

**UNIVERSIDADE ESTADUAL DO RIO GRANDE DO SUL
UNIDADE UNIVERSITÁRIA PORTO ALEGRE
CURSO SUPERIOR DE TECNOLOGIA EM AUTOMAÇÃO INDUSTRIAL**

ARNALDO SOARES LEDERHANS

**APLICAÇÃO DE HARDWARE DE BAIXO CUSTO E APLICATIVO PARA
SMARTPHONE EM AUTOMAÇÃO RESIDENCIAL**

PORTO ALEGRE

2019

ARNALDO SOARES LEDERHANS

**APLICAÇÃO DE HARDWARE DE BAIXO CUSTO E APLICATIVO PARA
SMARTPHONE EM AUTOMAÇÃO RESIDENCIAL**

Monografia a apresentada como requisito parcial para obtenção do título de Tecnólogo em Automação Industrial na Universidade Estadual do Rio Grande do Sul.

Orientador: Prof. Dr. André Borin Soares

PORTO ALEGRE

2019

ARNALDO SOARES LEDERHANS

**APLICAÇÃO DE HARDWARE DE BAIXO CUSTO E APLICATIVO PARA
SMARTPHONE EM AUTOMAÇÃO RESIDENCIAL**

Monografia a apresentada como requisito parcial para obtenção do título de Tecnólogo em Automação Industrial na Universidade Estadual do Rio Grande do Sul.

Orientador: Prof. Dr. André Borin Soares.

Aprovado em...../...../.....

BANCA EXAMINADORA

Orientador: Prof. Dr. André Borin Soares
Universidade Estadual do Rio Grande do Sul – UERGS

Prof. Dr. Luiz Fernando Gonçalves
Universidade Estadual do Rio Grande do Sul – UERGS

Prof. Dr. Nei Marçal
Universidade Estadual do Rio Grande do Sul – UERGS

PORTO ALEGRE

2019

Dedico este trabalho a Deus que me proporcionou o dom de aprender, para que eu possa replicar todo conhecimento aprendido e a meus filhos, que são minha inspiração para a busca do conhecimento.

AGRADECIMENTOS

Agradeço, acima de tudo, a Deus, que me concedeu a graça de finalizar mais um desafio, ajudando a enfrentar os obstáculos diários e a encontrar as respostas que parecem sem solução.

A minha mãe, pelo apoio e pelo incentivo, tanto pessoal quanto profissional, pela profissão que escolhi.

A minha esposa Fernanda, pelo companheirismo e pela compreensão de sempre e aos amigos e colegas de curso pela união prestada para alcançar o objetivo conjunto da aprovação.

Aos meus filhos Maria Eduarda, João Gabriel e Ana Clara, pela paciência e esperar um pouco mais para brincar e a entender que o objetivo final foi sempre dar o melhor para eles.

Ao professor e orientador Prof. Dr. André Borin, que com suas orientações, me proporcionou a realização deste trabalho.

RESUMO

Domótica é a automatização e o controle aplicados às residências e já existe há muitos anos no mercado, no entanto, ainda é considerada um artigo de luxo pela maioria dos consumidores. A falta de padronização dos equipamentos e das estruturas das instalações elétricas afetam diretamente o custo de implementação dos projetos. Sendo assim, o presente trabalho busca apresentar uma solução de automação a partir da interligação de hardwares pela rede *Wireless Fidelity (Wi-Fi)* e que seja acessível em termos de custo. O sistema é composto por sensores (movimento, gás, chama, umidade, temperatura, chuva e luminosidade) e controladores, que são administrados por meio de um aplicativo móvel para o sistema Android, utilizando a comunicação *wireless*. Além disso, são utilizados módulos de prototipagem Arduino interligados com controladores ESP8266 através da rede Wi-Fi e gerenciados pelo aplicativo e controlados por uma central que é o próprio dispositivo móvel do usuário. Para este projeto, foi desenvolvido um circuito elétrico com fonte e relé utilizável por controles, utilizando o ESP8266. Assim, é possível realizar a interligação do mesmo com a rede elétrica sem a necessidade de cabeamento de dados ou alteração da ligação elétrica já existente e do software para os controladores (responsável pelo controle das informações no sistema de automação residencial). Foi desenvolvido um aplicativo para sistema Android em plataforma livre, com o intuito de promover uma otimização do custo e do projeto com base nas necessidades de cada usuário. O projeto final permite que o sistema seja ampliado para outras finalidades, conforme as necessidades específicas de cada usuário e de quais variáveis se deseja monitorar e controlar.

Palavras-chave: Automação Residencial. Android. Arduino. Controlador. Wi-Fi

ABSTRACT

Home automation is the automation and control applied to residences and has been around for many years in the market, however, it is still considered a luxury item by most consumers. The lack of standardization of the equipment and the structures of the electrical installations directly affect the cost of implementing the projects. Therefore, the present work presents an automation solution from the interconnection of hardware through the Wi-Fi network and that is affordable in terms of cost. The system consists of sensors (motion, gas, flame, humidity, temperature, rain and light) and controllers, which are managed through a mobile application to the Android system using wireless communication. In addition, Arduino prototype modules are interconnected with ESP8266 controllers over the Wi-Fi network and managed by the application and controlled by a central unit that is the user's own mobile device. For this project, an electric circuit with source and relay usable by controls was developed, using ESP8266. Thus, it is possible to interconnect it with the electrical network without the need for cabling or alteration of the existing electrical connection and the software for the controllers (responsible for controlling the information in the residential automation system). An application for Android system was developed in free platform, with the aim of promoting cost and design optimization based on the needs of each user. The final design allows the system to be extended to other purposes, according to the specific needs of each user and which variables to monitor and control.

Keywords: Home automation. Arduino. Controller. Wi-Fi.

LISTA DE ILUSTRAÇÕES

Figura 1 - Domótica, seus componentes e sua relação com outras ciências	17
Figura 2 - Investimento realizado em domótica	17
Figura 3 - Placa Arduino Uno	23
Figura 4 - Exemplos de módulos periféricos Arduino	24
Figura 5 - Tela do <i>software</i> de programação do IDE do Arduino	25
Figura 6 - Arquitetura do microcontrolador ESP8266	27
Figura 7 - Descrição dos pinos do ESP8266	28
Figura 8 - Interface do programa MIT APP INVENTOR 2	30
Figura 9 - Diagrama da arquitetura do projeto	33
Figura 10 - Esquemático da placa de controle	34
Figura 11 - Esquema de ligação do microcontrolador ESP-01 ao Arduino Uno	34
Figura 12 - Fluxograma do <i>software</i>	36
Figura 13 - Placa de controle	37
Figura 14 - Prototipagem do circuito de controle da iluminação	39
Figura 15 - Circuito de controle de temperatura	40
Figura 16 - Esquemático do circuito de detecção de gás e chama	40
Figura 17 - Local onde são desenvolvidas as telas	42
Figura 18 - Índice disponível para criação da tela	43
Figura 19 - Tela inicial do aplicativo	44
Figura 20 - Tela do sistema de iluminação	45
Figura 21 - Tela do sistema de controle de temperatura	46
Figura 22 - Tela do sistema de irrigação	46
Figura 23 - Tela do sistema de alarme de gás e fogo	47
Figura 24 - Tela do controle das cortinas	48
Figura 25 - Tela do sistema de tomada inteligente	48
Figura 26 - Editor dos blocos	49
Figura 27 - Ilustração de um trecho do programa da tela inicial	50
Figura 28 - Opções de compilação do aplicativo	51
Figura 29 - Compilação o aplicativo	51
Figura 30 - QR Code gerado pelo compilador	52

Figura 31 - Configurações do Arduino	53
Figura 32 - Obtendo o endereço IP do Arduino.....	54
Figura 33 - Smartphone com os aplicativos instalados	55
Figura 34 - Tela do menu principal do aplicativo no celular	56

LISTA DE TABELAS

Tabela 1 - Lista de Endereços de IP dos controladores	35
Tabela 2 - Descrição das entradas e saídas de cada controle	54
Tabela 3 – Resumo dos testes realizados de forma efetiva	56

LISTA DE ABREVIATURAS E SIGLAS

ADC	Conversor Analógico-Digital
APP	Aplicativo
AURESIDE	Associação Brasileira de Automação Residencial e Predial
DSSS	<i>Direct Sequence Spread spectrum</i>
FTP	<i>File Transfer Protocol</i>
GPIO	<i>General-purpose input/output</i> (Portas de Entrada e Saída)
HTTP	<i>Hypertext Transfer Protocol</i>
IDE	Ambiente de desenvolvimento integrado
IEEE	Instituto de Engenheiros Elétricos e Eletrônicos
IP	<i>Internet Protocol</i> (Protocolo de Internet)
Mbps	Megabit por segundo
MIT	Massachusetts Institute of Technology
OFDM	<i>Orthogonal Frequency Division Multiplexing</i>
PDA	<i>Personal Digital Assistant</i>
PWM	<i>Pulse Width Modulation</i>
QSPI	<i>Quad Serial Peripheral Interface</i>
SMTP	<i>Simple Mail Transfer Protocol</i>
SPI	Interface Serial Periférica
TCP/IP	<i>Transmission Control Protocol/Internet Protocol</i>
UART	Portas seriais de hardware
UART	Receptor/Transmissor Universal Assíncrono
UERGS	Universidade Estadual do Rio Grande do Sul
USB	<i>Universal Serial Bus</i>
WI-FI	<i>Wireless Fidelity</i>
WLAN	<i>Wireless Local Area Network</i>
WLAN	<i>Wireless local area network</i> (Rede local sem fio)
WPA2	<i>Wi-Fi Protected Access</i>
WPAN	<i>Wireless personal area network</i> (Rede pessoal sem fio)

