

**FACULDADE DE TECNOLOGIA SÃO BERNARDO DO CAMPO
“ADIB MOISÉS DIB”**

**GABRIEL BALDINOTTI BARBOSA
RAFAEL VOLCOV SANTOS DE PAULA
ROGÉRIO BICHO**

AUTOMAÇÃO RESIDENCIAL VIA SMARTPHONE

São Bernardo do Campo – SP
Dezembro/2018

**GABRIEL BALDINOTTI BARBOSA
RAFAEL VOLCOV SANTOS DE PAULA
ROGÉRIO BICHO**

AUTOMAÇÃO RESIDENCIAL VIA SMARTPHONE

Trabalho de conclusão de curso apresentado à faculdade de tecnologia de São Bernardo do campo “Adib Moises Dib” como requisito parcial para a obtenção do título de Tecnólogo em Automação Industrial.

Orientador: Prof. Esp. Jorge Luiz Sarapka.

São Bernardo do Campo – SP
Dezembro/2018

**GABRIEL BALDINOTTI BARBOSA
RAFAEL VOLCOV SANTOS DE PAULA
ROGÉRIO BICHO**

AUTOMAÇÃO RESIDENCIAL VIA SMARTPHONE

Trabalho de conclusão de curso apresentado à faculdade de tecnologia de São Bernardo do campo “Adib Moises Dib” como requisito parcial para a obtenção do título de Tecnólogo em Automação Industrial.

Trabalho de conclusão de curso apresentado e aprovado em __/__/__

Banca examinadora:

Prof. Esp. Jorge Luiz Sarapka, FATEC SBC – Orientador

Prof. _____, FATEC SBC – Avaliador

Prof. _____, FATEC SBC – Avaliador

“A verdadeira motivação vem de realização, desenvolvimento pessoal, satisfação no trabalho e reconhecimento”.

FREDERICK HERZBERG

RESUMO

O intuito deste trabalho é utilizar conceitos obtidos ao longo do processo de aprendizagem para automatizar uma residência através de um aplicativo instalado em um *smartphone*. O sistema operacional utilizado é o Android, e a comunicação é feita pela através do *Bluetooth*, com a finalidade de proporcionar o monitoramento da residência a partir qualquer aparelho que disponha da ferramenta. Devido à sua praticidade, o uso de *smartphones* juntamente com novas tecnologias, tem se intensificado como interface de controle ou monitoramento. Este trabalho utiliza-se de um aplicativo para controle da iluminação e do motor do portão da garagem, Arduíno e relés para o funcionamento da maquete, *shield bluetooth* para comunicação, e caixa *steck* para agrupar os componentes da central de controle. O *shield bluetooth* recebe comandos do aplicativo e envia para o Arduíno que aciona os respectivos relés. A IHM serve como uma central para monitorar e acionar os dispositivos da casa, enviando sinais por meio de uma placa conversora de protocolo do Arduíno para a IHM.

Palavras-chave: Automação Residencial. Arduíno. Bluetooth. Smartphone. IHM.

ABSTRACT

The purpose of this work is to use concepts obtained throughout the learning process to automate a residence through an application installed on a smartphone. The operating system used is Android, and the communication is made through the Bluetooth, in order to provide the monitoring of the residence from any device that has the tool. Due to its practicality, the use of smartphones along with new technologies, has intensified as a control or monitoring interface. This work uses an application to control the lighting and the garage door motor, Arduino and relays for the layout, bluetooth shield for communication, and steck box to group the components of the control center. The bluetooth shield receives commands from the application and sends it to the Arduino that activates its relays. The HMI serves as a switchboard to monitor and trigger household appliances, sending signals through an Arduino protocol converter board to the HMI.

Keywords: Home Automation. Arduino. Bluetooth. Smartphone. IHM.

LISTA DE FIGURAS

Figura 1.1 – Evolução dos celulares	12
Figura 1.2 – Exemplo de automação residencial	13
Figura 1.3 – Consumo médio de energia residencial	14
Figura 1.4 – Relação de custo e tempo de uso de lâmpadas	15
Figura 1.5 – Vantagens da fechadura elétrica FX 2000	15
Figura 1.6 – Placa Arduíno uno	16
Figura 1.7 – Placa shieldbluetoothlowenergy.....	17
Figura 1.8 – Microcontrolador.....	17
Figura 1.9 – Relações entre meios de comunicação	20
Figura 1.10 – Funcionamento interno do android.....	24
Figura 2.1 – Organograma do projeto	26
Figura 2.2 – Fluxograma do projeto	27
Figura 3.1 – Projeto finalizado.....	30
Figura 3.2 – Dimensões da maquete utilizada	32
Figura 3.3 – Caixa <i>steck</i> sem componentes.....	32
Figura 3.4 – Caixa <i>steck</i> finalizada.....	33
Figura 3.5 – Ligação do circuito elétrico no software proteus.....	34
Figura 3.6 – Ligações do <i>shield</i> Bluetooth	35
Figura 3.7 – Montagem do circuito conversor RS-232/RS-485	36
Figura 3.8 – Menu do aplicativo	37
Figura 3.9 – Lógica do aplicativo.....	38
Figura 3.10 – Desenvolvimento da lógica do aplicativo	38
Figura 3.11 – Lógica para efeito visual de funcionamento	39
Figura 3.12 – Declaração de variáveis	40
Figura 3.13 – Tela do software da IHM	41
Figura 3.14 – Configuração do endereço lógico do software da IHM.....	42

LISTA DE ABREVIATURAS

ASCII	American Standard Code for Information Interchange
CD	Compact Disc
CLP	Controlador Lógico Programável
CPU	Central Processing Unit
DP	Decentralized Periphery
DVD	Digital Versatile Disc
FHSS	Frequency Hopping Spread Spectrum
FMS	Fieldbus Message Specification
Gbps	Gigabit por segundo
HD	Hard Disk
IBM	International Business Machines
IHM	Interação Homem - Máquina
INMETRO	Instituto Nacional de Metrologia, Normalização e Qualidade industrial
IP	Internet Protocol
IPC	Inter Process Control
ISO	International Organization for standardization
Kbps	Kilobit por segundo
LED	Light Emitting Diode
MBP	Manchester Bus Powered
Mbps	Megabit por segundo
OSI	Open System Interconnection
PA	Process Automation
PC	Personal Computer
RAM	Random Access Memory
RTU	Remote Terminal Unit
SMS	Short message service
TCP	Transmission Control Protocol
UWB	Ultra Wide Band
WI-FI	Wireless Fidelity
WWW	World Wide Web

SUMÁRIO

INTRODUÇÃO	09
1 FUNDAMENTAÇÃO TEÓRICA	11
1.1 Breve históricos da domótica e do <i>smartphone</i>	11
1.2 Eficiências do consumo de energia e controle de acesso	13
1.3 Plataformas Arduíno e microcontrolador.....	16
1.4 Transmissões de dados e protocolos de comunicação	19
1.5 Software e sistema operacional.....	23
2 METODOLOGIA	25
2.1 Temas problema e justificativa	25
2.2 Descrições do projeto através de organograma e fluxograma	26
2.3 Etapas teóricas e práticas para desenvolvimento do projeto	28
3 DESENVOLVIMENTO DO PROJETO	30
3.1 Montagem da maquete e caixa <i>steck</i>	32
3.2 Montagem dos circuitos eletrônicos no software Proteus.....	34
3.3 Construção do aplicativo para <i>smartphone</i>	36
3.4 Programação da lógica no programa Arduíno	40
3.5 Construção do software para IHM e integração com Arduíno.....	41
CONSIDERAÇÕES FINAIS	44
REFERÊNCIAS.....	45
APÊNDICE.....	49
ANEXO	54

INTRODUÇÃO

Ao longo do tempo a humanidade vem buscando formas de otimizar o conforto e economia de energia em suas residências. Com o avanço das tecnologias nas indústrias, é importante buscar aplicações em que se podem aproveitar ao máximo essas tecnologias para diminuir o trabalho e esforço humano.

Atualmente a domótica é um termo que se utiliza para implantação de tecnologias em atividades domésticas do dia-a-dia, se torna mais abrangente por incluir comunicação e sonorização de forma que o sistema seja capaz, não apenas interpretar parâmetros, mas também atuar conforme as circunstâncias, transmitindo as informações por sensores. Ela é uma tecnologia recente que tem como objetivo encontrar soluções para as necessidades do homem fazendo o mínimo de esforço possível, proporcionando maior conforto e qualidade de vida para quem utiliza.

A fim de melhorar a qualidade de vida dos cidadãos utilizando a tecnologia no âmbito residencial, foi pensado em desenvolver um projeto, qual consiste em realizar tarefas domésticas remotamente. Ao ser dado o comando, tarefas como controle de luminosidade, abertura de portas e janelas e acionamento de eletroeletrônicos podem ser controladas à distância.

Das explanações feitas, o objetivo deste trabalho que se intitula Automação Residencial Via Smartphone é desenvolver e implementar um sistema que gerencia consumo de energia e controle de acesso em ambientes residenciais. Justifica-se por trazer maior conforto e comodidade para seus usuários como ligar/desligar a luz de algum ambiente da casa, sem ter que se locomover até o interruptor físico para isso, além de proporcionar economia de energia elétrica. Outro benefício é a possibilidade de realizar esse controle independente da distância em que se encontra do local gerenciado.

Para o desenvolvimento do projeto, faz-se necessário o uso de uma placa Arduíno, um *shield bluetooth*, um aparelho *smartphone* que tenha o aplicativo desenvolvido instalado, além de componentes eletrônicos que são necessários para a instalação elétrica na residência.

Com a implementação do aplicativo para o controle da energia e dos outros sistemas presentes na residência, esperam-se melhorias nos aspectos destacados anteriormente, gerando níveis de comodidade e de consumo de energia satisfatórios, fazendo com que o sistema implementado compense o investimento, podendo sempre ser adaptado com outras melhorias desenvolvidas futuramente.

O trabalho é dividido em três capítulos:

Capítulo 1 – Fundamentação teórica: encontram-se as teorias e conceitos de autores renomados que solidificam a construção e desenvolvimento do projeto que se intitula Automação residencial via *smartphone*.

Capítulo 2 – Metodologia: encontram-se os métodos e técnicas que dão as diretrizes para a construção e desenvolvimento do projeto para criação do aplicativo.

Capítulo 3 – Desenvolvimento do projeto: descreve passo a passo a construção do projeto, ilustração de figuras e tabelas para melhor entendimento. Mostra a ligação entre a teoria e o processo de montagem, bem como o seu funcionamento.

E finalmente, as Considerações finais: encontram-se os objetivos propostos, justificativas, relações entre as teorias e práticas, conquistas alcançadas, pontos fracos e fortes e sugestões para futuros trabalhos.

1 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo encontram-se as teorias que dão sustentação ao projeto que se intitula Automação Residencial via *Smartphone*.

1.1 Breve histórico da domótica e do *smartphone*

Muratori (2011) destaca que domótica é a automatização de serviços através de sistemas tecnológicos integrados com a finalidade de atender as demandas de: segurança, gestão energética, comunicação e conforto, promovendo melhor qualidade de vida, reduzindo o trabalho doméstico e possibilitando constantemente novas aplicações.

Sena (2005) enfatiza que a domótica se originou a partir de fundamentos aplicados na indústria, onde são utilizadas lógicas de funcionamento mais complexas quando comparadas com as residenciais. Os conceitos empregados na domótica, apesar de mais simples, necessitam de diferentes formas de interfaces e dispositivos além de configurações requisitadas pelo cliente.

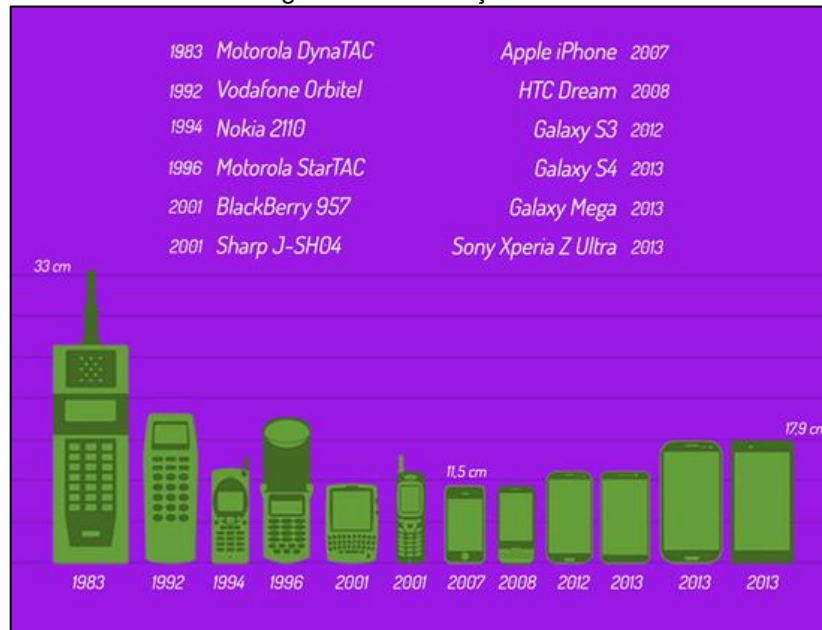
Na década de 80 novas possibilidades de automação surgem no mercado devido ao desenvolvimento da informática pessoal, entretanto, essa tendência demanda o funcionamento contínuo do PC além de centralizar o controle de forma que se houver alguma falha, todo o sistema será comprometido.

