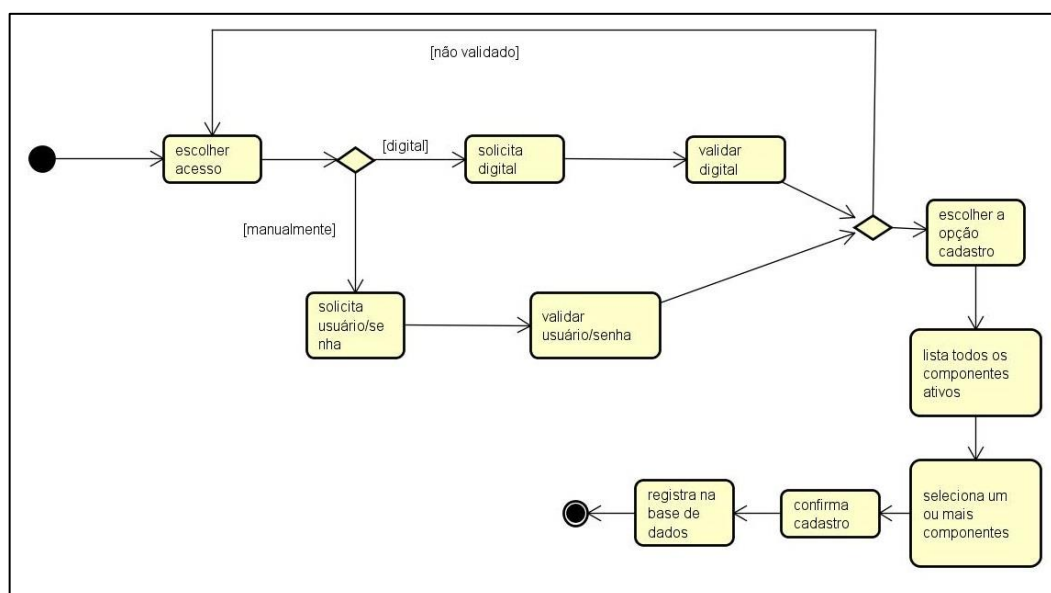


Figura 14 - Diagrama de atividade do caso de uso Cadastro de Componente.

FONTE: (AUTORIA PRÓPRIA).

- Agendamento de Componente

A Figura 15 ilustra o diagrama de atividade, Agendamento de Componente. A ação inicial é precedida pela ação escolher acesso, onde o usuário vai escolher o método de entrada, uma estrutura de ramificação seleciona a escolha do usuário. Caso a escolha seja digital, a ação procedida será solicitar a digital, logo após a ação valida digital é acionada. Caso a escolha seja manualmente, a ação procedida será

solicita usuário/senha e ação após será validar usuário/senha. Uma estrutura de ramificação mostra que em qualquer das duas escolhas, se validados o usuário, seguirá com a ação escolher opção agendamento, seguirá a ação acessar banco de dados de componentes salvos, em seguida a ação escolher componente para agendamento, depois a ação cadastra agendamento, confirma cadastro. Por fim, a ação registra no banco de dados de agendamento, encerrando no estado final.

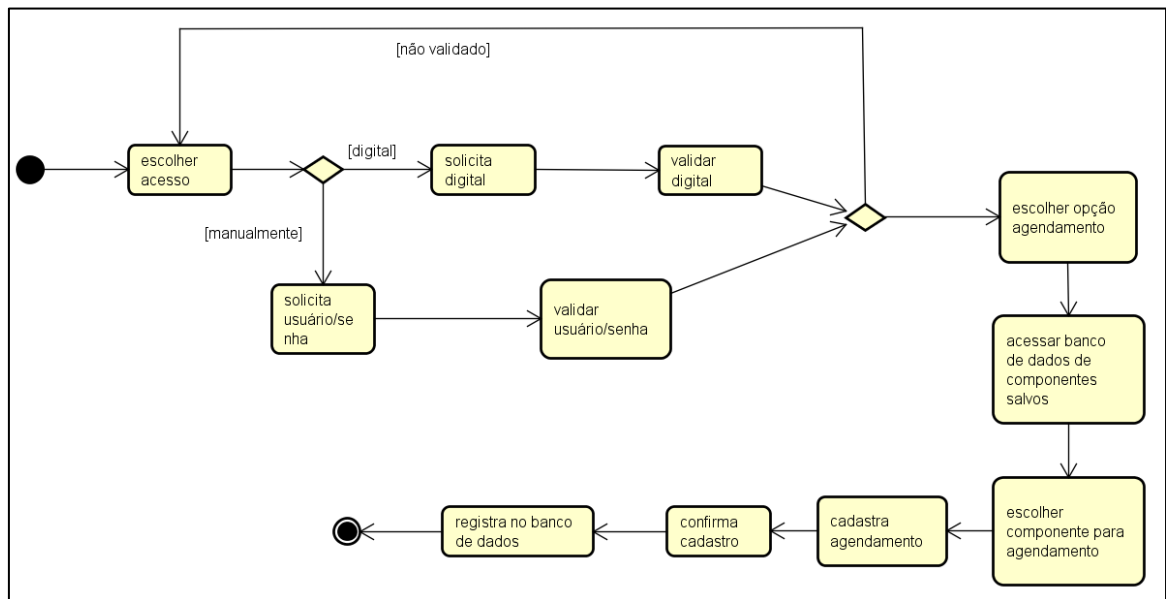


Figura 15 - Diagrama de atividade do caso de uso Agendamento de Componente.

FONTE: (AUTORIA PRÓPRIA).

4.2.7 DIAGRAMA DE ESTADO

Diagrama de estado é um diagrama que tem como objetivo descrever o comportamento, através de estados e transições que podem ser atribuídas durante um certo período de tempo, que podem ser um sistema, componente ou objeto (REVISTABW, 2013).

Nesta seção será abordado o diagrama de estado relacionado com Agendamento de Componente, pois a mesma engloba um bom cenário de uso. A Figura 16 ilustra o diagrama de estado do Agendamento de Componente.