SUMÁRIO

1	INTRODUÇÃO.....	13
1.1	PROBLEMÁTICA.....	14
1.2	HIPÓTESE.....	14
1.3	OBJETIVOS.....	14
1.3.1	Objetivo Geral.....	14
1.3.2	Objetivos Específicos.....	15
2	REFERENCIAL TEÓRICO.....	15
2.1	DOMÓTICA: Tecnologias e Conceitos.....	16
2.2	PROTOCOLOS E PADRÕES.....	18
2.2.1	Redes sem fio.....	18
2.2.2	WLAN (<i>Wireless Local Area Network</i>)	19
2.2.3	Arquitetura cliente-servidor.....	20
2.2.4	Modelo TCP/IP.....	20
2.2.5	Wi-Fi.....	21
2.3	ARDUÍNO.....	22
2.3.1	Módulos periféricos Arduino.....	23
2.3.2	Interface de programação do Arduino – IDE.....	24
2.3.3	Arduino Uno R3.....	25
2.4	MICROCONTROLADOR ESP8266.....	26
2.4.1	Módulos com o ESP8266.....	27
2.5	TELEFONES CELULARES – <i>SMARTPHONES</i>	28
2.6	APLICATIVOS PARA <i>SMARTPHONES</i>	28
2.6.1	Mercado de Aplicativos para <i>Smartphones</i>.....	29
2.6.2	App Inventor.....	29
2.7	PESQUISAS RELACIONADAS.....	30
2.7.1	Pesquisa 1 - Automação residencial utilizando a plataforma de hardware livre Arduino.....	30
2.7.2	Pesquisa 2 - Automação residencial controlada por dispositivos móveis por meio do comando de voz via <i>Bluetooth</i>.....	31
3	METODOLOGIA.....	32
3.1	ARQUITETURA DO SISTEMA.....	32

3.1.1	Arquitetura de hardware.....	33
3.1.2	Arquitetura do software.....	35
3.2	CIRCUITOS E SENSORES.....	37
3.2.1	Controlador.....	37
3.2.2	Sensores.....	38
3.2.2.1	<i>Sensor de Movimento (PIR).....</i>	38
3.2.2.2	<i>Sensor de Luminosidade (LDR).....</i>	38
3.2.2.3	<i>Sensor de Temperatura e Umidade (DHT22).....</i>	39
3.2.2.4	<i>Sensor de Gás e Chama.....</i>	40
3.2.2.5	<i>Sensores de Chuva e Umidade do solo.....</i>	41
3.3	APLICATIVO.....	41
3.3.1	Telas do Aplicativo.....	42
3.3.1.1	<i>Menu inicial.....</i>	43
3.3.1.2	<i>Tela para controle da iluminação.....</i>	44
3.3.1.3	<i>Tela para controle do sistema de temperatura.....</i>	45
3.3.1.4	<i>Tela para controle do sistema de irrigação do jardim.....</i>	46
3.3.1.5	<i>Tela do sistema de alarme de presença de gás ou fogo.....</i>	47
3.3.1.6	<i>Tela para controle de abertura da cortina.....</i>	47
3.3.1.7	<i>Tela para controle das tomadas.....</i>	48
3.3.2	Editor de blocos.....	49
3.3.3	Baixando o app para o celular.....	50
3.4	CONEXÃO DOS SISTEMAS.....	52
3.4.1	Configurando Entradas e Saídas.....	53
3.4.2	Endereços de IP dos dispositivos.....	54
4	RESULTADOS.....	55
5	CONCLUSÃO.....	58
	REFERÊNCIAS.....	60

1 INTRODUÇÃO

A domótica, ou automação residencial, é o uso da tecnologia a serviço do conforto dos moradores de uma casa. Apesar de ser um conceito básico, ela toma forma ao disponibilizar de maneira simples ao usuário doméstico tecnologias que foram criadas com o intuito de melhorar a qualidade de vida dos mesmos e que vem sendo aperfeiçoadas há pelo menos trinta anos (BIONDO, 2011).

É o conjunto de serviços proporcionados por uma coleção de equipamentos integrados com o objetivo de prover necessidades básicas de segurança, comunicação, gestão energética e conforto em uma habitação (MURATORI, 2014). É a atuação de dispositivos nas funções de elétrica, hidráulica e ar condicionado, permitindo o uso customizado de aparelhos elétricos e garantindo economia de energia elétrica e água (BOLZANI, 2004).

Atualmente, com a popularização dos telefones celulares dentre outros dispositivos móveis, é possível ter uma quantidade imensa de informações e acessá-las a qualquer hora e em qualquer lugar. Além disso, o crescente aumento da utilização de hardware livre, ou seja, a customização de hardware para os mais diversos segmentos, inclusive o de automação residencial, vem possibilitando o acesso e o desenvolvimento destes sistemas por um número maior de estudantes e profissionais a um custo acessível.

Diante deste contexto, surgiu a motivação para o desenvolvimento do presente trabalho, de maneira a explorar toda essa capacidade de controle da informação para criar um sistema de automação residencial. Para isso, foi utilizado o Arduíno, uma plataforma de *hardware* livre. Além disso, há o emprego da comunicação *Wi-Fi*, que é uma das tecnologias de comunicação sem fio mais difundidas atualmente.

Este trabalho busca desenvolver a integração entre microcontroladores Arduíno e *smartphones* com sistema operacional Android, de maneira que seja possível controlar dispositivos de uma residência. O usuário tem interação completa com os itens do imóvel conectados ao microcontrolador que, através de uma conexão *wireless*, podem ser monitorados e ativados através do aplicativo desenvolvido em plataforma aberta.

O objetivo deste trabalho é automatizar determinados processos de uma

residência através da utilização de ferramentas de desenvolvimento com protocolos abertos e aplicativo de celular Android, associados à implementação de circuitos com dispositivos de baixo custo para a obtenção de soluções de automação residencial, promovendo maior comodidade e segurança a um custo acessível, além da facilidade de instalação e customização.

1.1 PROBLEMÁTICA

Os principais problemas verificados na automatização de um protótipo residencial são: a segurança no acesso dos dispositivos pelo aplicativo via rede sem fio ou via cabo ethernet; eventuais faltas de energia que interrompem a comunicação entre os dispositivos; a dificuldade de instalação, uma vez que, conforme o projeto, o acesso a determinados locais possa ser mais desafiador. Além desses problemas citados, também é de se destacar a variação dos custos dos equipamentos no mercado, o que muitas vezes inviabiliza uma maior implementação do sistema.

1.2 HIPÓTESE

É possível o desenvolvimento de um sistema de automação residencial, utilizando microcontroladores, sensores e atuadores interligados por comunicação sem fio, em conjunto com um aplicativo desenvolvido para um dispositivo móvel e algoritmo para controle e configuração de forma a resolver parte dos problemas de sistemas de domótica, no que tange a instalação e custo reduzido, possibilitando também a customização do projeto para cada perfil de usuário de forma fácil e rápida.

1.3 OBJETIVOS

Abaixo estão elencados o objetivo geral e os objetivos específicos do presente trabalho.

1.3.1 Objetivo Geral

Este trabalho tem como objetivo geral o de compreender e investigar os conceitos envolvidos nos sistemas de automação residencial, bem como avaliar o emprego do Arduino em sistemas de automação. Para isso, é necessária a realização de um projeto prático, onde se possa desenvolver a interligação do hardware livre com a tecnologia de rede sem fio (também conhecida pelo nome em inglês *Wireless*).

Além disso, é fundamental a construção de um aplicativo para dispositivos móveis que possa se comunicar remotamente, por meio da rede *Wireless*, com os dispositivos de *hardware* Arduino.

1.3.2 Objetivos Específicos

O objetivo específico deste trabalho é desenvolver, através de uma plataforma de *software* e *hardware* abertos, a automação no âmbito de uma residência, possibilitando o controle da iluminação, controle de temperatura, o controle das tomadas elétricas, do sistema de irrigação, do sistema de segurança contra incêndio e vazamento de gás, bem como a abertura e fechamento das persianas da residência.

Seguem a seguir as etapas para o desenvolvimento do trabalho:

- a) primeira etapa: efetuar a montagem do protótipo da residência, no qual são aplicadas as ideias apresentadas neste trabalho, e a instalação dos equipamentos;
- b) segunda etapa: desenvolver o código para o Arduino e o aplicativo para o sistema Android em uma plataforma Web de código aberto;
- c) terceira etapa: efetuar a montagem final, integrando o *software* ao protótipo, e a realização dos testes de operação através da utilização de *smartphones*.