Com a implantação da internet banda larga surgem novas possibilidades para a automação residencial, pois é possível o controle/monitoramento da residência de lugares diferentes, necessitando apenas que o serviço esteja disponível. Devido à praticidade, esse sistema tem sido cada vez mais aperfeiçoado, e buscado por consumidores.

Coutinho (2014) destaca que o Simon, desenvolvido em 1992 pela IBM foi o primeiro celular considerado um *smartphone*, ele possuía funções consideradas avançadas para sua época como tela sensível ao toque além da capacidade de receber e enviar fax e *e-mails*.

Em 2001 é desenvolvido o *Blackberry* que se tornou um dos celulares mais populares da história, deu continuidade ao projeto de um *Pager* que enviava e-mails e mensagens SMS, além de ter sido o primeiro do mercado a possuir o teclado *Qwerty*. Posteriormente, em 2007 Steve Jobs proporciona uma revolução no mercado de celulares com o lançamento do *Iphone* da *Apple*, no qual se destaca até os dias atuais por sua interatividade e conectividade, conforme ilustra a Figura 1.1.

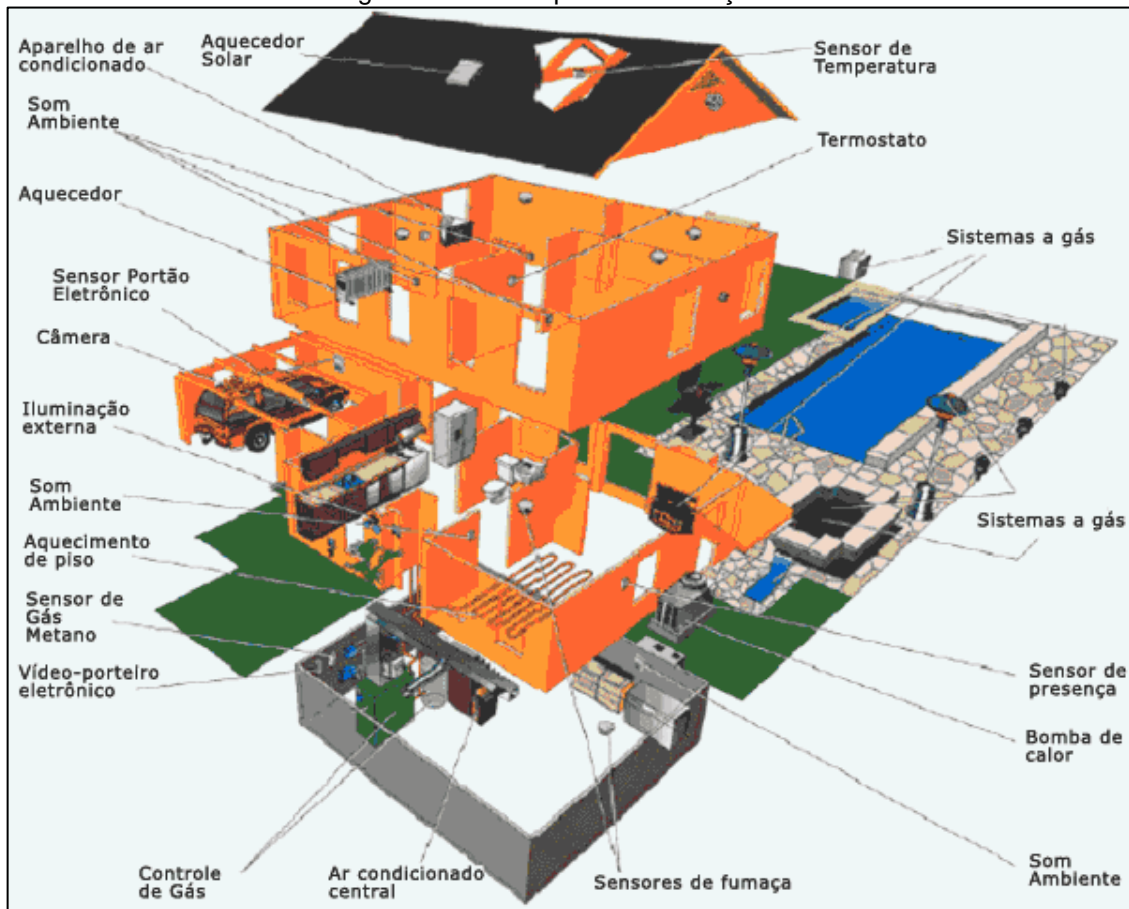
Figura 1.1 – Evolução dos celulares



Fonte: www.magicwebdesign.com.br, 2018

O conceito de automação residencial depende diretamente do uso de aparelhos eletrônicos para controle e monitoramento de um ambiente. O uso de *smartphones* é uma alternativa cada vez mais utilizada pela fácil interatividade dada ao usuário. Sua utilização, juntamente com uma plataforma de controle e os demais componentes, possibilita que diversas atividades como ligar lâmpadas e até controlar a temperatura ambiente sejam realizadas através dele. A Figura 1.2 mostra algumas opções de controle automatizado em uma residência.

Figura 1.2 – Exemplo de automação residencial

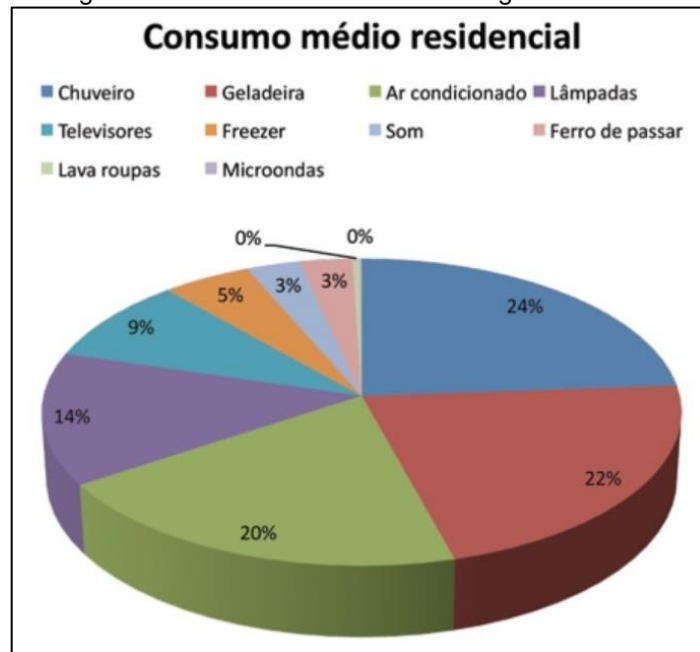


Fonte: www.controlled.com.br, 2009

1.2 Eficiência do consumo de energia e controle de acesso

O INMETRO – Instituto Nacional de Metrologia, Normalização e Qualidade Industrial (2015) enfatiza que a iluminação na domótica é um ponto vital para que a eficiência energética, o conforto visual e físico possa ser alcançado. A ideia de controle da iluminação artificial parte do princípio que durante o dia, são necessárias diferentes potências de luz para que o ambiente fique iluminado adequadamente, por exemplo, a quantidade de energia gasta em luz no período noturno é maior do que é usada no período matutino, devido à disponibilidade de luz natural ser maior e com isso pode-se poupar a energia elétrica. O foco em controle de luz artificial é devido ao fato de que se consome em média 14% da energia total gasta no mês, perdendo apenas para aparelhos que tem alto consumo de energia, como o ar condicionado, conservação de alimentos em *freezer*/geladeira e chuveiros para aquecimento de água, como mostra na Figura 1.3.

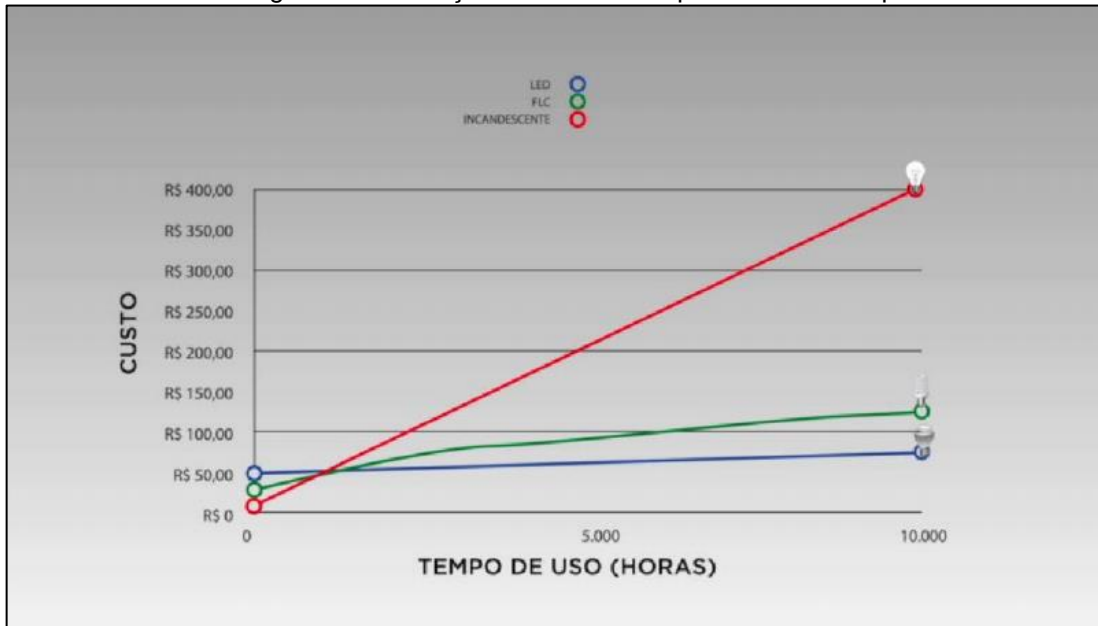
Figura 1.3 – Consumo médio de energia residencial



Fonte: www.osestoreletrico.com.br, 2018

O INMETRO – Instituto Nacional de Metrologia, Qualidade e Tecnologia (2015), enfatiza que ao longo dos anos foram surgindo diferentes tipos de lâmpadas para iluminação artificial, e cada vez mais vem surgindo tecnologias que melhoram a eficiência de energia consumida, custo - benefício e vida útil. O tipo de lâmpada que se escolhe influencia no resultado do consumo final e a relação do tempo de uso das lâmpadas, mostrando que a lâmpada incandescente (cor vermelha) tem um consumo maior que as lâmpadas do tipo compactas (cor verde) e as *LED's* (cor azul), levando em conta o fato de que sua vida útil é mais curta, aumentando os custos com manutenção. Essa característica não é um problema para as lâmpadas *LED*, que apesar de serem mais caras, proporcionam vantagens que os outros tipos de lâmpadas não têm, chegando a níveis de eficiência de até 10 vezes maiores que as incandescentes. Como mostra a Figura 1.4.

Figura 1.4 – Relação de custo e tempo de uso de lâmpadas

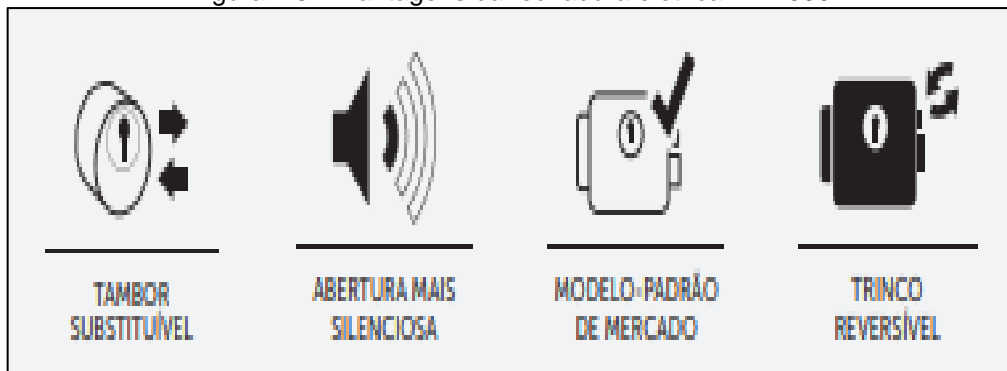


Fonte: www.inmetro.gov.br, 2018

Segundo Ferreira (2003), o controle de acesso físico é uma forma de proteger ambientes com o uso de algum equipamento, como por exemplo, catraca, cartão magnético, fechadura elétrica, biometria, crachá, chave e entre outros.