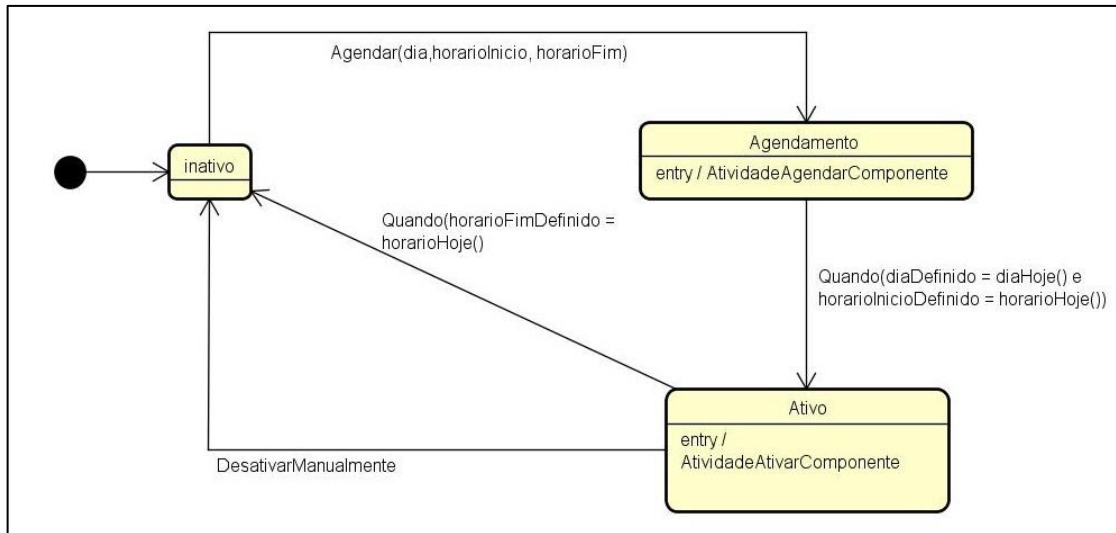


Figura 16 - Diagrama de estado da classe Agendamento de Componente.

FONTE: (AUTORIA PRÓPRIA).

A ação de início é no estado inativo, que tem como transição para disparar Agendar (dia, horarioInicio, horarioFim), no estado Agendamento. Estado esse que tem como cláusula entry especificando a ação AtividadeAgendarComponente e que tem como transição disparar Quando(diaDefinido = diaHoje() e horarioInicioDefinido = horarioHoje()), no estado Ativo. Estado esse que tem como cláusula entry, especificando a ação AtividadeAtivarComponente, está quando finalizada dispara a transição Quando(horarioFimDefinido = horarioHoje()). Isso significa que, quando a atividade de ativar o componente finalizar, o componente irá para o estado inativo. No caso da transição DesativarManualmente, quando o usuário escolhe desligar o agendamento de forma manual, o componente também irá para o estado inativo.

4.2.8 DIAGRAMA DE IMPLANTAÇÃO

Segundo Silva (2010), um diagrama de implantação tem como objetivo determinar os mecanismos em que o sistema deve ser executado. Portanto, podem ser hardware necessário, características físicas como servidores, estações topologia e protocolos de comunicação. Na Figura 17 é ilustrado o diagrama de implantação do aplicativo AutBility.

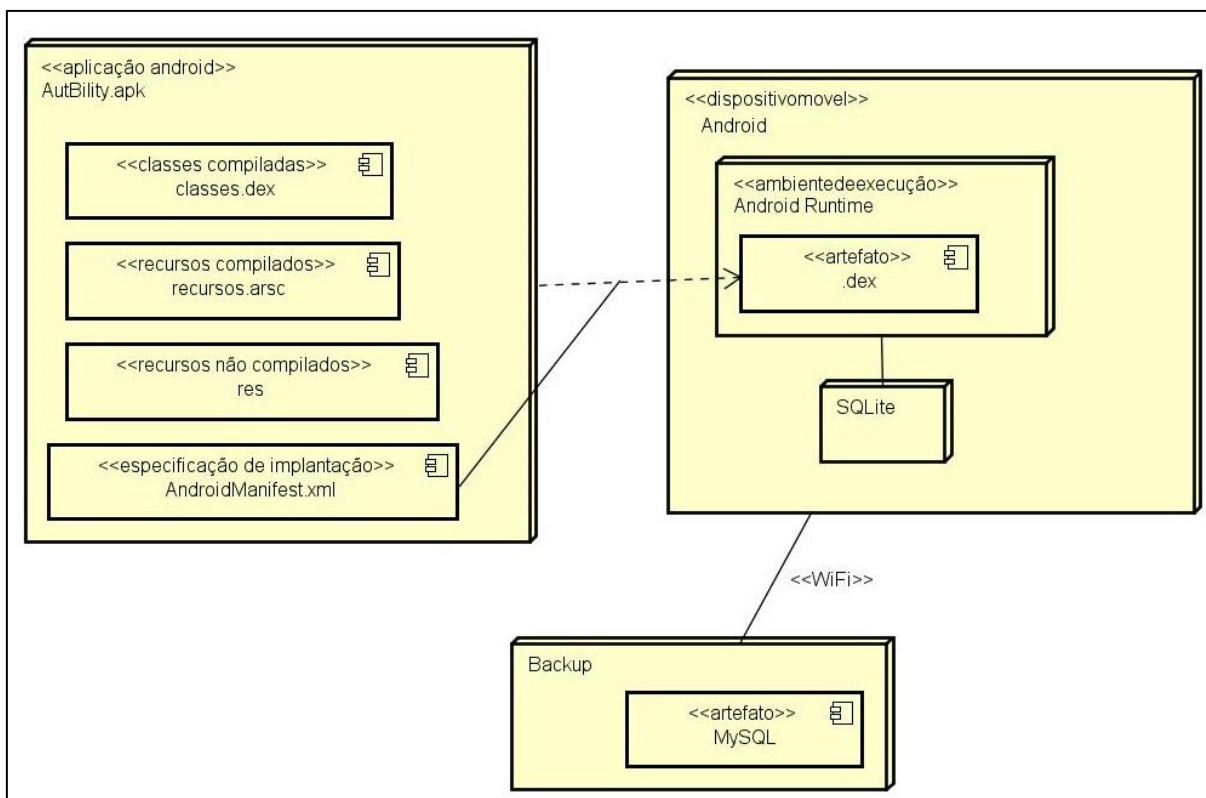


Figura 17 - Diagrama de implantação do aplicativo.

FONTE: (AUTORIA PRÓPRIA).