2 REFERENCIAL TEÓRICO

Este capítulo apresenta um breve conceito de Domótica, além de fornecer um embasamento histórico e de seu desenvolvimento. Haverá também, uma revisão de

literatura sobre os conceitos relevantes da rede *ethernet* e *wireless*, bem como o referencial tecnológico, abordando as tecnologias de *hardware* e *software* utilizados no trabalho.

2.1 DOMÓTICA: Tecnologias e Conceitos

A palavra domótica tem a sua raiz na palavra latina *domus* que significa casa. É a ciência moderna e multidisciplinar de engenharia que estuda a relação entre o homem e a residência (BOLZANI, 2010).

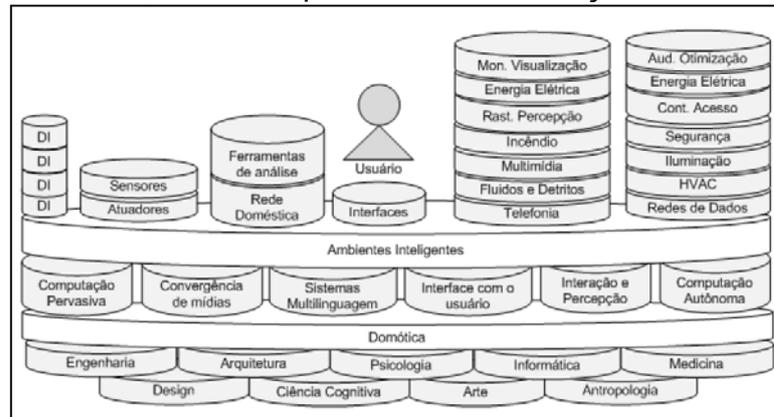
Segundo Muratori e Bó (2011), a domótica é automação e o controle aplicados a residência, que se realizam mediante o uso de equipamentos com capacidade de comunicarem-se entre si e de receberem comandos previamente estabelecidos pelos moradores, com possibilidade de otimização com base em seus interesses. Conseqüentemente, a domótica permite uma melhor qualidade de vida, reduz o trabalho doméstico, racionaliza o consumo de energia e água e, além disso, sua evolução permite oferecer continuamente novas aplicações.

Bolzani (2010, p. 17) ainda disserta sobre o conceito quando diz que:

A proposta de uma arquitetura de hardware e software tem como finalidade estabelecer critérios e métricas entre fabricantes e consumidores e fomentar o desenvolvimento das várias camadas de serviço que compõem uma residência inteligente: a integração de redes de controle; gerenciamento e análise de sensores e atuadores; interfaces; redes de comunicação; e todos os problemas que envolvem a imersão de pessoas em ambientes computacionalmente ativos. Este último deflagra uma série de discussões e questões em várias disciplinas quando o comportamento humano é avaliado. Os esforços da Engenharia, Medicina, Ciência da Computação, Inteligência Artificial e Psicologia se somam em uma nova ciência denominada Domótica.

A Fig. 1 ilustra o conceito de Bolzani, onde se pode notar a multiplicidade e complexidade das ciências que compõem a domótica.

Figura 1 - Domótica, seus componentes e sua relação com outras ciências

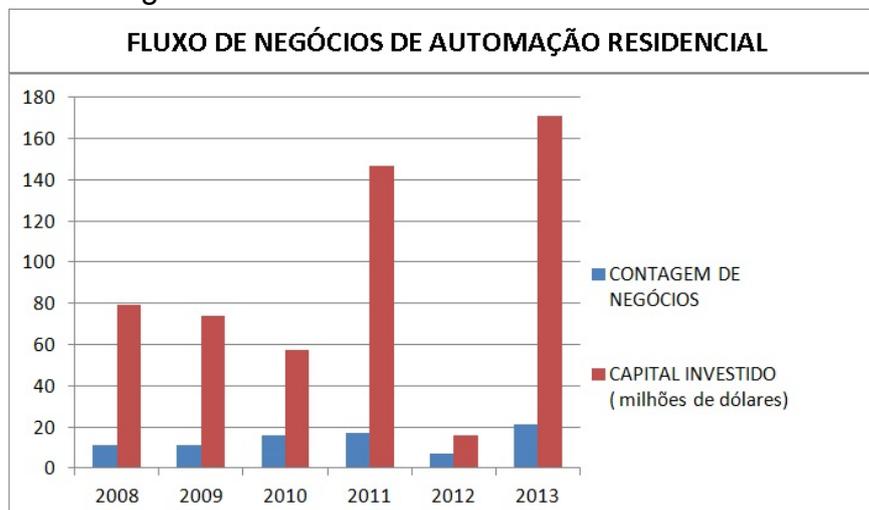


Fonte: Bolzani (2010)

Após a consolidação da automação industrial, a domótica se difundiu nos lares residenciais. Segundo dados divulgados pela Startupi (mais tradicional portal sobre o mercado de Startups, Inovação e Investimentos, Empreendedorismo e Tecnologia), no ano de 2013 foram investidos U\$130 milhões em domótica.

Segundo dados apresentados pela Associação Brasileira de Automação Residencial e Predial (AURESIDE), o mercado de automação residencial tem a projeção de crescer anualmente 11,35% entre 2014 e 2020, mundialmente. No Brasil, há cerca de 300 mil lares que já contam com essa tecnologia. Ou seja, o emprego da domótica estará cada vez mais consolidado no cotidiano das pessoas (AURESIDE, 2016). Na Fig. 2 pode ser visto o investimento realizado em domótica por ano.

Figura 2 - Investimento realizado em domótica



Fonte: Adaptado de Startupi

2.2 PROTOCOLOS E PADRÕES

Se um sistema eletrônico qualquer consiste de vários dispositivos dispostos em locais diferentes, e todos esses dispositivos necessitam trocar informações entre eles ou com uma central de controle, se faz necessário um meio de comunicação entre os mesmos. Par trançado de cabos, sinal de rádio frequência e fibra ótica são exemplos de meios de comunicação que podem ser utilizados para a troca de informações entre dispositivos eletrônicos. Para que haja a troca de informações, os dispositivos devem estar conectados entre si, ou seja, fazerem parte de uma rede de comunicação que os interliguem e lhes dêem a capacidade de se comunicarem.

Para a comunicação entre os dispositivos utilizados neste trabalho foi usada a comunicação sem fio, através da rede *Wi-Fi*.

2.2.1 Redes sem fio

Existem basicamente três tipos de redes sem fio:

- a) baseadas em infravermelho;
- b) baseadas em radiofrequência (*Wi-Fi* e *Bluetooth*);
- c) baseadas em laser.

Redes baseadas em infravermelho possuem como característica a não necessidade de licença para operação. Os produtos possuem cobertura mundial, portanto sem requerimentos específicos de cada país. Em geral são equipamentos de baixo custo e usam a mesma tecnologia que os sistemas de controle remoto, comumente utilizado em residências, com baixa taxa de erros. As primeiras redes sem fio criadas nos anos de 1990, antes mesmo da padronização, faziam uso dessa faixa de frequência.

Os sistemas baseados em laser utilizam a luz para a transmissão do sinal digital e não precisam de nenhum tipo de outorga ou autorização para o uso. Esses sistemas trabalham com alta largura de banda, chegando em alguns casos a até 2.5 gigabits por segundo e um alcance médio de dez quilômetros. Pelo alto custo e características, não são adequados para aplicação em domótica (MORAES, 2010).

As redes sem fio estão transformando-se em uma alternativa muito interessante para automação residencial. Os custos do sistema de cabeamento

estruturado são bastante elevados, fator este que contribuiu para o aumento de prédios não preparados para automação, aumentando-se ainda mais a aderência das redes sem fio, devido ao menor custo de implantação. A mobilidade dos pontos da *wireless Local Area Network* (WLAN), que permite movimentar-se em um escritório ou sala dentro de um prédio sem desconexão com a rede, é um dos pontos fortes para o uso de redes sem fio na automação predial. (ALVES, 2016).

2.2.2 WLAN (*Wireless Local Area Network*)

A WLAN permite a conexão entre dois ou mais computadores ou dispositivos digitais usando a tecnologia de modulação OFDM (Orthogonal Frequency Division Multiplexing) ou DSSS (Direct Sequence Spread spectrum) para permitir a comunicação entre esses dispositivos em uma área com a limitação de 10 a 100 metros. A rede WLAN possui uma taxa de transferência de dados nominal entre 11 e 54 Mbps (Megabits por segundo), que é mais lenta do que a rede local com fio que tem a taxa entre 100Mbps e 1000Mbps. Nesse caso, a WLAN é usada principalmente com dispositivos móveis, como laptop, celular e PDA (*Personal Digital Assistant*).

O Instituto de Engenheiros Elétricos e Eletrônicos (IEEE) é a autoridade na especificação e ratificação de padrões relacionados à tecnologia. Atualmente, os padrões WLAN são originários do IEEE, como por exemplo, o padrão IEEE 802.11a/b/g/n (interface que define características de um padrão de rede sem fio).