De acordo com a INTELBRAS – Indústria de Telecomunicação Eletrônica Brasileira (2018), por possuir características vantajosas à fechadura elétrica FX 2000 é a mais recomendável para uso residencial. A Figura 1.5 mostra vantagens da fechadura elétrica FX 2000.

Figura 1.5 – Vantagens da fechadura elétrica FX 2000



Fonte: www.intelbras.com.br, 2018

1.3 Plataforma Arduino e microcontrolador

Arduino (2018) destaca que a plataforma Arduino é composta por um software e hardware e seu principal objetivo é criar uma plataforma de fácil manipulação e prototipagem de projetos comandados por microcontrolador, fazendo uma interação direta com componentes periféricos como sensores, motores dentre outros através de uma linguagem de programação da plataforma.

Sua composição é como a de um computador, sendo composto por um microprocessador, memória RAM e *flash*, onde é guardado o software desenvolvido, contadores e temporizadores. A principal característica que difere um Arduino de um computador são os dispositivos que servem como entrada e saída, por exemplo, no Arduino utilizam-se circuitos elétricos e eletrônicos como entrada e saída, no computador são acessórios como teclado e caixas de som. Outra diferença com relação ao computador convencional é o tamanho e potência, onde é possível usar o Arduino em diversos circuitos montados, e não é necessário se limitar a periféricos disponíveis no mercado. Sua aparência é demonstrada na Figura 1.6.

Figura 1.6 – Placa Arduino uno



Fonte: www.arduino.cc, 2018

Segundo Arduino (2018), o Arduino opera com tensões entre 7V e 35V, que é fornecido por uma fonte que é ligada diretamente na tomada de 127V. É destacado também que essa tensão é usada para alimentar os *Shields*, que são placas complementares criadas para diversas aplicações, como conexão *Ethernet* e

Bluetooth. Os Shields Arduíno podem assumir diversas formas e funções como é mostrado na Figura 1.7.

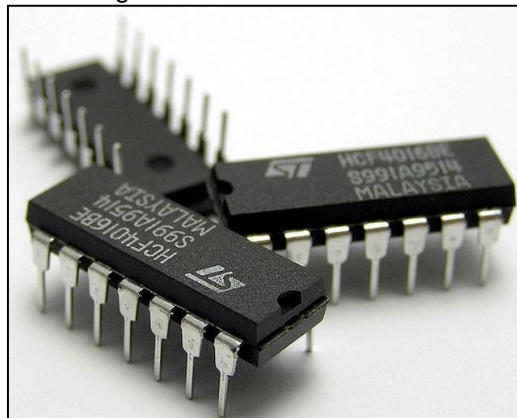
Figura 1.7 – Placa shield bluetooth low energy



Fonte: www.shieldlist.org, 2018

Penido e Trindade (2013) definem microcontrolador como um chip que tem a capacidade de um computador, contendo internamente um processador, CPU, memória, temporizadores e dispositivos que possibilitam a comunicação serial. Os circuitos digitais foram ficando cada vez mais complexos, impossibilitando realizar algumas ações apenas com portas digitais em termos de lógica por conta do espaço ocupado e preço. Eles são substituídos por microcontroladores, que gerenciam dispositivos periféricos com mais eficiência. A Figura 1.8 ilustra um microcontrolador.

Figura 1.8 – Microcontrolador



Fonte: www.eletronicaprogressiva.net, 2014

Souza (2005) ressalta que os microcontroladores são programáveis, estruturados e gravados dentro do componente pelo usuário, e que é esse programa que comanda toda a lógica do processo em que é implantado. O responsável pela inteligência e execução do programa é a CPU, que possui duas funções: a leitura do programa e sua interpretação. Algumas de suas aplicações são os eletrodomésticos, alarmes, celulares, brinquedos, dentre outras aplicações. As linguagens mais usadas para programar um microcontrolador são:

Linguagem C: usada pela primeira vez em um computador que tinha o sistema operacional UNIX, ela é desenvolvida por programadores para programadores, com o intuito de ser uma ferramenta bem conhecida pelos profissionais que atuam na área. É bastante difundida devido ao fato de ser uma linguagem flexível, portátil e eficiente, podendo ser usada em diversas aplicações proporcionando em todas elas, um alto desempenho em relação a velocidade de execução. É uma linguagem que reúne expressividade, portabilidade e eficiência, indicada para o desenvolvimento de *software* básico.

Assembly: é a linguagem mais antiga e a que a mais se parece com a linguagem nativa da máquina, devido ao fato de que seu acesso é diretamente ao *hardware* do computador. Para aplicar essa linguagem de programação são necessários conhecimentos sobre o sistema operacional do computador, uma vez que cada comando resulta em uma ação do sistema. Ele é programado em qualquer editor de texto, em seguida é transformado na linguagem da máquina para executar o que é programado, o que pode ser feito em diversos programas encontrados na internet (IRVINE, 2007).

1.4 Transmissão de dados e protocolos de comunicação

Segundo Nério (2003), a necessidade de verificação rápida de dados adquiridos em diferentes locais levou ao desenvolvimento das redes sem fio (*wireless*), que complementam as redes cabeadas. A possibilidade de transmissão remotamente promove maior flexibilidade em relação à localização das estações de controle, independente da distância e diferenciando-se das redes convencionais. Com o desenvolvimento da tecnologia, e aumento na velocidade de troca de dados, surgiram tecnologias como o *Bluetooth*, sendo este um método mais simples de transmissão levando em consideração seu baixo consumo energético.

A comunicação entre dispositivos utilizando o *Bluetooth* pode ser alternada entre cliente e servidor se necessário, um dos dispositivos é selecionado para ser o mestre da rede, enquanto o outro se torna um escravo, formando uma *piconet*, onde qualquer outro dispositivo que participe de sua área de alcance (cerca de 15 metros) torna-se um escravo, além disso, esse tipo de transmissão utiliza uma modulação FHSS (*frequencyhopping spread spectrum* – espectro de propagação de salto de frequência) na qual consiste em mudanças constantes de frequências entre seus 79 canais de modo sincronizado. Essas mudanças ocorrem cerca de 1600 vezes por segundo, com a finalidade de dificultar que ocorram interferências.

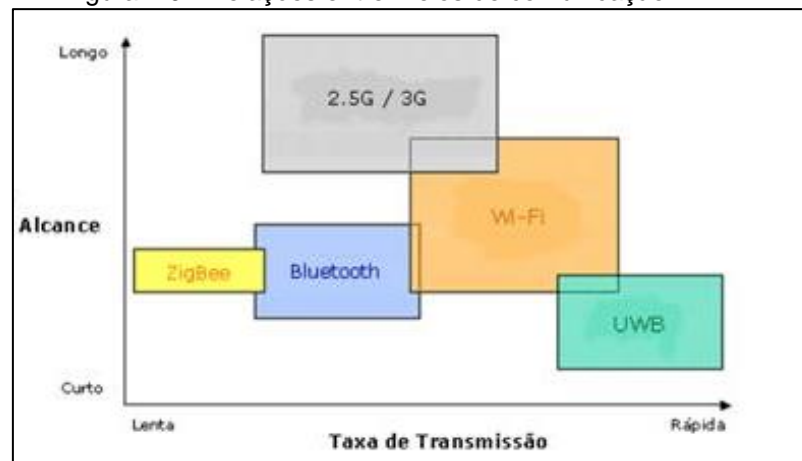
De acordo com Alecrim (2008), o termo “*Wi-Fi*”, refere-se a “*wireless fidelity*” e é uma forma de conexão segura e confiável. Sua configuração possibilita mais de um ponto de acesso para seus clientes, de modo que cada ponto nomeia a rede através de dados, permitindo a opção de ligar a rede que está em seu alcance. A rede Wi-Fi é amplamente utilizada, sendo popular devido às suas vantagens como sua flexibilidade para instalação, possibilidade de criar redes locais sem fios para usuários com redução dos custos de implantação e expansão como complemento a redes cabeadas, além da segurança promovida por senhas fortes utilizadas.

Otton (2009) destaca que a rede Wi-Fi também apresenta características negativas como alto consumo de energia quando comparado com o *Zigbee* e o *Bluetooth*, poluição excessiva devido a muitos pontos de acesso na mesma área, alta taxa de ruído podendo gerar interferência de dispositivos utilizando a mesma frequência 2,4 GHz.

Gislason (2007) enfatiza que além do *Bluetooth*, existe um protocolo de comunicação sem fio chamado *Zigbee*, no qual apresenta características como baixo consumo energético, infraestrutura de rede mais simples, o uso de baterias como alimentação e baixas taxas de transmissão.

O protocolo *Zigbee* possui baixo custo e se usa em redes de monitoramento de sensores e controle de dispositivos, porém não é utilizado em aplicações que necessitem grandes taxas de transmissões. A Figura 1.9 mostra relações entre meios de comunicação.

Figura 1.9 - Relações entre meios de comunicação



Fonte: www.teleco.com.br, 2016

A Figura 1.6 também mostra a relação entre alcance e taxa de transmissão da UWB (*Ultra Wide Band* - Banda ultra larga). É uma forma de transmissão que futuramente pode tornar-se padrão. Seu funcionamento permite que as frequências mudam em frações de segundo, não interferindo em aparelhos radiotransmissores.

Conforme Borges (2007), a rede de comunicação é a tecnologia que mais evoluiu nos últimos anos, pois ela permite uma comunicação rápida e confiável entre dispositivos.

Devido ao aumento na automatização de máquinas e sistemas, aumentaram a quantidade de cabos utilizados como entrada e saída na comunicação de equipamentos, com o intuito de economia e melhora no desempenho dos dispositivos, são utilizados barramentos (bus) para interligação dos equipamentos,

minimizando o tempo de manutenção. Seguem alguns protocolos e padrões que são utilizados na automação:

Para Caldiéri (2016), modbus é um protocolo de comunicação aberto utilizado para comunicação de dispositivos mestre-escravo permitindo que apenas um dispositivo possa iniciar as consultas enquanto os demais dispositivos respondem de acordo com a função que foi solicitada, empregando dois modos de transmissão que podem ser configurados de acordo com o seu usuário, que são: ASCII (*American Standard Code for Information Interchange* - Código Padrão Americano para Intercâmbio de Informações) e RTU (Remote Terminal Unit- Unidade de terminal remota), a transmissão dos seus dados se realiza fisicamente pelos padrões RS-232, RS-485 ou Ethernet (TCP/IP).

Para Freitas (2014), o padrão RS-232 admite apenas dois dispositivos conectados à rede, um mestre e um escravo, sua comunicação é feita de um ponto ao outro, sendo que a velocidade máxima de transmissão é de 115Kbps e a distância limite permitida entre os dispositivos conectados é 30 metros.

O padrão RS-485 é o mais utilizado no ramo industrial, permite velocidades de transmissão de dados de até 12 Mbps e em alguns casos específicos até 50Mbps, variando sua velocidade de transmissão de acordo com a distância total da rede. A distância máxima permitida nesse meio de comunicação é de 1200 metros e comporta no máximo 32 dispositivos conectados à rede.

O padrão Ethernet pode variar sua velocidade de transmissão de dados entre 10 Mbps até 10 Gbps, a distância máxima da rede neste caso é de 200 metros, variando sua velocidade de transmissão de acordo com a distância percorrida pela rede.

De acordo com Vidal e Vilela (2003), profibus é um protocolo de comunicação aberto, onde os dispositivos de diferentes fabricantes podem se comunicar sem a necessidade de qualquer adaptação em sua interface. Este protocolo utiliza como meio de comunicação os padrões RS-485, IEC 61158-2 ou até Fibra ótica. Sua arquitetura é baseada no modelo OSI (*“Open System*

Interconnection” - Interconexão de sistemas abertos) e está dividido principalmente em três perfis: Profibus FMS, Profibus DP e Profibus-PA.

- Profibus FMS (*FieldbusMessageSpecification* - Especificação da Mensagem Fieldbus) foi a primeira versão deste protocolo, projetado para comunicações entre CLP, IHM e PC, porém esta tecnologia apresenta pouca flexibilidade na transmissão de mensagens.

- Profibus DP (*DecentralizedPeriphery* - Periferia Descentralizada) é um protocolo de comunicação mais utilizado, foi desenvolvido basicamente para a comunicação entre sistemas de automação, devido sua alta velocidade de transmissão de dados e seu baixo custo, é voltado para sistemas de controle e utiliza como meio físico RS-485 ou fibra ótica.