4.3 INTERFACES GRÁFICAS DO USUÁRIO

Diante do que foi avaliado e elicitado, foi desenvolvido um protótipo de interface gráfica do usuário, do sistema de automação para o auxílio de pessoas idosas e com dificuldades de locomoção. O nome é dado ao aplicativo: AutBility, com enfoque na automação com acessibilidade (*accessibility* em inglês). As interfaces gráficas do usuário. Dispostas nesta seção foram criadas com auxílio da ferramenta Marvel⁹.

É representado o modelo do aplicativo, sendo desenvolvido no presente trabalho um protótipo para testes. Na Figura 18a é exibida a tela inicial do aplicativo com as opções de usuário e senha, ou opção de inserir a impressão digital, um atalho para o cadastro de novos usuários.

Na Figura 18b é ilustrado o menu principal após o acesso ao usuário ser concedido na tela inicial, com as opções para o usuário em caixas bem distribuídas e grandes, correlacionando o conforto ao toque. As seleções são: inserir, editar, listar e

⁹ Marvel App. Disponível em: <<https://marvelapp.com/41fha77>>. Acesso em: 06 Ago 2018.

excluir um componente, configurações, ajuda e sobre (dados do desenvolvedor e versão do aplicativo).



Figura 18 - a) Interface gráfica do usuário inicial do aplicativo. b) Interface gráfica do usuário Menu Principal

FONTE: (AUTORIA PRÓPRIA).

Na Figura 19a é ilustrado o menu de atalho quando se está na tela de alguma das opções oferecidas, caso o usuário queira acessar rapidamente outra função.

Na Figura 19b, é exibida a tela da opção inserir componente, após ser escolhida no menu principal, que tem uma lista anexa de dispositivos conectados ao Arduino e distribuídos via *bluetooth*, contendo o aviso de “ligar o *bluetooth*”. Caso o usuário escolha um ou mais, ele tocará no botão cadastrar e logo o dispositivo será adicionado à lista de dispositivos salvos. Também é disponibilizada a opção de notificar caso tenha atividade. Isso serve para quando os dispositivos forem acionados ou o sensor de proximidade apresentar alguma variação, o usuário receberá um alerta vibratório ou luminoso dependendo de sua escolha nas configurações.

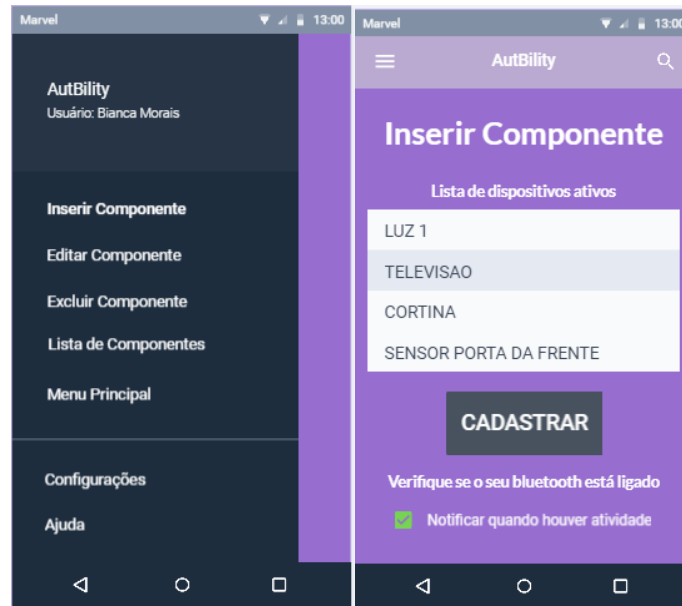


Figura 19 – a) Menu de navegação da interface gráfica do usuário. b) Interface gráfica do usuário para inserir componente.

FONTE: (AUTORIA PRÓPRIA)

A Figura 20a ilustra a lista de dispositivos salvos advindo da opção inserir. Nesta tela é disponibilizado um botão de ligar e desligar para o usuário fazer a opção desejada no dispositivo escolhido pela lista. A Figura 20b ilustra de forma semelhante a opção anterior. Nesse caso, o usuário vai ter a lista de dispositivos salvos e a opção de excluí-los.

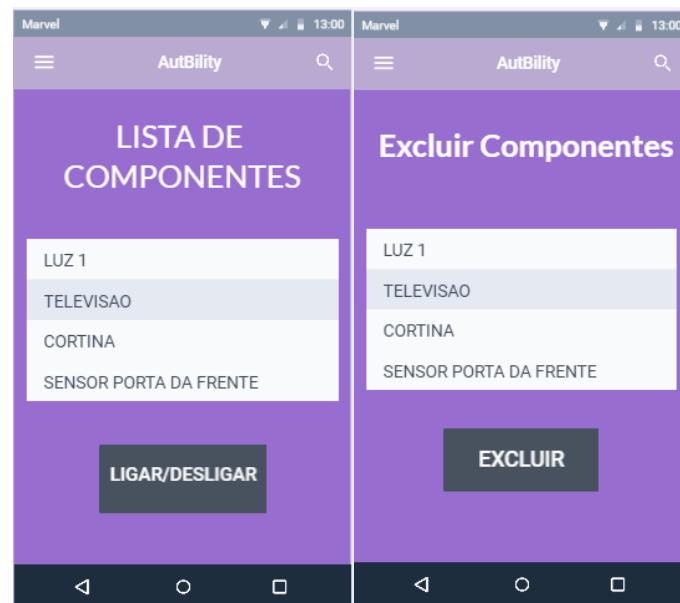


Figura 20 - a) Interface gráfica do usuário com a lista de componentes salvos. b) Interface gráfica do usuário para exclusão de componentes

FONTE: (AUTORIA PRÓPRIA)

Na Figura 21a é ilustrado a tela onde o usuário pode editar os dispositivos salvos, modificando alguma informação desejada. Caso a modificação tenha sido completada o usuário vai ter novamente a opção de notificação, caso haja atividade. Na Figura 21b é ilustrado o menu de configurações, onde tem as opções de cadastro, exclusão e edição de usuário e senha e/ou impressão digital, ativar *talkback*, edição de opções de funcionalidades (mudar cores do aplicativo, tamanho de fonte, tipos de alerta) e sair do aplicativo, caso o usuário queira finalizar seu acesso.