As redes de área local sem fio aumentaram durante os últimos 15 anos como aplicações críticas de negócios, particularmente desde 1997, quando o primeiro padrão IEEE 802.11 foi introduzido, seguido pela *Local Area Network* (LAN) de alto desempenho.

As principais vantagens em relação à implantação da LAN sem fio são a flexibilidade, a mobilidade e a economia. Empresas economizam milhões de dólares na implantação de redes sem fio em comparação com redes fixas permanentes. A WLAN recentemente testemunhou um crescimento explosivo, que tem grande relação com a ascensão de dispositivos de comunicação sem fio. As redes locais sem fio foram implantadas nos setores de educação, saúde, manufatura, varejo, hospitalidade, governo e transporte (MIND COMMERCE, 2012).

2.2.3 Arquitetura cliente-servidor

A arquitetura cliente-servidor é o modelo mais utilizado para aplicações distribuídas em redes de computadores. No modelo cliente/servidor, a comunicação costuma ocorrer através de uma mensagem do cliente enviada para o servidor, solicitando a execução de alguma tarefa. Em seguida, o servidor executa a tarefa e envia a resposta. Neste modelo cliente-servidor a comunicação sempre se inicia no cliente. O servidor é um *software* que mantém uma porta de comunicação aberta, a espera do cliente. O cliente também é um *software* e é normalmente acessado por um usuário através de uma interface, como por exemplo, um navegador web (OLIVEIRA, 2017).

2.2.4 Modelo TCP/IP

O protocolo TCT/IP foi desenvolvido originalmente pela Universidade da Califórnia para o Departamento de Defesa dos EUA. Atualmente o TCP/IP (*Transmission Control Protocol/Internet Protocol*) é o protocolo padrão para redes locais e remotas. Na realidade o TCP/IP é um conjunto de protocolos, dividido em quatro camadas: aplicação, transporte, rede e interface.

Cada camada é responsável por tarefas distintas de forma a garantir a integridade dos dados que trafegam pela rede. A camada de aplicação é utilizada pelos programas para enviar e receber informações de outros programas através da rede. Nela, encontram-se os protocolos como SMTP (*Simple Mail Transfer Protocol*) utilizado para email, o FTP (*File Transfer Protocol* - Protocolo de transferência de arquivos) e o HTTP (*Hypertext Transfer Protocol* - Protocolo de transferência de Hipertexto).

Uma vez que os dados tenham sido processados pela camada de aplicação, eles são enviados para a camada de transporte e rede, que é responsável por receber os dados enviados, verificar a integridade deles e dividi-los em pacotes. Na rede, os dados empacotados são recebidos e anexados ao endereço lógico (endereço IP) do computador remetente e do destinatário. Posteriormente os pacotes são enviados pela internet, para isso, são passados para a camada Interface. A tarefa da Interface é receber e enviar pacotes pela rede. Os protocolos

utilizados nessa camada dependem do tipo de rede que está sendo utilizado. Atualmente, o mais comum é o Ethernet, disponível em diferentes velocidades (PALMA; PRATES, 2000).

2.2.5 Wi-Fi

Wi-Fi é uma marca registrada da *Wi-Fi Alliance*. É utilizada por produtos certificados que pertencem a classe de dispositivos de rede local sem fios (WLAN) baseados no padrão IEEE802.11. Opera em faixas de frequência que não necessitam de licença para instalação/operação, o que a torna atrativa. No Brasil os equipamentos precisam ter o selo da Anatel, entretanto, não necessitam pagar nenhuma taxa para operar nas frequências (DAVIS, 2004).

Para se conectar a uma rede *Wi-Fi*, um computador deve ser equipado com uma interface de rede sem fio compatível. A combinação de um computador com uma interface controladora é chamada de "Estação". Todas as estações compartilham um único canal de comunicação de rádio frequência.

Transmissões neste canal são recebidas por todas as estações dentro do alcance. O hardware não informa ao usuário que a transmissão foi entregue e por isso é chamado de mecanismo de entrega de melhor esforço. A onda portadora é usada para transmitir os dados em pacotes, referidos como quadros ethernet. Cada estação está constantemente modificando o canal de comunicação de radiofrequência para pegar transmissões disponíveis (MONTICO, 2009).

Os preços dos circuitos *Wi-Fi* são relativamente baixos, transformando-os numa opção de rede econômica, cada vez com mais dispositivos incluídos.

Diferentes marcas concorrentes de pontos de acesso e interfaces de rede para clientes podem se comunicar operando em um nível básico de serviço. Os produtos designados como *Wi-Fi Certified* pela Aliança *Wi-Fi* são compatíveis. Ao contrário de telefones móveis, qualquer dispositivo *Wi-Fi* padrão irá funcionar em qualquer lugar do mundo (DAVIS, 2004).

2.3 ARDUÍNO

O Arduíno começou como um projeto do professor Massimo Banzi desenvolvido no Interaction Design Institute (Instituto de Design de Interação) em Ivrea, na Itália. No ano de 2005, em conjunto com colegas de trabalho, Banzi teve a ideia de desenvolver sua própria placa de prototipagem eletrônica como forma de prover a necessidade dos alunos por uma ferramenta acessível em termos de custo que substituísse a BASIC Stamp, utilizada por eles na época.

Eles criaram uma plataforma de *software* e hardware de fonte aberta, ou seja, que o projeto pode ser utilizado por qualquer pessoa para qualquer propósito, na qual seus estudantes pudessem facilmente criar projetos de eletrônica baseados em micro controladores baratos, o que rapidamente se difundiu pelo mundo (ARDUINO, 2010).

O Arduíno é, em termos práticos, um pequeno computador que você pode programar para processar entradas e saídas entre o dispositivo e os componentes externos conectados a ele, ou seja, um sistema que possibilita a interação com seu ambiente por meio de hardware e *software*.

Para programar o Arduíno utiliza-se uma interface de desenvolvimento denominada IDE. A IDE do Arduíno é um *software* livre no qual se escreve o código do programa, baseado na linguagem de programação chamada C++. O *hardware* é a placa do Arduíno em si e seus componentes são um microprocessador Atmel AVR, um cristal ou oscilador (relógio simples que envia pulsos de tempo em uma frequência especificada, para permitir sua operação na velocidade correta) e um regulador de tensão interno.

Há uma grande comunidade de pessoas utilizando plataformas Arduíno, compartilhando seus códigos e diagramas de circuito para que outros os utilizem e modifiquem. A maioria dessa comunidade também está muito disposta a auxiliar outros desenvolvedores (MCROBERTS, 2011). A Fig. 3 ilustra uma placa Arduíno Uno.

Figura 3 – Placa Arduino Uno



Fonte: Arduino (2018)

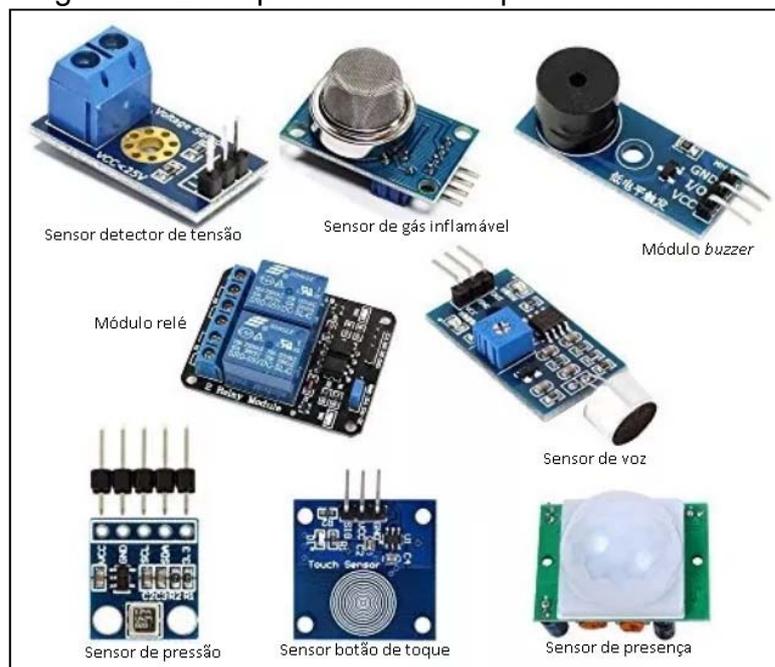
2.3.1 Módulos periféricos Arduino

A popularização do Arduino trouxe algumas vantagens importantes para o seu desenvolvimento. Acessíveis e muito úteis na integração de sistemas embarcados, conhecidos como *Shields*, diversos módulos começaram a ser produzidos e consumidos em larga escala.

Dentre os diversos módulos disponíveis, destacam-se os sensores, atuadores e módulos de comunicação. Entre os módulos sensores, é possível destacar o módulo de sonar, módulo detector de movimento, módulo sensor de umidade do solo, módulo sensor de temperatura e umidade do ar, módulo sensor de gases inflamáveis, módulo sensor de luminosidade, módulo sensor infravermelho, dentre dezenas de outros módulos de extrema utilidade em projetos de automação residencial.