- Profibus PA (*Process Automation* - Automação do processo) é um protocolo projetado para processos de automação industrial para trabalhar em ambientes de alto risco.

A arquitetura de rede Profibus conforme Souza (2012) é constituída por sete camadas, porém se usa somente três do modelo OSI (*Open System Interconnection* - Interconexão de sistemas abertos): camada 1 - física, camada 2 – enlace e camada 7 – aplicação.

- camada 1 - física: compreende apenas sinais binários, ela determina a distância máxima do cabeamento e o tipo de conector, ou seja, é a camada que recebe os dados e inicia a operação.

- camada 2 - enlace: trata as topologias de rede, onde são formados os pacotes de dados e direciona o caminho que a mensagem trafega.

- camada 7 - aplicação: responsável por realizar a interface entre a máquina e o usuário, nela ocorre à aplicação das informações que são transmitidas.

1.5 Software e sistema operacional

Pressman (2011) enfatiza que a execução dos softwares depende dos comandos baseados em programação pré-estabelecidos nas configurações dos hardwares, no qual formam os programas que são lidos pelo computador.

Existem inúmeros tipos de softwares, dentre eles os principais são: Software de sistemas e de aplicação.

- Software de sistemas: constituído por sistemas operacionais, dando base aos recursos utilizados para resolver problemas de programas de aplicação. É a junção de informações utilizadas para o processamento interno do computador que permite a comunicação entre o usuário e os periféricos do computador através de uma interface gráfica. Também cria programas-base para atender outros programas.

- Software de aplicação: trata-se de um programa de computador criado para ajudar o usuário a realizar uma tarefa específica, são aplicações que solucionam uma necessidade específica de negócio, processando dados comerciais ou técnicos a fim de facilitar a tomada de decisão do usuário, como o Microsoft Word, Excel, Power point e navegadores de internet.

Schulte (2016) diz que o sistema operacional é um meio de ligar usuário com o software, ele possui quatro funções típicas que são consideradas principais para o controle e monitoramento:

- Gerenciamento de processos: sistema multitarefas capaz de executar diversos processos simultaneamente, havendo também comunicação entre eles por meio de comunicação IPC (*Inter Process communication* - Comunicação entre processos) existente no sistema operacional.

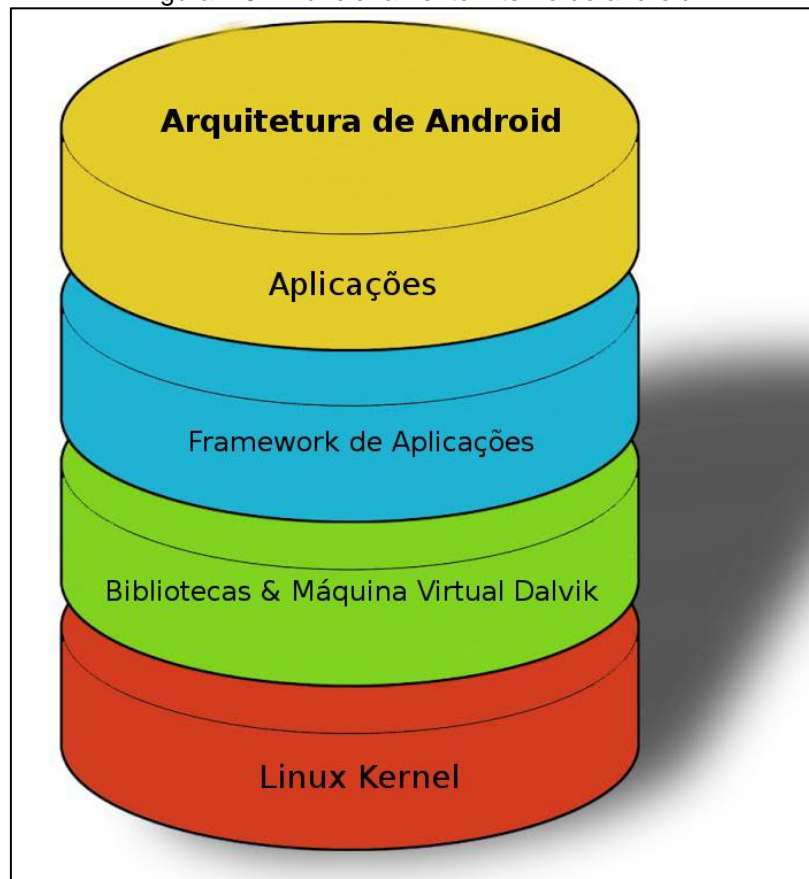
- Gerenciamento de memória: responsável por certificar e reservar memória para cada processo, para não ocorrer uma sobreposição de processo e economia de espaço de memória.

- Gerenciamento de arquivos: a memória de um computador é dependente de sua fonte de alimentação, ao ser desligada se perde os arquivos além da capacidade máxima de memória que não pode ser excedida. Uma maneira de evitar esse tipo de problema é utilizar memórias não instáveis, como pen drive, CD e HD.

- Gerenciamento de dados de entrada e saída: para funcionar corretamente é preciso que haja no sistema dados para serem processados e transmitidos, no caso do *smartphone* a coleta de dados se deve a informações de entradas por voz e toque na tela (*touch*), e dados de saída por meio da tela, alto falante ou conexão de acessórios.

Segundo Schulte (2016) o android é um sistema virtual, utilizado em *smartphones* que teve como inspiração o sistema operacional Linux. Operando com a biblioteca base do Kernel, ele é capaz de desenvolver aplicações em Java para os dispositivos que o utilizem, conforme mostra a Figura 1.8.

Figura 1.8 – Funcionamento interno do android



Fonte: www.mibqyyo.com, 2014

2 METODOLOGIA

Neste capítulo encontram-se os métodos e técnicas que dão as diretrizes para a construção e desenvolvimento do projeto intitulado Automação residencial via *smartphone*. Trata-se de uma pesquisa aplicada que é desenvolvida nas dependências da Fatec SBCampo.

Dentre os vários autores que abordam a metodologia científica, Severino (2013) aponta que a metodologia é a preparação metódica e planejada de um trabalho científico que se encontra inserida em uma sequência de etapas compreendendo: o tema-problema e justificativa, levantamento bibliográfico, seleção da bibliografia, desenvolvimento da construção do projeto e redação do texto.

O trabalho é construído de acordo com o Manual de Normalização de Projeto de Trabalho de Graduação da Fatec São Bernardo do Campo (2017) que se encontra ancorado nas normas ABNT. Ele é escrito em uma linguagem simples com características próprias e específicas e com terminologias adequadas.

2.1 Tema-problema e justificativa

O objetivo do trabalho intitulado Automação residencial via *smartphone* é desenvolver e implementar um sistema que gerencia consumo de energia e controle de acesso em ambientes residências, ressaltando a possibilidade de dimerização da iluminação. Tem como justificativa trazer maior conforto e comodidade para seus usuários e proporciona economia de energia elétrica. Outro benefício é a possibilidade de realizar esse controle independente da distância em que se encontra do local gerenciado.

Nos dias atuais a busca por conforto e economia torna-se cada vez mais presente na sociedade, com os avanços tecnológicos que possibilitam controles de subsistemas através da interação entre residência e usuário.

A partir do ponto destacado, é feita uma análise FOFA (Forças, Oportunidades, Fraquezas e Ameaças) para a viabilização do projeto.

Forças: muitas empresas fazem a instalação de equipamentos que são capazes de automatizar a residência, mas nenhuma oferece o baixo custo de e a facilidade de instalação. O projeto visa a eficiência energética e economia.

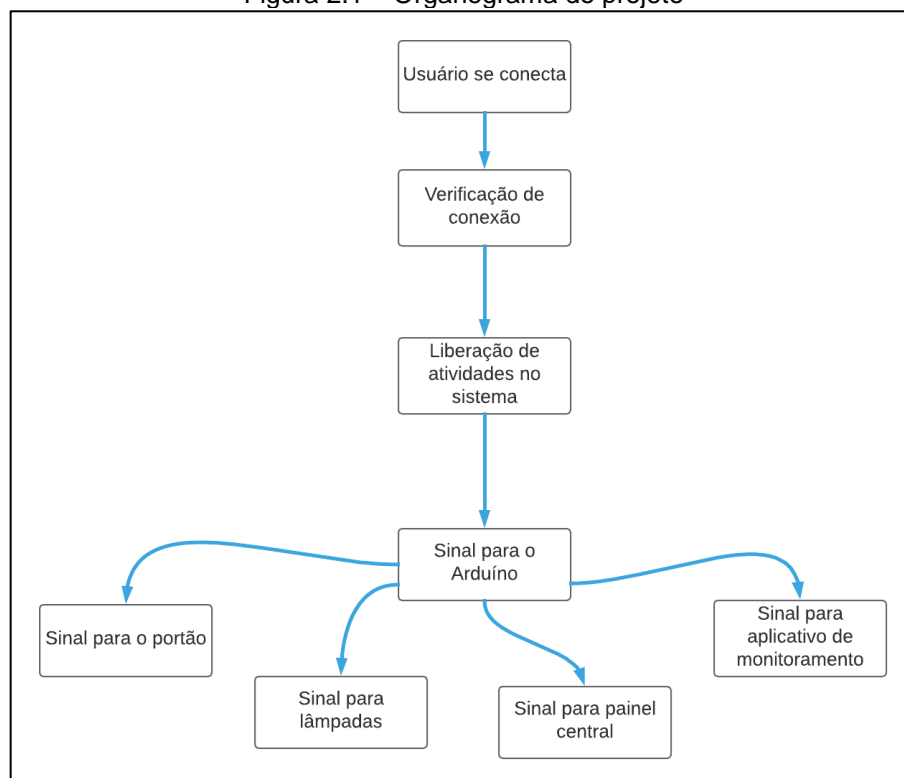
Oportunidades: o projeto tem a intenção de alcançar o público que tem algum tipo de dificuldade de locomoção entre os cômodos da residência ou quando a distância entre eles for muito grande, fazendo com que não seja necessário estar fisicamente no ambiente para apagar a luz do mesmo.

Fraquezas e ameaças: o controle não pode ser feito a longas distancias, devido à limitação do meio de comunicação de dados e o usuário deve portar um celular com tecnologia compatível com o software de controle.

2.2 Descrição do projeto através de organograma e fluxograma

Após a delimitação do tema e justificativa faz-se uma descrição do projeto através de um organograma e fluxograma. A Figura 2.1 ilustra o organograma e exemplifica a estrutura da operação.