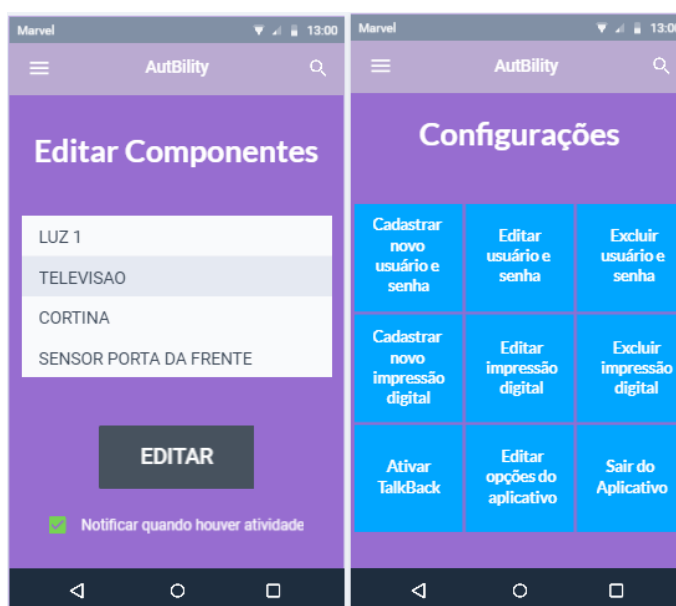


Figura 21 – a) Interface gráfica do usuário para inserção de novos componentes. b) Interface gráfica do usuário do menu de configurações.

FONTE: (AUTORIA PRÓPRIA).

4.4 PROTÓTIPO DESENVOLVIDO

O protótipo desenvolvido no presente trabalho foi criado na plataforma *Android Studio*¹⁰. O AutBility foi criado com base em instruções de ensino e código disponibilizado na plataforma GitHub¹¹, pelo criador Justin Bauer¹². O código do aplicativo foi adaptado em algumas partes para as necessidades do presente trabalho e para melhor entendimento de leitores que quiserem ter acesso.

¹⁰ <https://developer.android.com/studio/>

¹¹ <https://github.com/bauerjj/Android-Simple-Bluetooth-Example>

¹² <http://mcuhq.com/27/simple-android-bluetooth-application-with-arduino-example>

O AutBility funciona a partir da plataforma Android 6, o requisito para que ele seja usado é apenas ter *bluetooth*. Seu funcionamento não necessita de “altas” “configurações, suas funções de protótipo são simple.

Foi usado o Arduino Mega com o módulo *bluetooth* HC-05. Nas Listagens 1 e 2 estão ilustrados os códigos usado para preparar a placa do Arduino para receber as informações do aplicativo. No código da função *setup* (configuração) encontra-se o *Serial.begin*, que é uma função de comunicação serial onde é configurado a taxa de comunicação em *bits* por segundo, neste caso, 9600 *bits* por segundo. Em *pinMode* é configurado o pino 13, que está conectado a um led, do Arduino como saída.

```
void setup() {  
  // put your setup code here, to run once:  
  Serial.begin(9600); //Tx0 and Rx0 //Set Baud Rate to 9600 for Serial Communication Tx0 and Rx0  
  Serial1.begin(9600); //Tx1 and Rx1 //Connected to Bluetooth Module HC-05 (Bluetooth 2.0)  
  //Serial2; //Tx2 and Rx2  
  //Serial3; //Tx3 and Rx3  
  
  pinMode(13, OUTPUT); //Set Pin 13 as Output (Connected to LED)  
}
```

Listagem 1 - Código do Arduino (Parte 1)

FONTE: (AUTORIA PRÓPRIA).

Na função *loop* (retorno), se *Serial.available* for maior que zero, isso significa que a leitura só será realizada se houver dados disponíveis, consequentemente retornará a quantidade de *bytes* disponíveis no *buffer* de leitura, então se o valor do *buffer* for 1, *digitalWrite* inverte o valor do *led* conectado na porta 13 para *High* (alto), liga o *led*, imediatamente *Serial.println* imprime na tela a mensagem “LED ON” (*Led* ligado). Caso contrário, se o valor do *buffer* for 0, *digitalWrite* inverte o valor do *led* conectado na porta 13 para *Low* (baixo), desliga o *led*, imediatamente *Serial.println* imprime na tela a mensagem “LED OFF” (*Led* desligado).

```

void loop() {
  // put your main code here, to run repeatedly:

  if(Serial1.available ()>0)
  {
    int buffer_value = Serial1.read();
    if(buffer_value == '1')
    {
      digitalWrite(13, HIGH);    //Turn ON LED
      Serial.println("LED ON");  //Arduino Terminal of Desktop
      Serial1.println("LED ON"); //Bluetooth Terminal on Mobile
    }
    else if(buffer_value == '0')
    {
      digitalWrite(13, LOW);     //Turn OFF LED
      Serial.println("LED OFF"); //Arduino Terminal on Desktop
      Serial1.println("LED OFF"); //Bluetooth Terminal on Mobile
    }
  }
}

```

Listagem 2 - Código do Arduino (Parte 2)

FONTE: (AUTORIA PRÓPRIA).

Na Figura 22 é ilustrado a tela do aplicativo no programa *Android Studio*, com todos os componentes usados para a criação do mesmo. Foram usados componentes como botões, textos, listas de visualização e *switch button*. As cores escolhidas no protótipo e na interface futura são para melhor visualização das letras e conforto visual. O tamanho da fonte se refere a mesma diretriz de usabilidade: conforto visual.