Entre os módulos atuadores, os mais comuns são os módulos relé, módulos de controle de servomotores, módulo ponte H. Os módulos de comunicação incluem *Wi-Fi*, Ethernet e Bluetooth (OLIVEIRA, 2017). Na Fig. 4 são mostrados alguns exemplos de módulos periféricos Arduino.

Figura 4 - Exemplos de módulos periféricos Arduino



Fonte: Autor (2019)

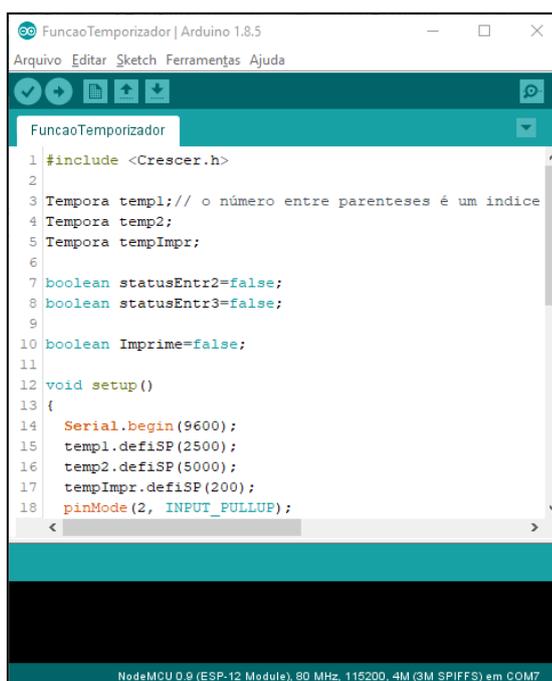
2.3.2 Interface de programação do Arduino - IDE

O ambiente de programação do microcontrolador foi desenvolvido pelos mesmos criadores da placa. Ele permite que o programador escreva um conjunto de instruções, passo a passo (que é seguida pelo dispositivo linha por linha), ou seja, um programa conhecido como *sketch*. Através do comando do usuário é realizada a compilação e o *upload* para o microprocessador da placa Arduino. Ele poderá então, por meio de tal conjunto de instruções, executar a ação, interagindo com os dispositivos ligados as entradas e saídas do Arduino (MCROBERTS, 2011).

Todos os programas para Arduino devem conter as duas funções principais: `setup()` e `loop()`. A primeira é executada assim que a placa é ligada, antes de qualquer outra função. Por isso geralmente é usado para configuração dos pinos de entrada/saída da placa e declaração das variáveis. Já a segunda, executada exatamente após o `setup()`, é um laço infinito, isto é, o bloco de comandos recomeça sempre que ele chega ao fim. É possível chamar uma dessas funções no escopo da outra. A vantagem de se utilizar o Ambiente de Desenvolvimento Integrado (IDE) é a possibilidade de se programar qualquer placa Arduino, bem como outras diversas

placas (NodeMCU, Sparkfun, entre outras), no mesmo *software*. Isso ocorre graças à compatibilidade que o programa tem com as principais placas do mercado (ARDUINO, 2019). Na Fig. 5 pode ser vista a tela de programação da IDE Arduíno.

Figura 5 - Tela do software de programação do IDE do Arduíno



```
FuncaoTemporizador | Arduino 1.8.5
Arquivo  Editar  Sketch  Ferramentas  Ajuda

FuncaoTemporizador
1 #include <Crescer.h>
2
3 Tempora temp1; // o número entre parenteses é um indice
4 Tempora temp2;
5 Tempora tempImpr;
6
7 boolean statusEntr2=false;
8 boolean statusEntr3=false;
9
10 boolean Imprime=false;
11
12 void setup()
13 {
14   Serial.begin(9600);
15   temp1.defiSP(2500);
16   temp2.defiSP(5000);
17   tempImpr.defiSP(200);
18   pinMode(2, INPUT_PULLUP);

```

NodeMCU 0.9 (ESP-12 Module), 80 MHz, 115200, 4M (3M SPIFFS) em COM7

Fonte: Autor (2019)

2.3.3 Arduíno Uno R3

Arduíno / Genuíno Uno é uma placa microcontroladora baseada no microcontrolador ATmega328P. Possui 14 pinos que podem ser configurados como entrada ou saída digital. Destes 14 pinos, 6 podem ser usados como saída PWM. Além disso, possui 6 entradas analógicas, um cristal de quartzo de 16 MHz, uma conexão USB, um conector de energia, um botão de reset e um conector ICSP (In-Circuit Serial Programming) que permite atualização do firmware diretamente no circuito integrado através de um dispositivo de programação da fabricante Atmel.

A placa Uno é compatível com a maioria dos módulos periféricos projetados para a placa Arduíno Mega e para as antigas placas Duemilanove ou Diecimila. Esta placa foi escolhida para este trabalho pela quantidade de entradas, sua robustez e quantidade de memória acima das demais placas Arduíno (ARDUINO, 2019).

Neste trabalho serão utilizadas shields para comunicação via ethernet e para

acionamento do motor de corrente contínua, chamada Shield Ponte H (Circuito usado no acionamento e para inversão do sentido de rotação de motores de corrente contínua), integrada ao Arduíno Uno.

2.4 MICROCONTROLADOR ESP8266

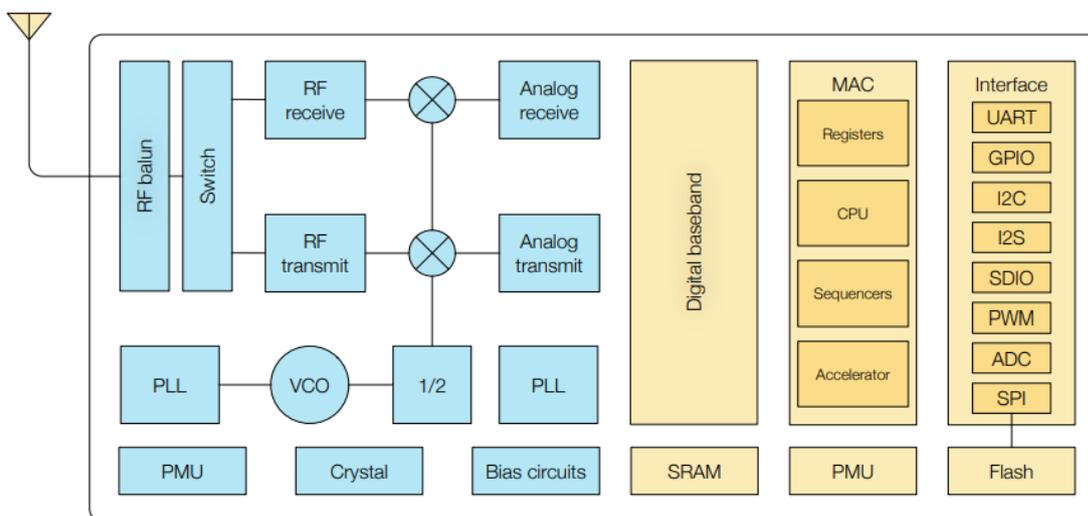
De forma simples, Internet das Coisas é o modo como os objetos físicos estão conectados e se comunicando entre si e com o usuário, através de sensores inteligentes e softwares que transmitem dados para uma rede (OLIVEIRA, 2017). Com a entrada da Internet das Coisas em nossas casas e em nossas vidas, aliada a necessidade de consumir e controlar uma enorme quantidade de dados todos os dias, torna-se possível o controle de nossas casas através de um *smartphone* (BATRINU, 2018).

Para tornar a Internet das Coisas uma realidade, foi preciso que o custo das soluções utilizadas para realizar as conexões com os objetos fosse compatível com os mesmos, sendo o microcontrolador ESP8266 considerado uma solução a altura para tal tarefa (OLIVEIRA, 2017).

O ESP8266 é um microcontrolador RISC de 32 bits e baixo custo com recursos de TCP/IP e *Wi-Fi*, desenvolvido pela Espressif Systems, uma empresa localizada em Xangai. É capaz de funcionar a 80MHz ou 160MHz. Possui 64KB de RAM para instruções e 96KB de RAM para dados. Para o *firmware* e outros armazenamentos de dados utiliza uma memória flash QSPI (*Quad Serial Peripheral Interface*) externa e o tamanho pode variar de 512KB a 4MB.

O próprio chip tem 16 pinos GPIO (Pinos de entrada ou saída de uso geral), porém, alguns deles são usados para conexão com a memória flash QSPI. Os demais pinos podem funcionar como SPI (*Serial Peripheral Interface*), I²C, I²S, Receptor/Transmissor Universal Assíncrono (UART) e um conversor Analógico-Digital (ADC) de 10 bits. A conectividade *Wi-Fi*, compatível com o padrão IEEE 802.11b/g/n, oferece autenticação WPA e WEP, mas também pode se conectar a redes abertas. A Fig. 6 ilustra o diagrama funcional do microcontrolador ESP8266, indicando o conjunto de componentes para processamento (CPU + memória), interfaces de entrada e saída para diversos tipos de protocolos e aplicações, e comunicação (OLIVEIRA, 2017).