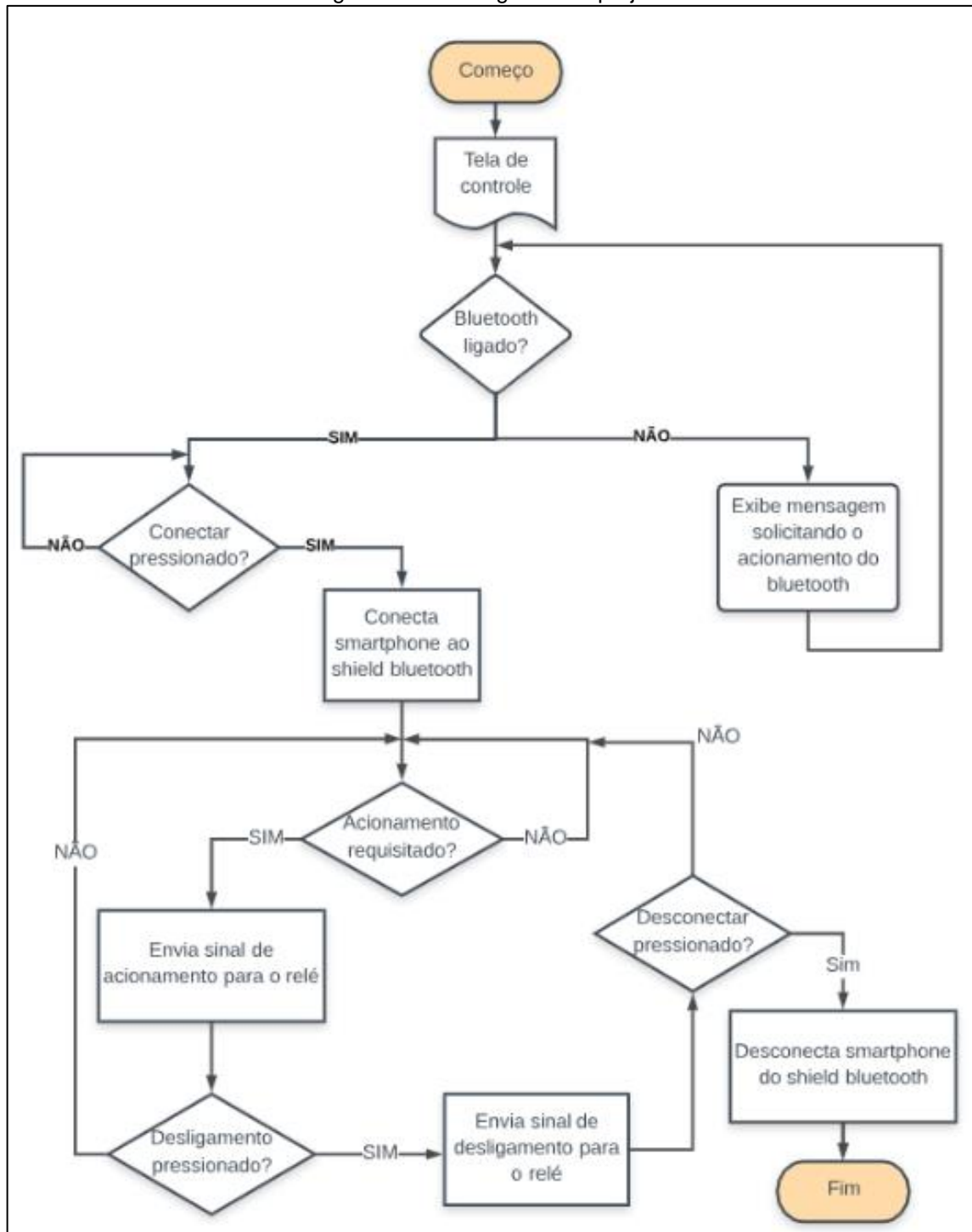
Figura 2.1 – Organograma do projeto



Fonte: Autoria própria, 2018

O organograma da Figura 2.1 consiste em exemplificar o projeto por meio de diagrama de blocos que apontam as etapas do funcionamento e ligações físicas. Tendo como diferencial um aplicativo para *smartphone* que controla todas as ações remotamente, gerenciadas pelo Arduino. Para melhor entendimento da lógica de funcionamento segue-se o fluxograma do projeto representado pela Figura 2.2.

Figura 2.2 – Fluxograma do projeto



Fonte: Autoria própria, 2018

A Figura 2.2 representa o fluxograma que descreve o funcionamento lógico do projeto, sendo as ações que serão realizadas indicadas em retângulos, os losangos representam a condição “SE” e têm duas vertentes, “SIM” ou “NÃO” e as bandeiras são as mensagens a serem exibidas na tela do *smartphone*. Inicialmente, será feito o teste de conexão do aplicativo com a central de comando, após confirmado que o mesmo se encontra conectado, o aplicativo apresenta uma tela que possibilita a escolha da ação a ser tomada no momento, tendo como opções: controle de luminosidade e controle de acesso, com possibilidade de melhoria futura o controle de temperatura.

2.3 Etapas teóricas e práticas para desenvolvimento do projeto

Após delimitar o tema-problema, justificativa e descrição do projeto através do organograma e fluxograma parte-se para as seguintes etapas:

Primeira etapa: reunião dos integrantes do grupo com o orientador para traçar as diretrizes de pesquisa e desenvolvimento do projeto. O orientador fez uma explanação geral sobre o trabalho e marcou obrigatoriamente um dia por semana para apresentar-lhe o andamento da pesquisa.

Segunda etapa: pesquisa bibliográfica com relação ao tema na biblioteca da Fatec, em sites especializados, catálogos de empresas. Seleção dessa pesquisa e construção do capítulo 1 – Fundamentação teórica.

Terceira etapa: levantamento dos materiais a serem usados no projeto. Pesquisa de preços em sites e lojas especializadas. Estudo da viabilidade econômica. Aquisição dos materiais, conforme Tabela 2.1

Tabela 2.1 – Materiais utilizados no projeto

Componentes	Quantidade	Preço(R\$)
Arduino UNO	1	40,00
Shield Bluetooth Arduino	1	26,00
Módulo 8 Relés 5 V	2	70,00
Placa conversora de protocolo	1	14,00
Motor para portão da maquete	1	12,00
Led's	10	2,00
Maquete	1	110,00
Caixa Steck com canaletas	1	63,00
Total		337,00

Fonte: Autoria própria, 2018

Quarta etapa: montagem da estrutura da maquete em madeira MDF, posicionamento dos led's nos cômodos e motor para a abertura do portão além do cabeamento necessário para o funcionamento.

Quinta etapa: montagem da caixa *steck* para o agrupamento dos componentes: Arduino, *shield bluetooth*, relés, e cabeamento, são utilizadas também canaletas e anilhas para facilitar a visualização e entendimento.

Sexta etapa: elaboração dos circuitos eletrônicos no software Proteus, e elaboração do aplicativo para *smartphone* no software App inventor.

Sétima etapa: programação da lógica no programa Arduino com base na programação do App inventor.

Oitava etapa: integração do aplicativo instalado *smartphone* com o Arduino e os relés para acionamento dos led's e do portão e testes de funcionamento.

Nona etapa: obstáculos e soluções encontradas durante o desenvolvimento do projeto.

Décima etapa: finalizado o projeto, fazem-se as considerações finais e o resumo do trabalho.

3 DESENVOLVIMENTO DO PROJETO

Neste capítulo encontra-se passo a passo o desenvolvimento e a construção do projeto, ilustrando-o com figuras para melhor entendimento. Mostra a ligação entre a teoria e o processo de montagem, bem como o seu funcionamento. Para melhor entendimento, a Figura 3.1 ilustra o projeto finalizado.

Figura 3.1 – Projeto finalizado



Fonte: Autoria própria, 2018

O projeto de automação residencial via *smartphone* visa a implementação de um sistema de controle que permita economia de energia remotamente, dispondo dos seguintes componentes:

- Arduíno UNO;
- *Shield Bluetooth*;
- Conversor RS-232/RS-485;
- Relés 5 V;
- Aplicativo (Sistema operacional Android);
- Conexões físicas de lâmpadas.

Para melhor visualização e entendimento da proposta de automação via celular, é elaborada uma maquete representativa. O Arduino juntamente com o *Shield Bluetooth*, é responsável pela comunicação entre o circuito e o *smartphone*,

permitindo o controle dos circuitos via aplicativo. Os relés são identificados por anilhas indicando o cômodo que controla, e eles compõem a interface para o acionamento das saídas, que são lâmpadas, além do motor para a abertura do portão da garagem.

A estrutura do desenvolvimento do projeto se encontra amparada nos seguintes tópicos:

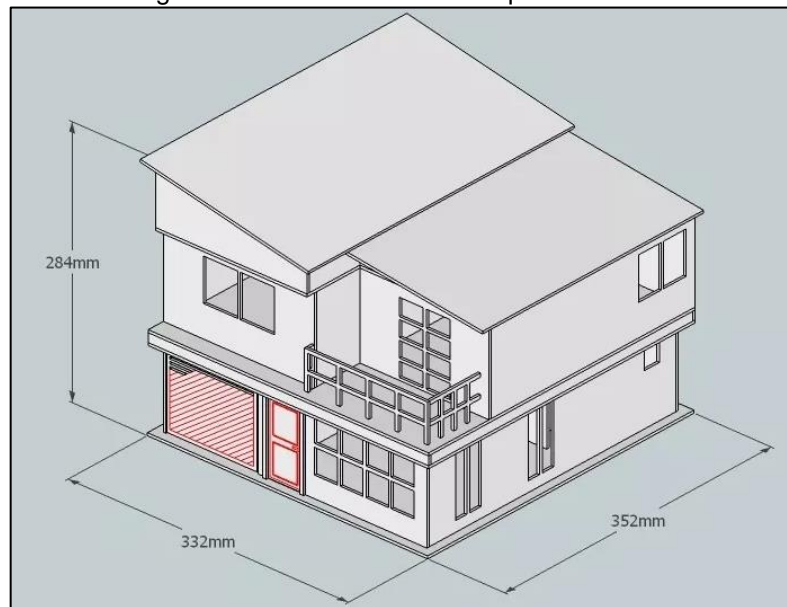
- construção da maquete em madeira MDF e caixa *steck*;
- montagem dos circuitos eletrônicos no software proteus;
- construção do aplicativo para *smartphone*;
- programação da lógica no programa Arduino;
- integração do aplicativo com o Arduino e teste de funcionamento;
- obstáculos encontrados e soluções.

3.1 Montagem da maquete e caixa *steck*

É montada uma maquete em MDF nas medidas 332 mm x 352 mm x 284 mm, respectivamente largura, comprimento e altura, para visualizar o funcionamento do projeto de automação residencial, na qual são utilizados led's brancos para simular lâmpadas dispostas nos devidos cômodos da residência. O acionamento dos led's são controlados através de relés 5 V que são requisitados pelo aplicativo no *smartphone* do usuário, e a comunicação é feita pelo *shield bluetooth* presente na caixa *steck* juntamente com o Arduino e os relés. Na parte inferior da maquete, estão presentes 4 cômodos denominados como: sala, garagem, dispensa, quintal, e na parte superior estão os cômodos: quarto 1, quarto 2 e banheiro.

A iluminação da sala é feita por 3 lâmpadas, e a dos demais cômodos por apenas uma. Na garagem também está presente o motor para abrir e fechar o portão, função esta que também é controlada pelo aplicativo no *smartphone* do usuário. A visualização interna da maquete está presente no Anexo A, e a Figura 3.2 mostra suas dimensões.

Figura 3.2 – Dimensões da maquete utilizada.



Fonte: Manual da montagem, 2018 p.03

Após a montagem da maquete dá-se início a montagem física da caixa *steck*. Para essa montagem usam-se os seguintes componentes: Arduino UNO, *shield Bluetooth*, relés 5 V, conexões físicas de lâmpadas. A Figura 3.3 mostra a caixa *steck* em sua montagem inicial.

Figura 3.3 – Caixa *steck* sem componentes

Fonte: Autoria própria, 2018.

A caixa *steck* tem como função organizar e distribuir os componentes e cabos do projeto, sendo uma central de monitoramento por meio da IHM e afim de facilitar a manutenção dos componentes utilizados. Primeiramente, o projeto é adequado a uma placa de aço na cor laranja seguindo os padrões da norma NR12. Para a montagem da caixa utiliza-se um trilho TS-35 de aço que é responsável pela fixação dos componentes na placa e canaletas de PVC para organizar e direcionar os cabos que se utilizam para efetuar as ligações físicas. A Figura 3.4 mostra a caixa *steck* finalizada.

Figura 3.4 - Caixa *steck* finalizada



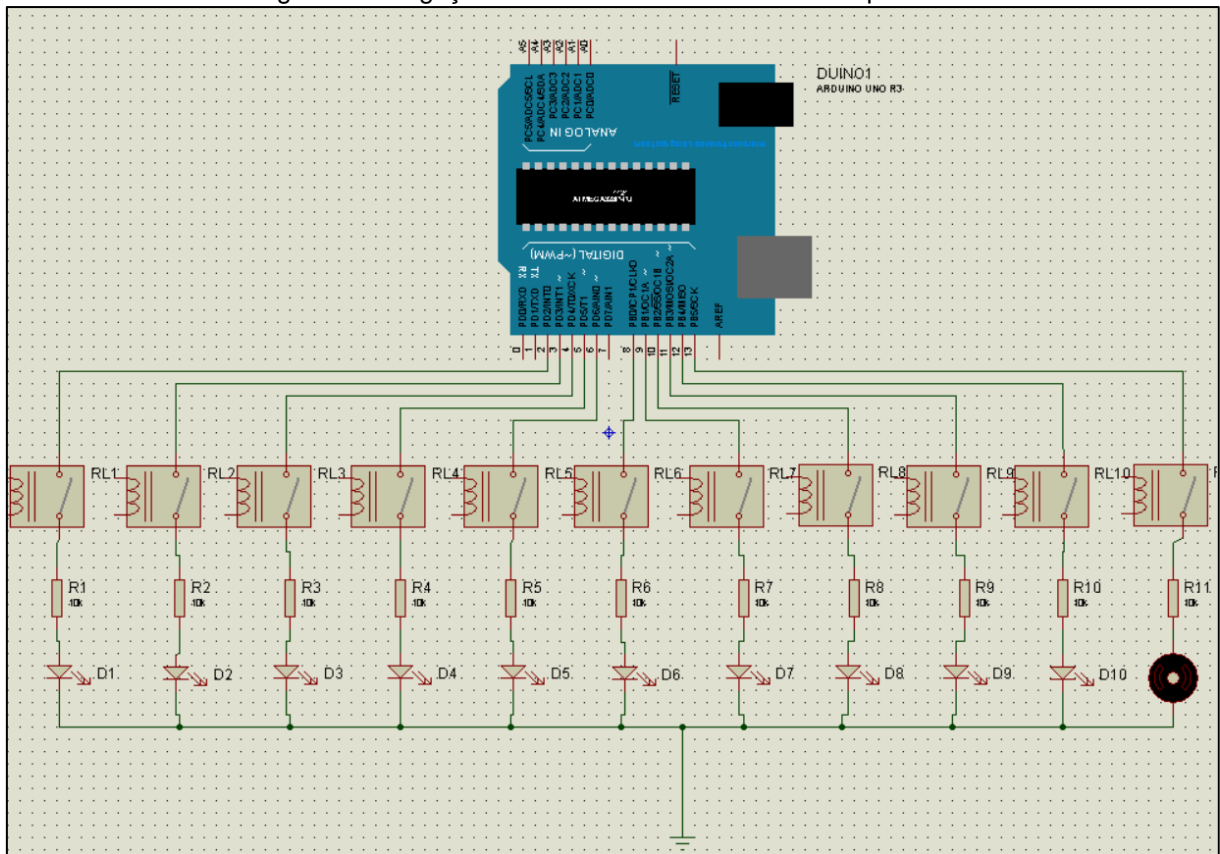
Fonte: Aatoria própria, 2018

O Arduino juntamente com o *Shield Bluetooth* são componentes responsáveis pela comunicação entre o circuito eletrônico e o *smartphone*, permitindo o controle dos circuitos via aplicativo. Para a composição da caixa *steck* faz-se uso de 11 relés 5 V que compõem a interface para o acionamento das saídas do Arduino que são led's ligados nos relés por meio de fios de cobre.