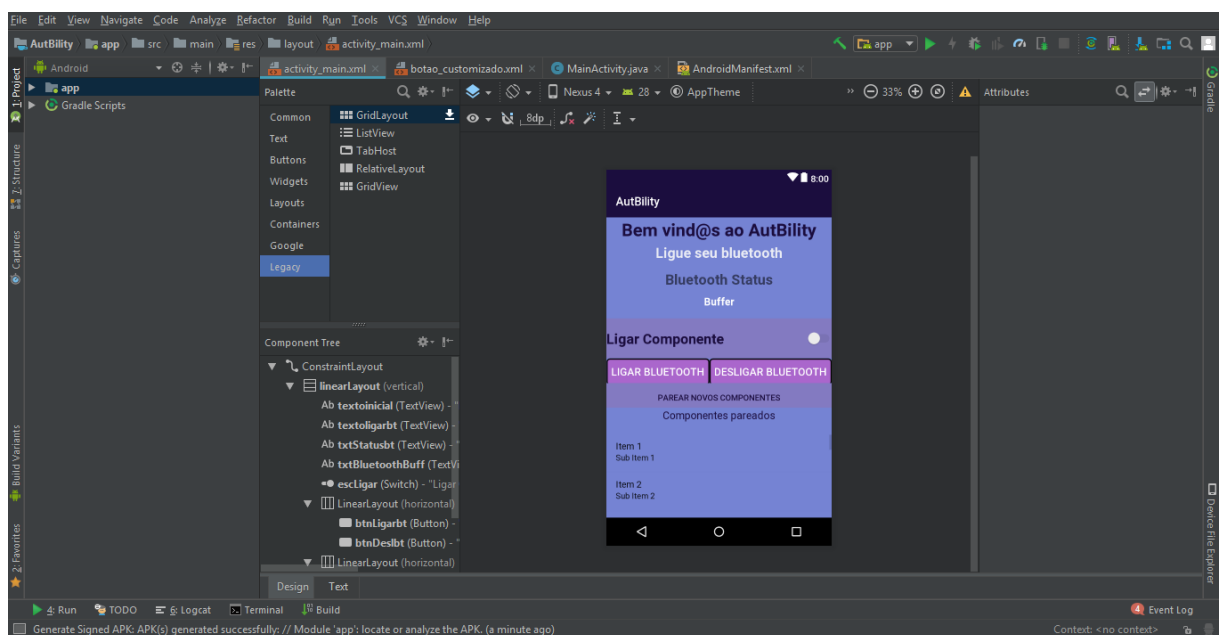


Figura 22 - Interface gráfica do protótipo no programa *Android Studio*

FONTE: (AUTORIA PRÓPRIA).

Na Figura 23 é ilustrada a tela do aplicativo em *smartphone* Android. É exibido um texto de boas-vindas ao acessar o aplicativo, também se exibe um texto com a mensagem “Ligue seu *bluetooth*” para o usuário ligar antes de iniciar o programa. O botão ligar *bluetooth* se refere a iniciar o serviço *bluetooth* dentro do aplicativo. O botão Parear Novos Componentes serve para enviar os componentes achados pelo *bluetooth* do usuário para a lista abaixo do texto “Componentes pareados”. O texto “*Bluetooth Status*” é exibido quando o dispositivo emparelhar com o *bluetooth* do Arduino. O *buffer* serve para mostrar a atividade atual do aplicativo. Por fim, ligar componente é um texto precedido de um interruptor para envio do sinal ligar ou desligar para o Arduino. A versão Android usada no dispositivo da Figura é a 8.1.0.

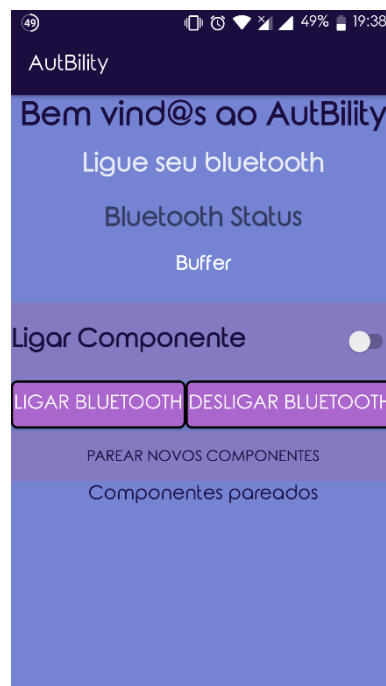


Figura 23 - Interface gráfica do protótipo em um *smartphone*.

FONTE: (AUTORIA PRÓPRIA).

O protótipo obteve um resultado satisfatório na verificação de funcionalidade, que estão exibidos nas Figuras 24 e 25, demonstrando assim que o aplicativo é funcional e que pode ser melhorado ao longo do tempo, com o estudo e o aprofundamento de acordo com as propostas de interface gráfica do usuário aqui exibidas, juntamente, com a tabela de requisitos proposta neste documento. Na Figura é ilustrado um *smartphone* com o aplicativo aberto, a placa Arduino conectada com o

módulo *bluetooth* e o *protoboard*. Neste *protoboard* é conectado um pequeno Led que está desligado no momento, já no aplicativo é exibida a mensagem “LED OFF”.

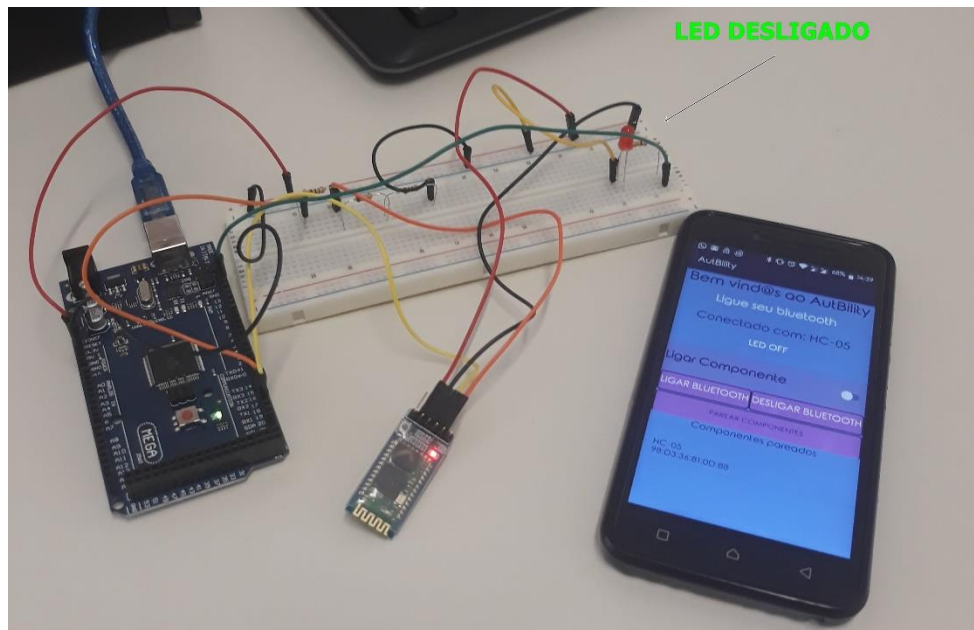


Figura 24 - Protótipo do aplicativo inicialmente com o Led desligado

FONTE: (AUTORIA PRÓPRIA).

Na Figura 25 é ilustrado o *smartphone*, a placa Arduino conectada no módulo *bluetooth* e o *protoboard*. Neste *protoboard* está conectado um pequeno Led aceso, no aplicativo exibe a mensagem “LED ON”.