Figura 6 - Arquitetura do microcontrolador ESP8266



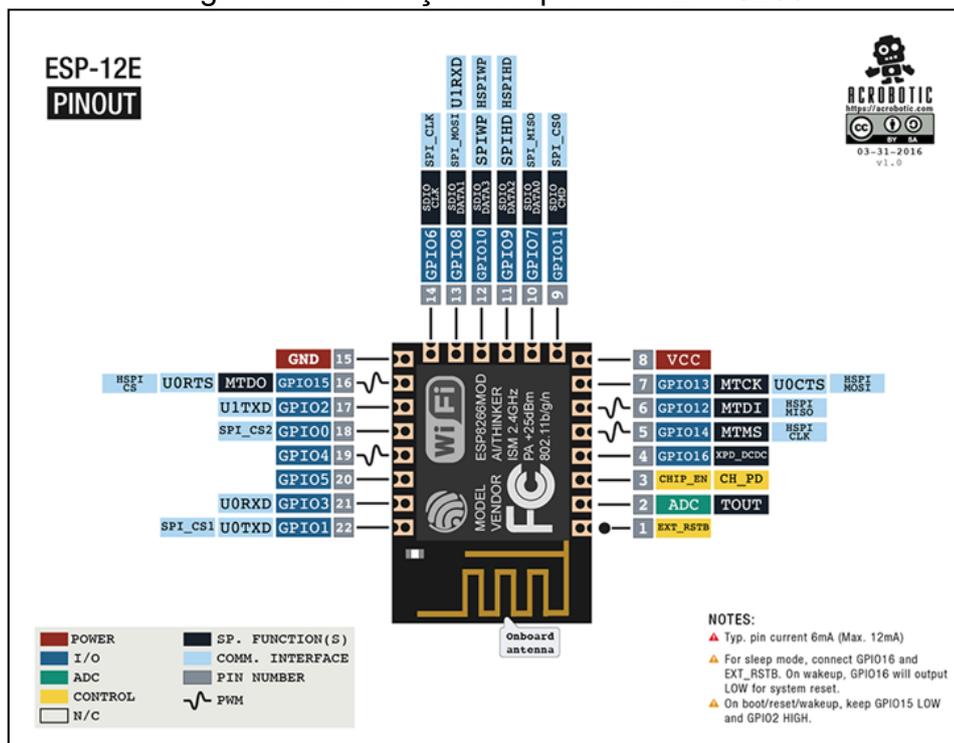
Fonte: Folha de dados do ESP8266 (2019)

2.4.1 Módulos com o ESP8266

Existem diversos módulos disponíveis no mercado que usam o microcontrolador ESP8266. Além do tamanho reduzido, quando comparado com os módulos Arduino, o baixo custo destes dispositivos os torna uma opção interessante para utilização em projetos de automação residencial. O módulo ESP-01 é o mais simples disponível, possui apenas 8 pinos. Para ser gravado é necessário o uso de um adaptador USB (*Universal Serial Bus*). O módulo ESP-12 foi construído para ser integrado em uma placa de circuito impresso, possibilitando agregar outros componentes. É utilizado em projetos de média e larga escala.

O módulo mais popular é o chamado NodeMCU, que agrega um conversor USB-serial e um regulador de tensão de 5 volts para 3,3 volts. O NodeMCU é um módulo bastante indicado para projetos de prototipagem devido a sua pinagem e facilidade de montagem. O consumo de energia deste módulo varia entre 10 micro amperes até 170 mili amperes, dependendo do modo de processamento (OLIVEIRA, 2017). Na Fig. 7 é apresentado o esquema de pinagem do módulo ESP8266.

Figura 7 – Descrição dos pinos do ESP8266



Fonte: Acrobotics Industries (2019)

2.5 TELEFONES CELULARES – SMARTPHONES

São dispositivos de altíssima tecnologia que oferecem recursos avançados comparados ao de um computador pessoal e que possuem um sistema operacional de um terceiro. Capazes de rodar *softwares* de terceiros, conhecidos como Apps, ou seja, suportam uma gama de aplicativos desenvolvidos por outras empresas ou mesmo pessoas físicas. Estes aplicativos são disponibilizados em lojas virtuais que funcionam tendo como base um sistema operacional, como por exemplo, o Android da Google e o IOS da Apple (COUTINHO, 2014).

2.6 APLICATIVOS PARA SMARTPHONES

Atualmente qualquer *smartphone* dispõe de diversos aplicativos instalados no seu sistema, bem como permite a posterior instalação de outros programas. E este mercado de aplicativos tem movimentado muito dinheiro, uma vez que as pessoas estão cada vez mais conectadas e utilizando os seus telefones celulares. Além disso, os aplicativos estão sendo desenvolvidos para diversas utilidades e

aplicações, o que atrai a procura dos usuários.

Se por um lado a demanda tem crescido, por outro, nunca foi tão fácil desenvolver um aplicativo. Existem diversas empresas que já oferecem plataformas gratuitas para a criação do seu próprio aplicativo e suporte para os usuários (COUTINHO, 2014).

2.6.1 Mercado de Aplicativos para *Smartphones*

Com a popularização dos *smartphones* e o avanço tecnológico dos mesmos, as pessoas estão fazendo mais downloads de aplicativos e gastando neste importante mercado. Segundo dados recentes apresentados pela App Annie, empresa norte-americana de dados de mercado de aplicativos, o número de aplicativos para celulares baixados em 2017 alcançou o valor de 175 bilhões de programas no mundo, com destaque para os países emergentes Brasil e Índia.

Além disso, o Brasil figura o quarto lugar na lista dos países que mais consomem aplicativos. Para a Apple Store (loja virtual da Apple – Sistema IOS), o país é o nono maior mercado, enquanto que para a Google Play (loja virtual da Google – Sistema Android) o Brasil é o segundo principal mercado, atrás somente da Índia.

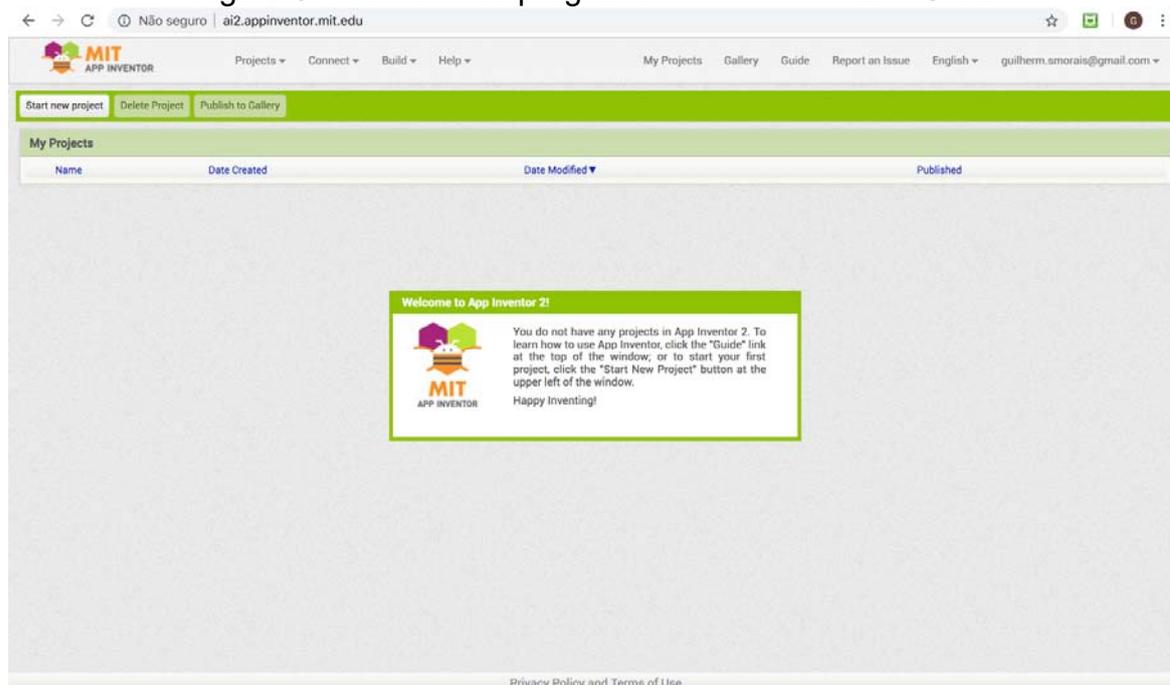
Ainda segundo o estudo, o brasileiro tem em média cerca de 80 aplicativos instalados em seu celular e gasta aproximadamente 200 minutos por dia com aplicativos conectados à internet. Em dois anos, houve um crescimento de 20% no número de downloads e as projeções mostram que este mercado tende a se manter aquecido (ESTADÃO, 2018)

2.6.2 App Inventor

O APP INVENTOR é uma aplicação de código aberto com o intuito de permitir que novos usuários pudessem criar aplicativos para o sistema Android. Para isso, a plataforma utiliza uma interface gráfica bastante intuitiva, onde é possível criar e deslocar objetos na tela, que posteriormente, farão parte do funcionamento do aplicativo. De acordo com a empresa, até mesmo uma criança tem condições de interagir com a plataforma e criar o seu próprio aplicativo.