3.2 Montagem dos circuitos eletrônicos no software Proteus

Dando sequência, faz-se a montagem dos circuitos eletrônicos no software Proteus e simulação. Para a montagem do circuito responsável pela ligação elétrica no software proteus, usa-se a placa Arduino ligada em 11 relés, que representam os canais usados do *shield's* usados no projeto, 11 resistores de 1000 *ohm's* ligados nos contatos NA de cada relé e por fim 11 led's que representam as lâmpadas utilizadas na residência e um motor DC 5 V que é o responsável pela abertura e fechamento do portão da maquete. A Figura 3.5 mostra a ligação do circuito.

Figura 3.5 - Ligação do circuito elétrico no software proteus.



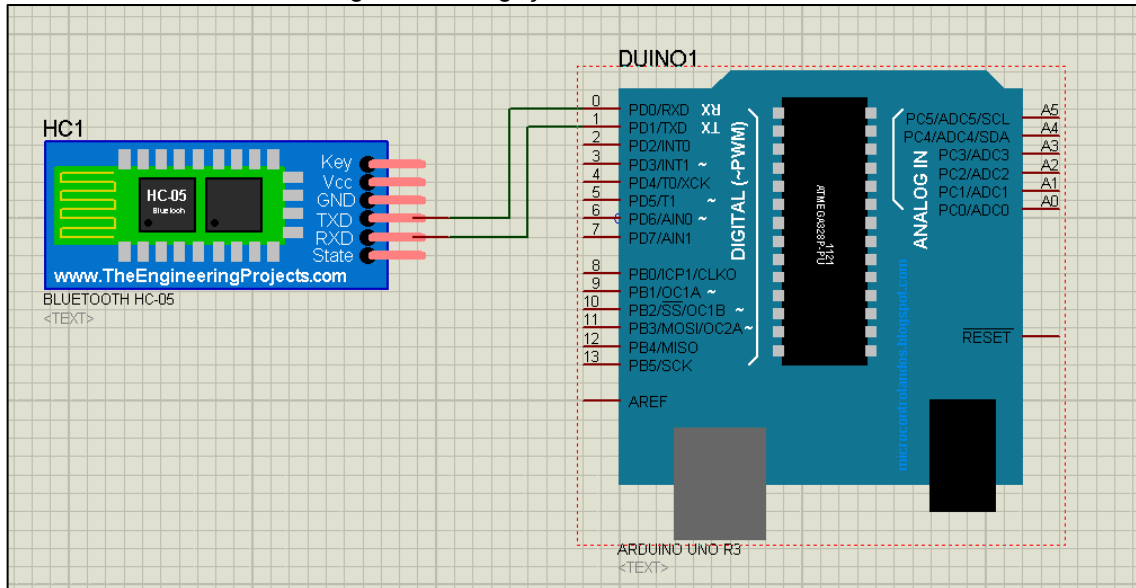
Fonte: Autoria própria, 2018

O circuito baseia-se na programação que é feita no *software* Arduino e é descarregada na placa microcontroladora Arduino. A programação informa qual será a entrada utilizada para o controle dos relés, usando apenas uma saída para acionar qualquer um dos 11 componentes, definindo qual será acionado por meio de um

signal recebido do usuário. O sinal é recebido através da comunicação *bluetooth* que é estabelecida com o aparelho *smartphone* do usuário.

A ligações necessárias para realizar a conexão *bluetooth* com o Arduino são feitas na própria placa e em um módulo *shield bluetooth* HC-06. Ainda no software proteus são feitas as ligações do *shield bluetooth* como é mostrado na Figura 3.6.

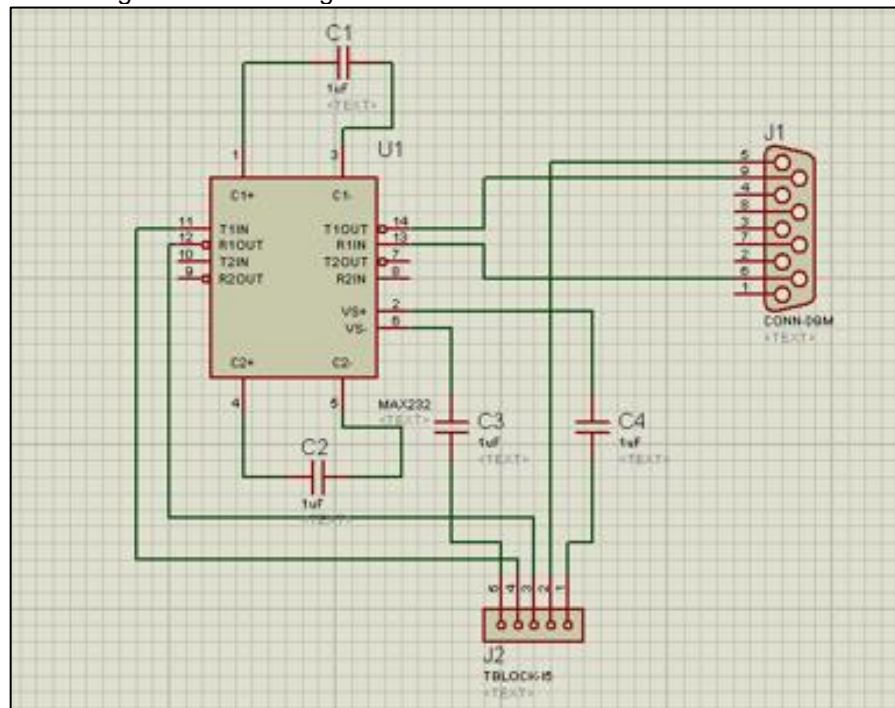
Figura 3.6 – Ligações do *shield* Bluetooth



Fonte: Autoria própria

Para que seja possível a comunicação entre Arduino e IHM é desenvolvido um circuito no software Proteus onde se realiza a conversão do protocolo RS-232 para RS-485. O circuito é confeccionado tendo como base o circuito integrado MAX232, que é o responsável pela conversão de protocolos. Os capacitores posicionados nos pinos 1, 2, 3, 4, 5 e 6 são especificados no *datasheet* do fabricante, contendo valores e encapsulamentos adequados para o funcionamento correto do circuito. Também é utilizado um terminal DB9 que é o responsável pela transferência de dados para o equipamento no novo protocolo. Para a interface de ligações entre o circuito integrado e o Arduino, é utilizado um *jumper* com terminais específicos utilizados pelo controlador, onde será ligado em suas respectivas entradas para possibilitar a conversão de protocolo. A Figura 3.7 mostra a montagem do circuito no software Proteus ISIS.

Figura 3.7 – Montagem do circuito conversor RS-232/RS-485



Fonte: Autoria própria, 2018.

3.3 Construção do aplicativo para *smartphone*

Dando continuidade, faz-se a construção do aplicativo disponível para *download* no *smartphone* para o sistema operacional Android, opta-se pela plataforma de desenvolvimento “AppInventor” para a construção do aplicativo. É elaborado o Menu de controle na tela *Designer* possibilitando o controle da iluminação dos cômodos desejados pelo usuário.

Os títulos presentes na tela “*Bluetooth*” e “*Lâmpadas*” são inseridos pela ferramenta “*label*” sendo *label1* e *label2* respectivamente. Para colocar os botões que fazem o acionamento dos leds, inicialmente é necessário a criação de blocos para o posicionamento dos mesmos, estes blocos são denominados como “*verticalarrangement*” ou “*horizontalarrangement*” dependendo da maneira que se quer dispor os botões. Além dos botões de controle da iluminação, são inseridos botões para a comunicação do aplicativo com o *shield bluetooth* sendo eles: conectar e desconectar. Os botões de controle assim como os de conexão, são inseridos através da função “*button*”. A Figura 3.8 mostra o menu do aplicativo finalizado.

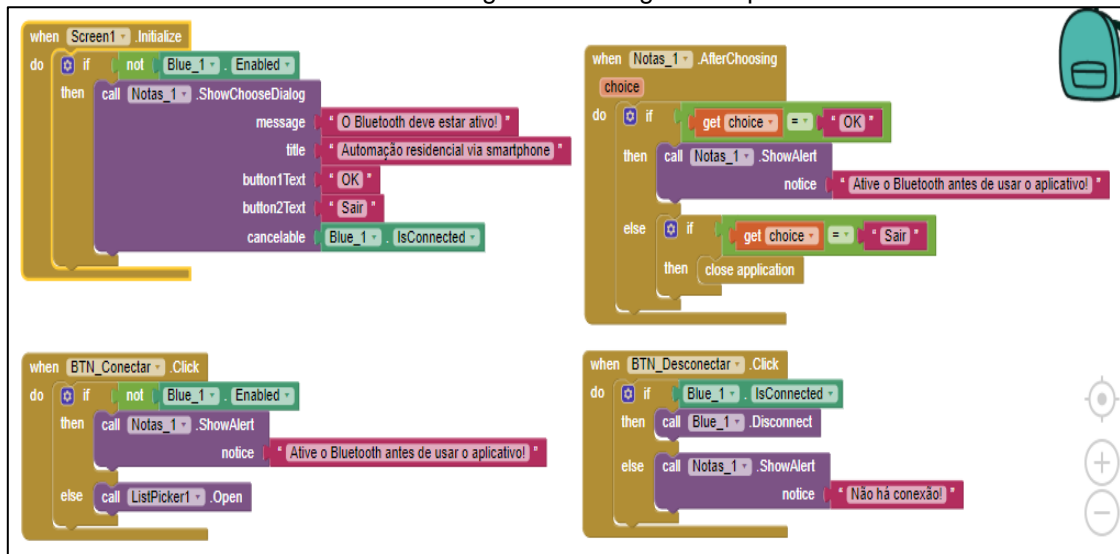
Figura 3.8 – Menu do aplicativo



Fonte: Autoria própria, 2018

Finalizado o Menu dá-se início a construção da lógica de funcionamento do aplicativo, a partir da tela *Blocks*. Inicialmente, através da função “*initialize*” é exposto do Menu de controle e em seguida verifica se a conexão *bluetooth* está ocorrendo, de modo que, seja exibida uma instrução para o acionamento, caso a função esteja desligada. Também é elaborada a lógica dos botões de ligar e desligar a conexão através da função “*when.click*” conforme a Figura 3.9.