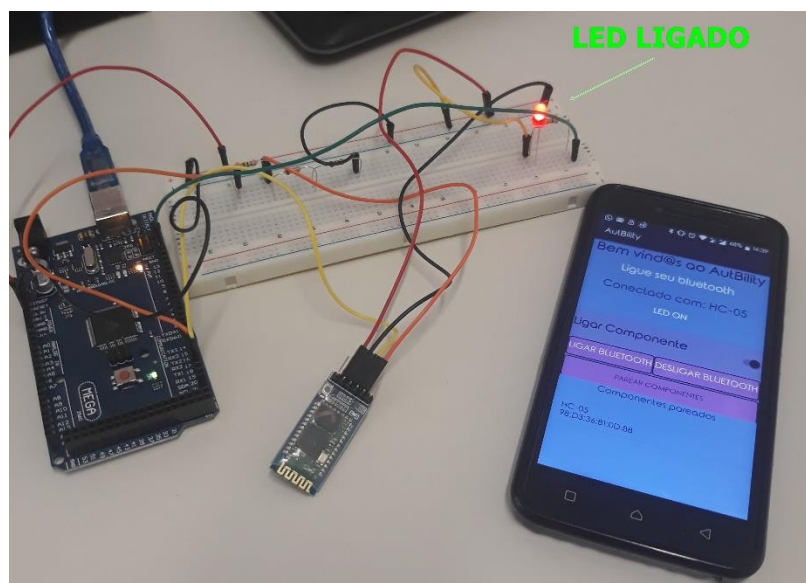


Figura 25 - Protótipo do Arduino e aplicativo em funcionamento, com o Led ligado.

FONTE: (AUTORIA PRÓPRIA).

Na Figura 26 é ilustrado o Arduino Mega conectado ao módulo *bluetooth* HC-05 e suas conexões com placa *protoboard*.

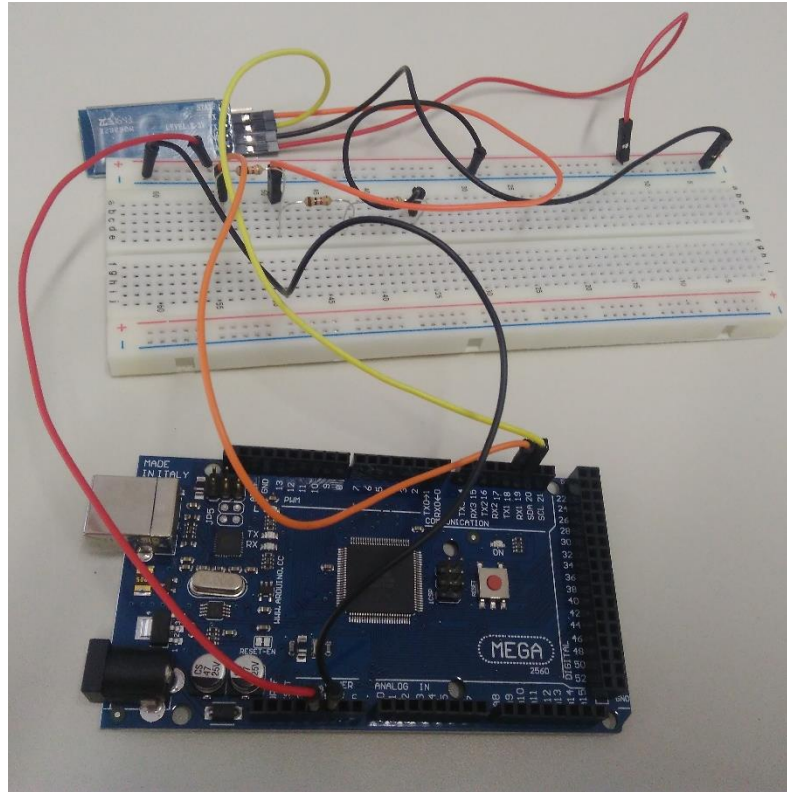


Figura 26 - Circuito Arduino na placa *protoboard* com o módulo *bluetooth*.

FONTE: (AUTORIA PRÓPRIA).

5 CONCLUSÕES

Desenvolver um aplicativo é uma tarefa árdua e minuciosa, principalmente com um tema onde tem que analisar todas as opções viáveis para manter-se dentro das necessidades das pessoas idosas e com dificuldade de locomoção. Os requisitos coletados na análise literária e na criação de diagramas UML ajudaram a chegar no protótipo que foi exibido neste trabalho.

A pesquisa, o estudo dos aplicativos relacionados e a análise literária, foi de grande importância para aprofundar a teoria de que não existe tantos aplicativos voltados para a real necessidade das pessoas idosas e com dificuldade de locomoção, que é diminuir seus obstáculos diários, principalmente dentro de casa. Os diagramas UML ajudaram a avançar significativamente na elaboração e no entendimento dos requisitos em sua completude, apesar de que não se pode compreender todos os requisitos aqui especificados.

As limitações encontradas no protótipo é que este só funciona para ligar e desligar componentes que estejam conectados ao Arduino com o módulo *bluetooth*, ou seja, o aplicativo não engloba todos os requisitos elicitados, isto se deu pelo prazo para sua criação e aprofundamento de desenvolvimento e estudo.

O protótipo inicial teve uma grande importância pois seu funcionamento simples, mesmo que de fase apenas de verificação de performance, demonstrou em resultados positivos, mas que podem ser melhorados com estudo e tempo para aprofundamento, para por fim ser totalmente funcional e abranger todos os requisitos e até receber mais aplicações ao longo da pesquisa.

Para trabalhos futuros este trabalho será de grande ajuda principalmente para saber quais requisitos compreender e atender a mais requisitos, se necessário. Servindo também como base para melhorias no aplicativo e no desenvolvimento de uma continuação melhorada para este tema.