Criado em 2010, o projeto passou a ser mantido pela MIT (*Massachusetts Institute of Technology*), com apoio financeiro da Google. Recentemente, o aplicativo foi atualizado e lançada a versão MIT APP INVENTOR 2 (APPINVENTOR, 2019). Na Fig. 8 pode ser vista a interface do programa MIT APP INVENTOR 2.

Figura 8 - Interface do programa MIT APP INVENTOR 2



Fonte: Autor (2018)

2.7 PESQUISAS RELACIONADAS

No âmbito de automação residencial, foram encontrados os seguintes trabalhos mais próximos a este: *Automação residencial utilizando a plataforma de hardware livre Arduino*, de Wglierber Sousa Freitas, e *Automação residencial controlada por dispositivos móveis por meio do comando de voz via bluetooth*, de Rafael Garcia Pinheiro e Marcelo Tadeu Boer. Ambos os trabalhos utilizam Controladores Arduino, entretanto, a comunicação não é refeita via *Wi-Fi*.

2.7.1 Pesquisa 1 - Automação residencial utilizando a plataforma de *hardware* livre Arduino

No trabalho desenvolvido por Wglierber Sousa Freitas, é criado um sistema de automação residencial fazendo uso de *hardware* livre. Com o objetivo de demonstrar que é possível utilizar tecnologias abertas e de baixo custo, foi construído um sistema que é capaz de controlar remotamente a iluminação e o portão de uma casa feita em pequena escala por meio do uso de uma placa Arduíno, da tecnologia Bluetooth e de um aplicativo para dispositivo móvel.

Segundo o autor, o resultado do trabalho mostrou que, graças às inúmeras possibilidades oferecidas pelo *hardware* livre, é possível a realização de projetos de formas mais simples e acessíveis.

2.7.2 Pesquisa 2 - Automação residencial controlada por dispositivos móveis por meio do comando de voz via bluetooth

Neste trabalho, os autores desenvolveram um aplicativo para o celular, através da ferramenta MIT App Inventor, e a programação do controle dos dispositivos foi feita com a tecnologia Arduíno. Segundo os mesmos, o MIT App Inventor apresentou características que supriram as necessidades na criação de aplicativos para *smartphones*, para o perfeito funcionamento da conexão do Bluetooth do celular com o módulo Bluetooth do Arduíno e no controle de ativação e desativação dos dispositivos eletrônicos por meio do comando de voz.

3 METODOLOGIA

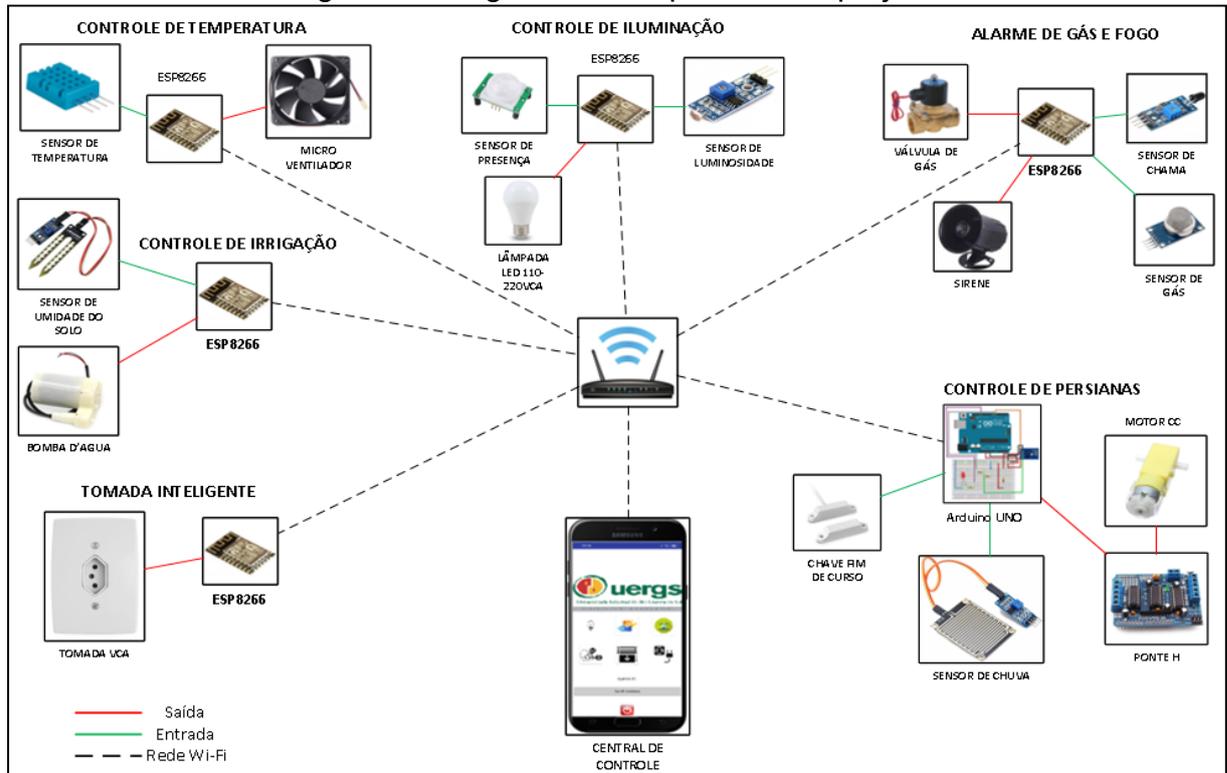
O sistema de automação residencial implementado neste trabalho aborda algumas das possibilidades existentes para controle de dispositivos por meio de uma rede sem fio, com o propósito de customizar e facilitar instalação. Foram utilizados dispositivos Arduíno e microcontroladores de baixo custo, além da descentralização dos controles, o que só foi possível com a utilização da rede *Wi-Fi* e de um circuito de acionamento com fonte estabilizada e relés de acionamento em cada ponto. No decorrer deste capítulo serão mostrados e explicados todos os componentes utilizados e como a interação entre eles ocorre.

O projeto foi desenvolvido em quatro partes, definidas da seguinte maneira: a primeira parte é constituída pelo *hardware*, onde serão explicados os circuitos. A segunda é constituída pelo aplicativo criado para a plataforma Android, a terceira pelo código criado para os microcontroladores e por fim a conexão entre o aplicativo e os dispositivos via rede *Wi-Fi*.

3.1 ARQUITETURA DO SISTEMA

A arquitetura foi escolhida em função do foco do trabalho em distribuir os dispositivos de forma a facilitar a instalação. Para compreender o seu funcionamento mostrado na Fig. 9 diagrama com os principais componentes que fazem parte do sistema. O usuário acessa as funcionalidades do sistema através de um telefone celular, utilizada como central. Ao selecionar uma ação, um comando específico é enviado por *Wi-Fi* para o módulo correspondente.

Figura 9 - Diagrama da arquitetura do projeto



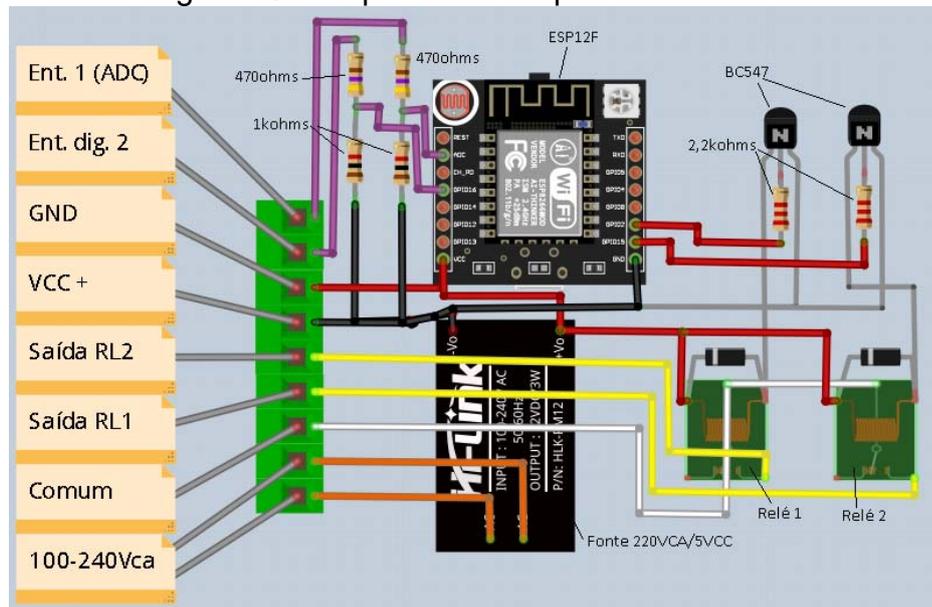
Fonte: Autor (2019)

3.1.1 Arquitetura de *hardware*

Antes de trabalhar cada circuito de forma independente, será mostrado como ocorre a conexão física do circuito composto por um microcontrolador ESP8266, fonte chaveada com entrada de tensão alternada que pode variar entre 100 e 240 volts, relé de 3 ou 5VCC, bem como circuito para acionamento do relé.