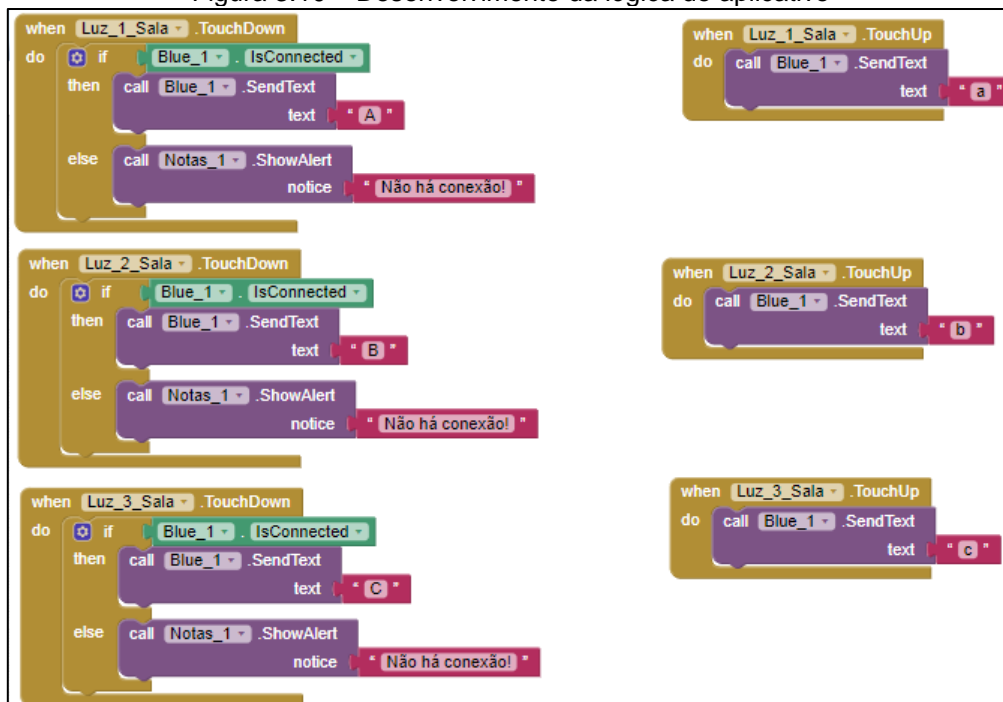
Figura 3.9 – Lógica do aplicativo



Fonte: Autoria própria, 2018

Em seguida são programadas as entradas que são lidas pelo *shield bluetooth* para que sejam executadas as funções que são requisitadas através do aplicativo. A indicação “*Touchdown*” refere-se a o estado pressionado do botão, no qual é ligada uma saída para o acionamento de um led representando na maquete uma lâmpada. Através da função “*showalert*” são exibidas mensagens indicando que a conexão *Bluetooth* não está ocorrendo, se for o caso, conforme ilustra a Figura 3.10.

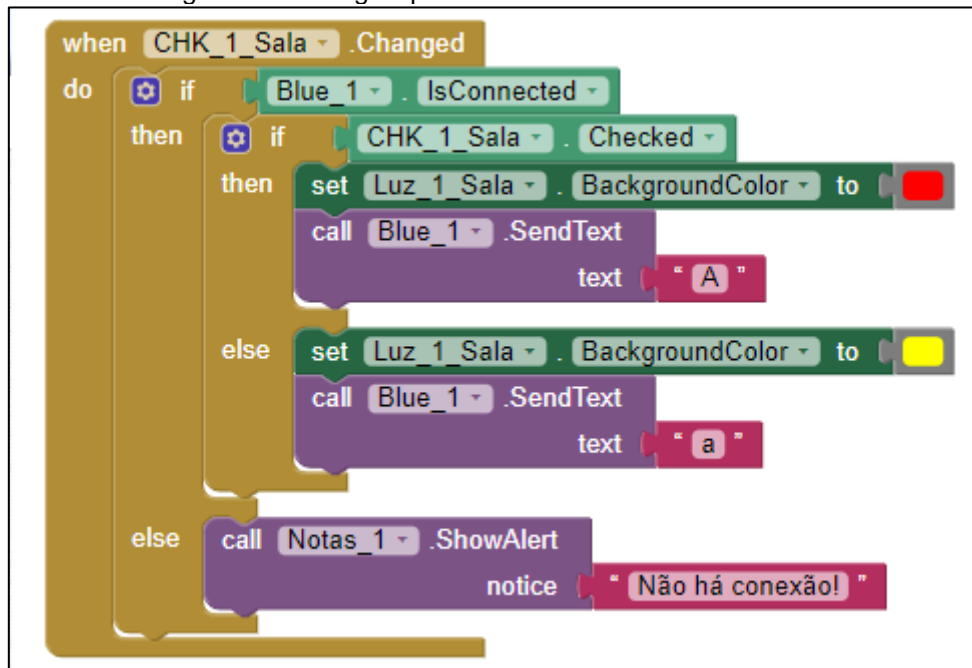
Figura 3.10 – Desenvolvimento da lógica do aplicativo



Fonte: Autoria própria, 2018

Com o objetivo de melhorar a visualização do funcionamento e do controle dos led's é elaborada uma lógica através da função "*when.Changed*" que muda a cor de fundo das teclas de comando do menu para a cor vermelha, padrão este que é aplicado em todos os cômodos da residência, conforme ilustra a Figura 3.11.

Figura 3.11 – Lógica para efeito visual de funcionamento

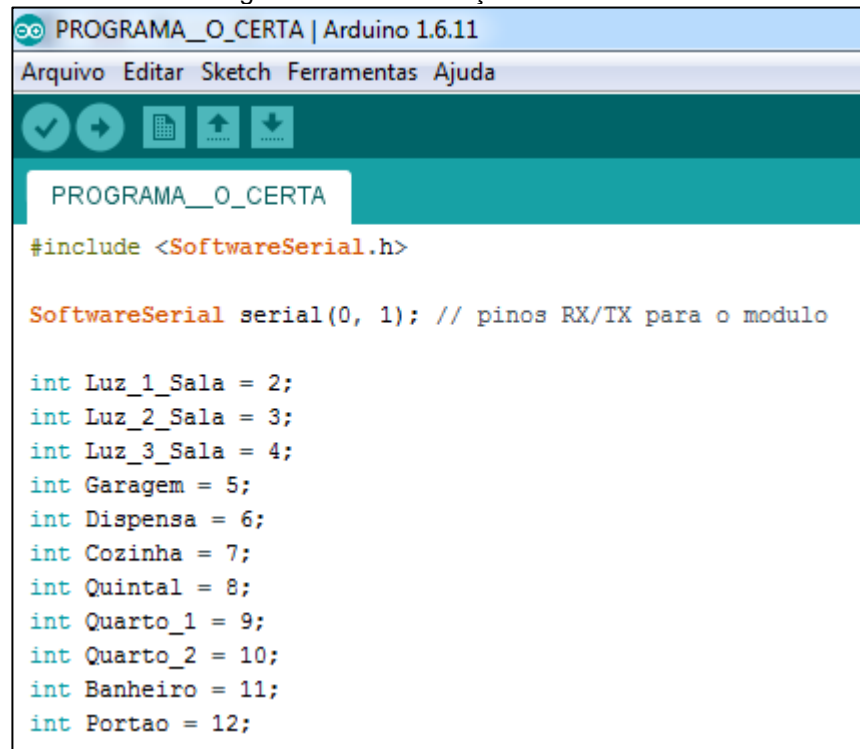


Fonte: Autoria própria, 2018

3.4 Programação da lógica no programa Arduino

É elaborada a lógica de funcionamento do projeto no programa "Arduino". A programação é responsável por declarar as variáveis de controle e estruturar o funcionamento do projeto. Na parte superior do programa são declaradas as variáveis que podem ser controladas em qualquer momento em que o programa esteja funcionando, sendo estas, os cômodos da residência em cada pino do Arduino declaradas nos pinos de 2 até 12. A Figura 3.12 demonstra a declaração de variáveis.

Figura 3.12 – Declaração de variáveis



```
PROGRAMA_O_CERTA | Arduino 1.6.11
Arquivo Editar Sketch Ferramentas Ajuda

PROGRAMA_O_CERTA
#include <SoftwareSerial.h>

SoftwareSerial serial(0, 1); // pinos RX/TX para o modulo

int Luz_1_Sala = 2;
int Luz_2_Sala = 3;
int Luz_3_Sala = 4;
int Garagem = 5;
int Dispensa = 6;
int Cozinha = 7;
int Quintal = 8;
int Quarto_1 = 9;
int Quarto_2 = 10;
int Banheiro = 11;
int Portao = 12;
```

Fonte: Autoria própria, 2018

Em seguida, configuram-se as variáveis dos pinos como saídas, para isso utiliza-se a função *pinmode* como *output* escrito em “void setup”. Nesta etapa são programadas as informações que necessitam ser lidas apenas uma vez. Para o funcionamento do programa cria-se uma variável do tipo “*char*” que lê caracteres e a mesma é representada pela palavra “caracter”.

Na etapa “*void loop*” são programadas as funções de controle de saídas, de modo que cada cômodo recebe uma letra maiúscula para o acionamento da saída através da função “*HIGH*”, e minúscula para o desligamento, através da função “*LOW*”. Para que o acionamento das lâmpadas funcione corretamente é necessário que as letras designadas para os cômodos no programa Arduino sejam as mesmas da programação elaborada no “Appinventor”. A programação completa do Arduino encontra-se no Apêndice A.

3.5 Construção do software para IHM e integração com Arduino

É elaborado um software que se utiliza em um aparelho IHM (interface homem-máquina) onde é possível visualizar quais são as lâmpadas da casa que estão acessas que é colocado junto a caixa *steck* para servir como central de gerenciamento de energia. Opta-se pelo desenvolvimento do software na aplicação *EasyBuilder*, ferramenta que é disponibilizada para *download* gratuito na internet pelo fabricante da IHM.

O software contém apenas uma tela onde se vê onze indicadores que representam as lâmpadas de cada cômodo e o motor do portão. Quando é enviado um sinal para ligar uma lâmpada ou motor, seu respectivo indicador na tela do software encontra-se acionado. A Figura 3.13 mostra a tela do software sem nenhum acionamento.

Figura 3.13 - Tela do software da IHM

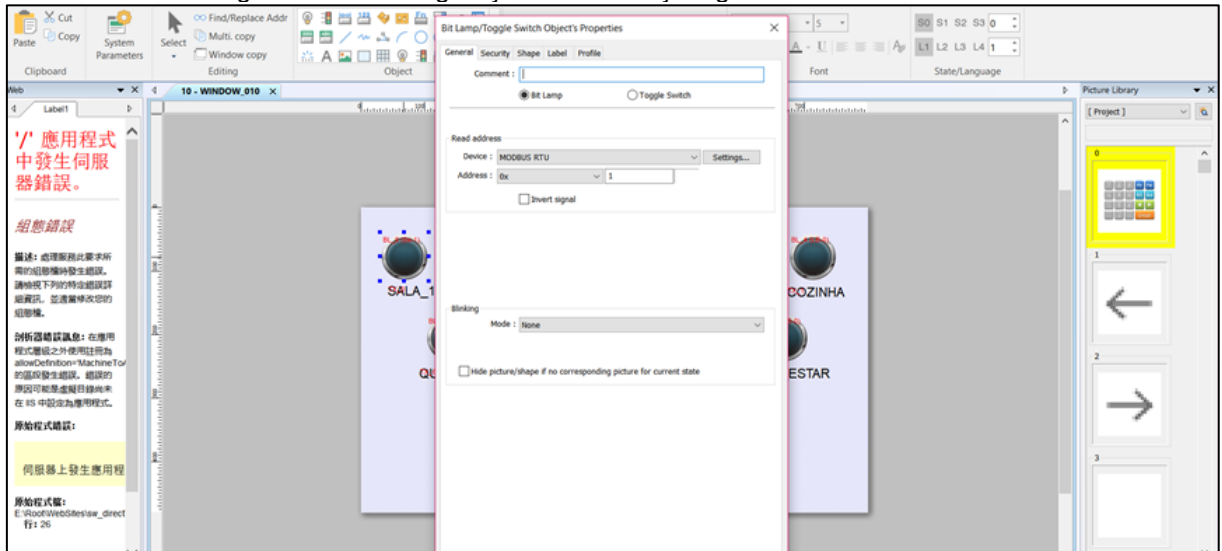


Fonte: Autoria própria,2018

Para a montagem dessa tela é criado um projeto em branco no aplicativo e em seguida é selecionado na aba "*Object*" o item "*Bit Lamp*" e adicionado onze vezes na tela para representar as lâmpadas dos cômodos e o motor da garagem. Após a configuração gráfica concluída faz-se o endereçamento de cada *Bit Lamp*

com um endereço lógico da sua respectiva porta de dados do Arduino. Para isso é selecionado um *Bit Lamp* de cada vez e na aba "Atributtes" é colocado o endereço lógico usado na programação do Arduino. A Figura 3.14 mostra a configuração da porta lógica da lâmpada da sala.

Figura 3.14 - Configuração do endereço lógico do software da IHM



Fonte: Autoria própria, 2018

Após a realização desse procedimento em todos os *Bit Lamp*, o software está pronto para se comunicar com o Arduino, tanto para receber como para enviar dados.

3.7 Obstáculos encontrados e soluções

Durante o desenvolvimento e construção do projeto surgiram alguns obstáculos que foram solucionados mediante pesquisas, conhecimentos adquiridos na vida profissional dos integrantes do grupo, consultas ao orientador e professores na área.

Obstáculo 1: o controle do portão da garagem necessita de uma lógica diferente das lâmpadas, pois é uma saída que precisa ser desligada após ter chegado ao final do percurso.