REFERÊNCIAS

- ASAI, Yuki *et al.* 2016. **ExtendedHand on Wheelchair**. In: Proceedings of the 29th Annual Symposium on User Interface Software and Technology (UIST '16 Adjunct). ACM, New York, NY, USA, 147-148.
- ASSIS, Cleverson Faber de *et al.* O impacto da automação na redução do estoque em processo (WIP). **Revista Espacios**, São Paulo, v. 38, n. 11, p.15-29, 28 set. 2016. Disponível em: <<http://www.revistaespacios.com/a17v38n11/a17v38n11p15.pdf>>. Acesso em: 26 ago. 2018.
- BEZERRA, Eduardo. **Princípios de análise e projeto de sistemas com UML**. 3.ed. Rio de Janeiro: Elsevier, 2015. 398p.
- CONDE, Miguel Á. *et al.* 2014. **Mobile apps repository for older people**. In: Proceedings of the Second International Conference on Technological Ecosystems for Enhancing Multiculturality (TEEM '14). ACM, New York, NY, USA, 725-731.
- DEITEL, P; DEITEL, H. **Java: Como Programar**. Tradução Edson Furmankiewicz. 10. Ed. São Paulo: Pearson Education do Brasil, 2017. 935p. Disponível em: <<https://ufersa.bv3.digitalpages.com.br/users/publications/9788543004792/pages/2>>. Acesso em: 11 mar. 2018.
- DURELLI, V. H. S., VIANA, M. C. e PENTEADO, R. A. D. “**Uma proposta de reuso de interface gráfica com o usuário baseada no padrão arquitetural MVC**”. In: Simpósio Brasileiro de Sistemas de Informação, Rio de Janeiro – RJ, anais SBSI, 2008, p. 48-59.
- FRÓES, Gabriel. **Artigo sobre a linguagem Java**. Disponível em: <<https://www.codigofonte.com.br/artigos/artigo-sobre-a-linguagem-java>>. Acesso em: 23 mar. 2018.
- HASHIZUME, Satoshi *et al.* 2018. **Telewheelchair: the Remote Controllable Electric Wheelchair System combined Human and Machine Intelligence**. In: Proceedings of the 9th Augmented Human International Conference (AH '18). ACM, New York, NY, USA, Article 7, 9 pages.
- IBGE - Instituto Brasileiro de Geografia e Estatística – (2015). Disponível em: <ftp://ftp.ibge.gov.br/Acesso_a_Internet_e_posse_celular/2015/Tabelas_de_Resultados/ods/01_Pessoas_de_10_Anos_ou_Mais_de_Idade/> - Acesso em: 12 março 2018.
- _____. (2010). Disponível em: <<https://censo2010.ibge.gov.br/noticias-censo.html?busca=1&id=3&idnoticia=2170&t=censo-2010-numero-catolicos-cai-aumenta-evangelicos-espiritas-sem-religiao&view=noticia>> - Acesso em: 16 set 2018.
- LARMAN, Craig. **Utilizando UML e padrões: uma introdução à análise e ao projeto orientados a objetos e ao desenvolvimento iterativo**. 3.ed. Porto Alegre: Artmed, 2007. 695p.

LECHETA, Ricardo R. **Google Android: Aprenda a criar aplicações para dispositivos móveis com o Android SDK**. 3. Ed. São Paulo: Novatec Editora, 2013. 821p.

LIMA, Mariana. Aos 10 anos e com 2 bilhões de usuários, sistema Android mira emergentes. **O Estadão**. São Paulo, p. 13-3. 04 mar. 2018. Disponível em: <<https://link.estadao.com.br/noticias/cultura-digital,aos-10-anos-e-com-2-bilhoes-de-usuarios-sistema-android-mira-emergentes,70002212394>>. Acesso em: 26 ago. 2018.

MCROBERTS, M. **Arduino Básico**. Tradução Rafael Zanolli. 1. Ed. São Paulo: Novatec Editora, 2011. 456p.

NICHELE, Daniel B. **Automação Residencial: Um grande auxílio para idosos e deficientes**. 2010. 62 f. Monografia – Universidade de São Francisco, Itatiba, 2010. Disponível em: <<http://lyceumonline.usf.edu.br/salavirtual/documentos/1897.pdf>>. Acesso em: 22 maio 2010.

PINTO, Pedro M. T. L. N; GOMES, Antônio R. L. **Segurança na conectividade WiFi em dispositivos móveis: Estudo de caso do Iphone**. E-xacta, Belo Horizonte, v. 4, n. 3, dez. 2011. Disponível em: <<http://revistas2.unibh.br/index.php/dcet/article/viewFile/331/406>>. Acesso em: 11 mar. 2018.

REVISTABW. UML: Diagrama de Estados. **Revista Brasileira de Web: Tecnologia**. Disponível em <<http://www.revistabw.com.br/revistabw/uml-diagrama-de-estados/>>. 26 dez. 2013. Acesso em: 19 ago. 2018.

ROCKENBACH, Suzete. **Arquitetura, automação e sustentabilidade**. 2004. 193 f. Dissertação (Mestrado) - Curso de Arquitetura e Urbanismo, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2004. Disponível em: <<https://www.lume.ufrgs.br/bitstream/handle/10183/6773/000489913.pdf>>. Acesso em: 08 set. 2018.

RODRIGUES, Guilherme R. **Smartphone e suas tecnologias**. 2009. 77 f. Trabalho de Conclusão de Curso – Escola de Engenharia de São Carlos, da Universidade de São Paulo, São Carlos, 2009. Disponível em: <http://www.tcc.sc.usp.br/tce/disponiveis/18/180450/tce-23042010-094332/publico/Rodrigues_Guilherme_Rodrigues_e.pdf>. Acesso em: 11 mar. 2018.

SINGH, H. *et al.* "IoT based smart home automation system using sensor node", 2018, *4th International Conference on Recent Advances in Information Technology (RAIT)*, Dhanbad, 2018, pp. 1-5.

SIEBRA, Claurton *et al.* 2017. **Toward accessibility with usability: understanding the requirements of impaired uses in the mobile context**. In: Proceedings of the 11th International Conference on Ubiquitous Information Management and Communication (IMCOM '17). ACM, New York, NY, USA, Article 6, 8 pages.

SILVA, Bruna Roberta Seewald da. **Sistema de automação residencial de baixo custo para redes sem fio**. 2014. 55 f. Trabalho de Graduação – Instituto de Informática, da Universidade Federal do Rio Grande do Sul, Porto Alegre, 2013.

Disponível em: < <http://www.lume.ufrgs.br/handle/10183/101188>>. Acesso em: 6 dez. 2017.

SILVA, Jorge F. M. C. *et al.* **Sistema Operacional Android como Controle de Robôs Móveis via Bluetooth**. Sba Controle & Automação, Fortaleza, v. 0, n. 0, p.1-4, abr. 2012. Disponível em: <<http://sistemaolimpico.org/midias/uploads/66b7cf53ae6560bd8d8896db11c9ed5a.pdf>>. Acesso em: 22 jul. 2018.