Solução: é adicionado um componente eletrônico que mecanicamente movimenta o portão por um determinado tempo que é pré-definido na programação do Arduino.

Obstáculo 2: o espaço disponibilizado no telhado da maquete não suporta a quantidade de componentes necessários para o controle iluminação e circuito de comunicação.

Solução: é adquirida uma caixa *steck* para o posicionamento de todos os dispositivos que são responsáveis pelo controle de potência do projeto e os componentes responsáveis pela comunicação permanecem no telhado da maquete.

Obstáculo 3: a comunicação entre IHM e Arduino não são feitas no mesmo protocolo.

Solução: é desenvolvido um circuito capaz de converter o protocolo de comunicação do Arduino para que seja possível a comunicação com IHM. O circuito é montado no *software* Proteus e é transferido para uma placa de cobre virgem com os componentes necessários.

Obstáculo 4: o *shield bluetooth* e a placa conversora de protocolo utilizam os pinos RX/TX do Arduino.

Solução: foi pesquisado no *datasheet* do *shield bluetooth* um modo de comunicação e foi encontrada a solução usando o mesmo em modo *slave* sendo necessário duas saídas.

Obstáculo 5: dificuldade na programação do Arduino, montagem do aplicativo e na confecção da placa de comunicação de protocolos devido ao fato de que alguns conhecimentos não são aplicados durante o período de aula.

Solução: com pesquisas feitas na internet e com professores fora do horário de aula, as dúvidas e dificuldades encontradas nesse item foram sanadas por completo.

CONSIDERAÇÕES FINAIS

O objetivo deste trabalho que se intitula Automação Residencial Via Smartphone é desenvolver e implementar um sistema que gerencia consumo de energia e controle de acesso em ambientes residenciais. Justifica-se por trazer maior conforto e comodidade para seus usuários como ligar/desligar a luz de algum ambiente da casa, sem ter que se locomover até o interruptor físico para isso, além de proporcionar economia de energia elétrica. Outro benefício é a possibilidade de realizar esse controle independente da distância em que se encontra do local gerenciado.

Para o desenvolvimento do projeto fez-se o uso das teorias que envolvem caixa *steck*, Arduíno, *shields bluetooth* e relé, canaletas, trilhos, motor DC, cabos e uma IHM. O monitoramento do projeto é feito pelo celular do usuário ou pela IHM (Interface Homem-Máquina), que informam o estado atual dos leds que representam lâmpadas da residência.

Os métodos e técnicas obtidos pela metodologia científica dão suporte para organizar e planejar as etapas que direcionam o caminho para o desenvolvimento do projeto.

O usuário utiliza um aplicativo desenvolvido através da plataforma “AppInventor”, criado para o sistema operacional Android, no aplicativo contém um menu, que controla a iluminação dos cômodos e os acionamentos em geral da residência.

A comunicação entre o smartphone e o circuito é realizada através do Arduíno juntamente com o *Shield Bluetooth*, onde o usuário encaminha o comando desejado via *Bluetooth*. O Arduíno recebe o comando e posteriormente aciona os reles correspondentes a solicitação do usuário.

O maior ganho do projeto sobre a automação residencial é o conforto e a comodidade que proporcionam para seus usuários, possibilitando acesso remoto as funcionalidades da sua residência, independente do uso de uma rede de *wi-fi*.

Além disso o projeto trouxe uma visão mais ampla sobre o uso da automação, onde é possível aplica-la em atividades relacionadas eficiência energética, trazendo economia de energia conhecimentos para os integrantes do grupo como a funcionalidade da comunicação entre dois protocolos diferentes,

Durante o desenvolvimento do projeto deparou-se com alguns transtornos como a dificuldade no desenvolvimento do aplicativo para *smartphone*, comunicação da IHM com o Arduíno e o espaço reduzido para armazenar os componentes do projeto.

Como melhoria para futuros trabalhos, fica como sugestão a introdução de um intertravamento feito com a comunicação *bluetooth* e os interruptores da residência, podendo acionar as lâmpadas e demais equipamentos tanto pelo aplicativo como fisicamente.

REFERÊNCIAS

- ALECRIM, E. **Um pouco da história do wi-fi**. 2008. Disponível em: <<http://www.infowester.com/wifi.php>>. Acesso em: 20 mar. 2018.
- ARDUINO. **Arduino**. 2018. Disponível em: <<https://www.arduino.cc/>>. Acesso em: 04 abr. 2018.
- BORGES, F. **Redes de comunicação industrial**. 2007. Disponível em: <https://www.schneider-electric.pt/documents/product-services/training/doctecnico_redes.pdf>. Acesso em 24 abr. 2018.
- BRASIL, INSTITUTO NACIONAL DE METROLOGIA, QUALIDADE E TECNOLOGIA - INMETRO de 2015. **Lâmpadas de led**. Disponível em: <<http://inmetro.gov.br/inovacao/publicacoes/cartilhas/lampada-led/lampadaled.pdf>>. Acesso em: 13 mar. 2018.
- CALDIÉRI, M. R. **Implementação do modbus para aplicação em sistema de controle via rede sem fio**. 2016. Disponível em: <https://repositorio.unesp.br/bitstream/handle/11449/146695/caldieri_mr_me_bauru.pdf?sequence=3>. Acesso em 24 abr. 2018.
- COUTINHO, G. L. **A era dos smartphones: um estudo exploratório sobre o uso de smartphones no Brasil**. 2014. Disponível em: <http://bdm.unb.br/bitstream/10483/9405/1/2014_GustavoLeuzingerCoutinho.pdf>. Acesso em: 12 mar. 2018.
- FERREIRA, F. N. F. **Segurança da informação**. 1. ed. Rio de Janeiro: Editora Ciência Moderna, 2003.
- FREITAS, C. M. **Protocolo Modbus: fundamentos e aplicações**. 2014. Disponível em <<https://www.embarcados.com.br/protocolo-modbus>>. Acesso em: 24 abr. 2018.
- GISLASON, D. **Zigbee wireless networking**. 2007. Disponível em: <<https://www.waveshare.com/w/upload/4/4b/Zigbee-Wireless-Networking.pdf>>. Acesso em: 20 mar. 2018.
- INDUSTRIA DE TELECOMUNICAÇÃO ELETRÔNICA BRASILEIRA - INTELBRAS, **Controle de acesso**. 2018. Disponível em: <http://www.intelbras.com.br/sites/default/files/downloads/datasheet_a4_iaut_fx_2000.pdf>. Acesso em: 07 Mai. 2018.
- IRVINE, Kip R. **Assembly language for x86 processors**. 6. ed. Florida: Pearson Education. 2007.

MURATORI, J. R. **Automação residencial: histórico, definições e conceitos**. 2011. Disponível em: <http://www.osetoreletrico.com.br/wp-content/uploads/2011/04/Ed62_fasc_automacao_capl.pdf>. Acesso em: 12 mar. 2018.

NÉRIO, A. 2003. **Wireless**. Disponível em:<http://www.logicengenharia.com.br/mcamara/ALUNOS/Sem_fio.PDF>. Acesso em: 10 abr. 2018.

OTTON. **Redes**. 2009. Disponível em: <<http://www2.ic.uff.br/~otton/graduacao/informatical/redes.pdf>>. Acesso em: 20 mar. 2018.

PENIDO, E. C. C.; TRINDADE, R. S. 2013. **Microcontroladores**. Disponível em: <<https://www2.ifmg.edu.br/ceadop3/apostilas/microcontroladores>>. Acesso em: 10 abr. 2018.

PRESSMAN, R. S. **Engenharia de software: uma abordagem profissional**. 7. ed. Porto Alegre: Bookman, 2011.

SENA, D. C. S. **Automação residencial**. Projeto de Graduação. Centro Tecnológico da Universidade Federal do Espírito Santo, Vitória: UFES, 2005.

SEVERINO, A. J. **Metodologia do trabalho científico**. 23. ed. São Paulo: Cortez, 2007.

SCHULTE, R. G. **Escalonamento no sistema operacional android**. 2016. Disponível em: <http://www.olaria.ucpel.tche.br/soii/lib/exe/fetch.php?media=rafael_gouvea_schulte.pdf>. Acesso em: 04 abr. 2018.

SOUZA, D.J. **Desbravando o PIC**. 8. ed. São Paulo: Érica, 2005.

SOUZA, V. A. **Protocolo modbus**. 2012. Disponível em: < <http://www.cerne-tec.com.br/Modbus.pdf> >. Acesso em: 24 abr.2018.

VIDAL, F. J. T.; VILELA, P. S. C. **Profibus PA e profibus DP**. 2003. Disponível em: <https://www.dca.ufrn.br/~affonso/FTP/DCA447/trabalho3/trabalho3_5.pdf>. Acesso em: 07 Mai. 2018.

APÊNDICE

APÊNDICE A – PROGRAMAÇÃO DO ARDUÍNO

```
#include <SoftwareSerial.h>

SoftwareSerial serial(0, 1); // pinos RX/TX para o modulo

int Luz_1_Sala = 2;

int Luz_2_Sala = 3;

int Luz_3_Sala = 4;

int Garagem = 5;

int Dispensa = 6;

int Cozinha = 7;

int Quintal = 8;

int Quarto_1 = 9;

int Quarto_2 = 10;

int Banheiro = 11;

int Portao = 12;

void setup()

{

Serial.begin(9600); //porta serial (hardware) para monitorar os caracteres enviados

serial.begin(9600); // inicializacao da porta serial para o modulo

pinMode(2, OUTPUT);
```

```
pinMode(3, OUTPUT);

pinMode(4, OUTPUT);

pinMode(5, OUTPUT);

pinMode(6, OUTPUT);

pinMode(7, OUTPUT);

pinMode(8, OUTPUT);

pinMode(9, OUTPUT);

pinMode(10, OUTPUT);

pinMode(11, OUTPUT);

pinMode(12, OUTPUT);

}

void loop()

{

if (serial.available()){ // Se a serial estiver disponível, tem caractere para ler

char caracter = softSerial.read(); // salva o caractere lido na variável 'a'

Serial.write(caracter); // escreve o caractere recebido via bluetooth na serial do PC
para conferencia

//Checagem/teste de valores

if (caracter == 'A') digitalWrite(Luz_1_Sala, HIGH);

//Checagem/teste de valores

if (caracter == 'a') digitalWrite(Luz_1_Sala, LOW);
```

```
//Checagem/teste de valores  
  
if (caracter == 'B') digitalWrite(Luz_2_Sala, HIGH);  
  
//Checagem/teste de valores  
  
if (caracter == 'b') digitalWrite(Luz_2_Sala, LOW);  
  
  
//Checagem/teste de valores  
  
if (caracter == 'C') digitalWrite(Luz_3_Sala, HIGH);  
  
//Checagem/teste de valores  
  
if (caracter == 'c') digitalWrite(Luz_3_Sala, LOW);  
  
  
//Checagem/teste de valores  
  
if (caracter == 'D') digitalWrite(Garagem, HIGH);  
  
//Checagem/teste de valores  
  
if (caracter == 'd') digitalWrite(Garagem, LOW);  
  
  
//Checagem/teste de valores  
  
if (caracter == 'E') digitalWrite(Dispensa, HIGH);  
  
//Checagem/teste de valores  
  
if (caracter == 'e') digitalWrite(Dispensa, LOW);  
  
  
//Checagem/teste de valores  
  
if (caracter == 'F') digitalWrite(Cozinha, HIGH);
```

```
//Checagem/teste de valores
```

```
if (caracter == 'f') digitalWrite(Cozinha, LOW);
```

```
//Checagem/teste de valores
```

```
if (caracter == 'G') digitalWrite(Quintal, HIGH);
```

```
//Checagem/teste de valores
```

```
if (caracter == 'g') digitalWrite(Quintal, LOW);
```

```
//Checagem/teste de valores
```

```
if (caracter == 'H') digitalWrite(Quarto_1, HIGH);
```

```
//Checagem/teste de valores
```

```
if (caracter == 'h') digitalWrite(Quarto_1, LOW);
```

```
//Checagem/teste de valores
```

```
if (caracter == 'I') digitalWrite(Quarto_2, HIGH);
```

```
//Checagem/teste de valores
```

```
if (caracter == 'i') digitalWrite(Quarto_2, LOW);
```

```
//Checagem/teste de valores
```

```
if (caracter == 'J') digitalWrite(Banheiro, HIGH);
```

```
//Checagem/teste de valores
```

```
if (caracter == 'j') digitalWrite(Banheiro, LOW);
```

```
//Checagem/teste de valores  
  
if (caracter == 'K') digitalWrite(Portao, HIGH);  
  
//Checagem/teste de valores  
  
if (caracter == 'k') digitalWrite(Portao, LOW);  
  
}  
  
}
```

ANEXO

ANEXO A – DISPOSIÇÃO DOS CÔMODO DA MAQUETE