SILVA, Leandro L. B.; PIRES, Daniel F.; NETO, Silvio C. **Desenvolvimento de Aplicações para Dispositivos Móveis: Tipos e Exemplo de Aplicação na plataforma iOS**. In: II Workshop de Iniciação Científica em Sistemas de Informação, Goias, maio de 2015. Disponível em: <<http://www.lbd.dcc.ufmg.br/colecoes/wicsi/2015/004.pdf>>. Acesso em: 11 mar. 2018.

SILVA, Paulo César Barreto da. Utilizando UML: Diagramas de Implantação, Comunicação e Tempo. **Revista Sql Magazine**, São Paulo, v. 0, n. 0, p.2-22, 25 mar. 2010.

SOMMERVILLE, Ian. **Engenharia de software**. 9. ed. São Paulo: Pearson Addison Wesley, 2011. 529 p.

THOMSEN, A. **O que é Arduino?**. Disponível em: <<https://www.filipeflop.com/blog/o-que-e-arduino/>>. Acesso em: 11 mar. 2018.

_____. **Automação Residencial com Arduino: acenda lâmpadas pela Internet**. Disponível em: <<https://www.filipeflop.com/blog/automacao-residencial-com-arduino-acenda-lampadas-pela-Internet/>>. Acesso em: 11 mar. 2018.

TRINDADE, J. M. F.; FISCHER, L. G. **Estudo e Aplicação de Padrões**. Notas de Aula. Universidade Federal do Rio Grande do Sul. Rio Grande do Sul, p. 13. 2008. Disponível em: <https://www.researchgate.net/publication/268434484_Estudo_e_Aplicacao_de_Padrees>. Acesso em: 26 ago. 2018.

VIEGAS, Gustavo. Introdução a UML. **Revista Sql Magazine**, São Paulo, v. 1, n. 1, p.0-0, 05 ago. 2009. Disponível em: <<https://www.devmedia.com.br/introducao-a-uml/6928>>. Acesso em: 26 ago. 2018.

WORTMEYER, C.; FREITAS, F.; CARDOSO, L. **Automação Residencial: Busca de Tecnologias visando o Conforto, a Economia, a Praticidade e a Segurança do Usuário**. In: II Simpósio de Excelência em Gestão e Tecnologia – SEGeT'2005, Rio de Janeiro, 2015.

APÊNDICE A – TABELA DE REQUISITOS DO SISTEMA E DESCRIÇÃO.

Requisito	Descrição
R01 – Cadastrar usuário.	O usuário deve criar um login e uma senha para usar as funcionalidades do aplicativo. Esses dados de entrada serão armazenados no banco de dados SQLite.
R02 – Editar usuário.	O usuário pode editar o login e senha do aplicativo. O novo usuário e/ou a senha será armazenado no banco de dados SQLite.
R03 – Excluir usuário.	O usuário pode excluir o login e a senha, caso decida por não usar mais o aplicativo e queira remover seus dados dele, então o usuário poderá excluir seus dados armazenados no banco de dados SQLite, por meio da funcionalidade Excluir Usuário, no menu Configurações do aplicativo. Logo após ele voltará para a interface inicial do aplicativo
R04 – Cadastrar impressão digital.	O usuário pode cadastrar sua impressão digital como chave de acesso ao aplicativo por meio do menu de entrada, na opção Cadastrar Impressão Digital, onde o aplicativo irá <i>scanear</i> o dedo escolhido e armazenará a digital no banco de dados SQLite.
R05 – Editar impressão digital.	O usuário pode editar a impressão digital cadastrada no banco de dados por meio do menu Configurações, na

	opção Editar Impressão Digital, onde o aplicativo vai disponibilizar essa opção e o usuário cadastrará outro dedo de sua preferência.
R06 – Excluir impressão digital.	O usuário pode excluir a impressão digital cadastrada no banco de dados, indo no menu Configurações, opção Excluir Impressão Digital e a sua digital será removida do banco de dados permanentemente.
R07 – Cadastrar componente.	O usuário pode cadastrar os componentes no menu Cadastrar componentes na interface inicial de usuário, onde ele poderá observar a lista de dispositivos conectados ao <i>bluetooth</i> do Arduino e selecionando o dispositivo desejado, toca no botão cadastrar e esse componente será listado no menu de lista de componentes salvos.
R08 – Editar componentes.	O usuário pode editar os componentes no menu Editar componentes, onde ele observará uma lista de dispositivos salvos por ele, selecionará o desejado e por toque no botão Editar, ele poderá mudar o rotulo do componente.
R09 – Excluir componentes.	O usuário pode excluir os componentes no menu Excluir componentes, onde ele observará uma lista de dispositivos salvos por ele previamente. Selecionará por

	meio de toque e no botão Excluir o removerá do banco de dados.
R10 – Descrição auditiva.	O usuário que tenha problemas visuais pode tocar no item desejado para ouvir qual função se está tocando e o que ela faz, com a funcionalidade <i>talkback</i> .
R11 - Emitir alerta vibratório.	O usuário poderá ativar um alerta vibratório caso queira ser avisado do início da atividade de algum componente.
R12 – Edição de cores.	O usuário poderá mudar as cores do aplicativo para maior conforto indo no menu Configurações e selecionando a opção Editar opções do aplicativo, nessa funcionalidade vai ter o menu de cores onde o usuário poderá colocar a cor de fundo, fonte e blocos da forma que lhe agradar.
R13 – Edição de fonte.	O usuário poderá mudar a fonte do aplicativo para maior conforto indo no menu Configurações e selecionando a opção Editar opções do aplicativo, nessa funcionalidade vai ter o menu de opção de fontes onde o usuário poderá decidir a fonte e o tamanho da forma que lhe agradar.
R14 - Emitir alerta luminoso.	Caso o usuário queira ser notificado sobre o início do funcionamento do aparelho, ele poderá decidir por seu dispositivo emitir alerta luminoso no

	começo do funcionamento da funcionalidade desejada.
R15 – Agendamento de tarefas.	O usuário poderá agendar que os dispositivos desejados sejam ligados e desligados em horários de sua preferência.
R16 – Didática de usabilidade.	No menu ajuda, o usuário receberá instruções por vídeo aula ou por tutorial descritivo de como usar as funcionalidades desse aplicativo, proporcionando maior independência.

Tabela 2 - Requisitos do sistema