Esta montagem é base para a maioria dos circuitos de controle, pois foi projetada para prover alimentação de contínua de 5 volts para os sensores Arduino, além de possuir duas entradas e duas saídas a relé. Um dos objetivos do projeto foi o de obter uma montagem com tamanho reduzido, com o intuito de facilitar a montagem e disposição em caixas de passagem elétricas. Na Fig. 10 é mostrado o esquemático da placa de controle.

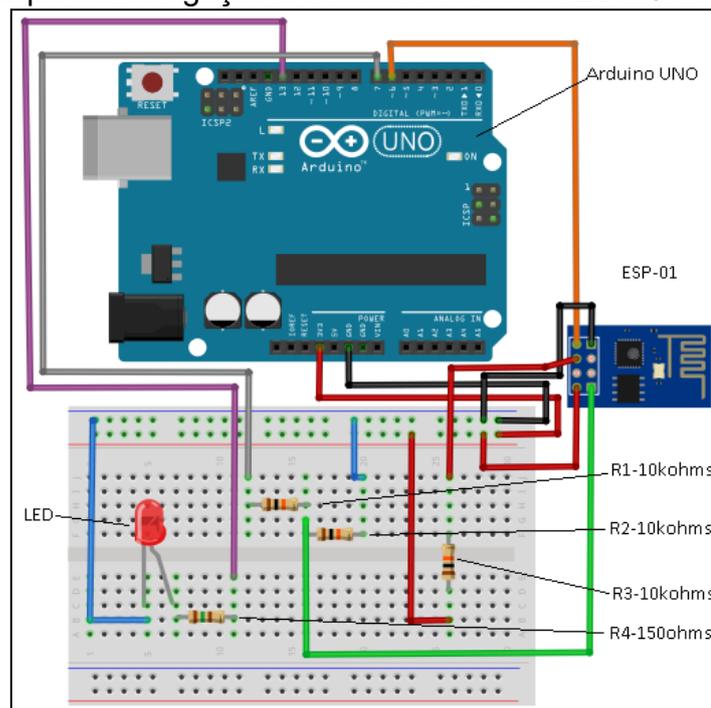
Figura 10 - Esquemático da placa de controle



Fonte: Autor (2019)

Para acionamento e controle da abertura da cortina foi utilizado com módulo Arduino Uno ligado a uma *shield* de ponte H. Como o módulo Uno não possui comunicação *Wi-Fi*, foi necessário adequar o circuito com a utilização de um módulo ESP-01 com *Wi-Fi* conectado a porta serial, conforme ilustrado na Fig. 11.

Figura 11 - Esquema de ligação do microcontrolador ESP-01 ao Arduino Uno



Fonte: Autor (2019)

3.1.2 Arquitetura do *software*

A implementação das rotinas de programação é composta basicamente por dois módulos. O primeiro encontra-se na interface do usuário no dispositivo com Android e foi programado através do App Inventor. O segundo módulo de *software* fica no microcontrolador. Cada circuito de controle possui um microcontrolador e conseqüentemente um *software* específico, dependendo da aplicação.

No módulo do Android existe uma tela inicial com opções de acesso a outras seis telas, cada uma delas específica para um determinado controle, conforme listado abaixo:

- a) Controle de Iluminação;
- b) Controle de temperatura;
- c) Controle de irrigação;
- d) Alarme de gás e fogo;
- e) Controle da cortina;
- f) Tomadas inteligentes.

Quando o usuário acessar determinada tela o programa do App Inventor enviará um comando ao um determinado dispositivo com endereço de IP específico. A tabela 1 mostra uma lista com os IP's dos controladores criada para facilitar a programação dos mesmos.

Tabela 1 - Lista de Endereços de IP dos controladores

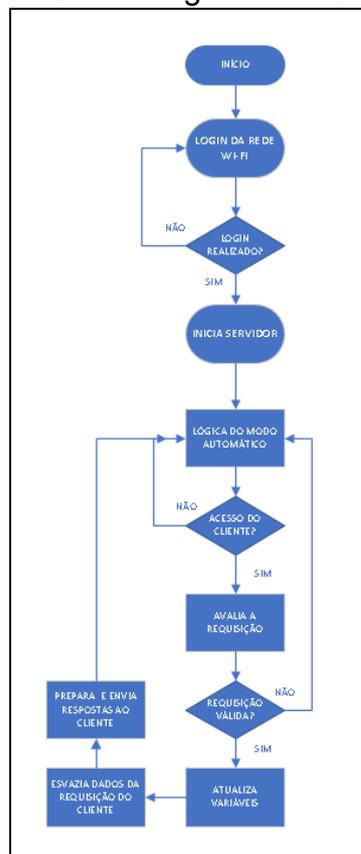
IP'S APP INVENTOR		
TELA	DESCRIÇÃO	IP DO MICROCONTROLADOR
1	ILUMINAÇÃO SALA	192.168.1.25
1	ILUMINAÇÃO QUARTO	192.168.1.26
1	ILUMINAÇÃO COZINHA	192.168.1.27
1	ILUMINAÇÃO BANHEIRO	192.168.1.28
1	ILUMINAÇÃO EXTERNA	192.168.1.29
2	CONTROLE DE TEMPERATURA	192.168.1.30
3	CONTROLE DA IRRIGAÇÃO	192.168.1.31
4	ALARME DE GÁS E FOGO	192.168.1.32
5	CONTROLE PERSIANA	192.168.1.33
6	CONTROLE TOMADA	192.168.1.34

Fonte: Autor (2019)

As rotinas de *software* desenvolvidas para os dispositivos de controle compartilham a mesma estrutura básica de inicialização e comunicação, diferindo no que diz respeito à lógica de sensores e endereçamento de rede. Os microcontroladores foram programados como servidor e o aplicativo programado para atuar como cliente, enviando uma requisição (*frame*) com o comando a ser executado. Após a requisição do cliente ser recebida e tratada o servidor envia uma resposta ao cliente. Essa resposta carrega informações do *status* dos sensores e saídas, os quais são tratados e mostrados na tela do aplicativo. O fluxograma mostrado na Fig. 12 ilustra o funcionamento básico da rotinas do programa desenvolvido para o servidor (microcontrolador).

Inicialmente o microcontrolador efetua o *login* na rede do usuário. Caso o *login* na rede seja efetivado o programa executa a inicialização do servidor. A partir desse ponto, as rotinas lógicas de leitura das entradas e ativação das saídas são executadas, levando-se em conta os dados enviados pelo cliente através do tratamento das requisições enviadas pelo mesmo.

Figura 12 – Fluxograma do *software*



Fonte: Autor (2019)

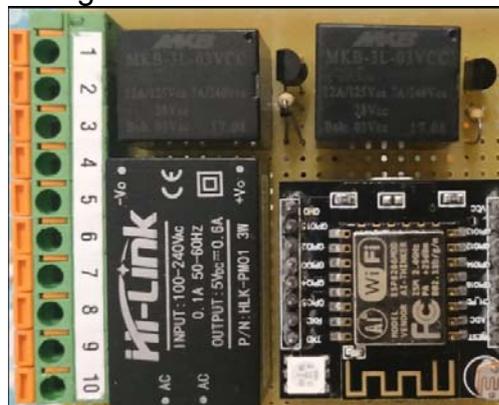
3.2 CIRCUITOS E SENSORES

Para este trabalho foram criados circuitos para leitura de diferentes sensores e acionamentos de saídas distintas através de relés e ponte H. Para cada circuito, com exceção do controle da cortina, utilizou-se um módulo ESP12-E com microcontrolador ESP8266. Este módulo foi utilizado devido a possuir um circuito regulador de tensão de 5 Volts para 3,3 Volts. O ESP8266, bem como suas entradas e saídas, opera com tensão máxima de 3,3 Volts, entretanto, a maioria dos sensores Arduino operam com tensões de 5 Volts, sendo necessário utilizar um divisor de tensão nas entradas.

3.2.1 Controlador

Os módulos controladores utilizam o modulo ESP12 com *Wi-Fi* nativo que possui um microcontrolador ESP8266 produzido pela empresa Espressif Systems, uma fabricante de semicondutores Chinesa. Na montagem foi utilizada um fonte chaveada com saída estabilizada de 5 Volts, para alimentação do módulo ESP8266 e para alimentação dos sensores. Também utilizou-se um circuito para acionamento dos relés e um divisor de tensão para ajustar a tensão ao valor máximo admissível pelas entradas do ESP8266, que é de 3,6 Volts. A Fig. 13 mostra a montagem da placa finalizada.

Figura 13 - Placa de controle



Fonte: Autor (2019)

3.2.2 Sensores

A ligação dos sensores de cada aplicação a placa de controle ocorreu de forma padrão, ou seja, os sensores que possuem saídas analógicas foram ligados na entrada analógica (A0) e os sensores com saída digital foram ligados a entrada digital 16.

3.2.2.1 Sensor de Movimento (PIR)

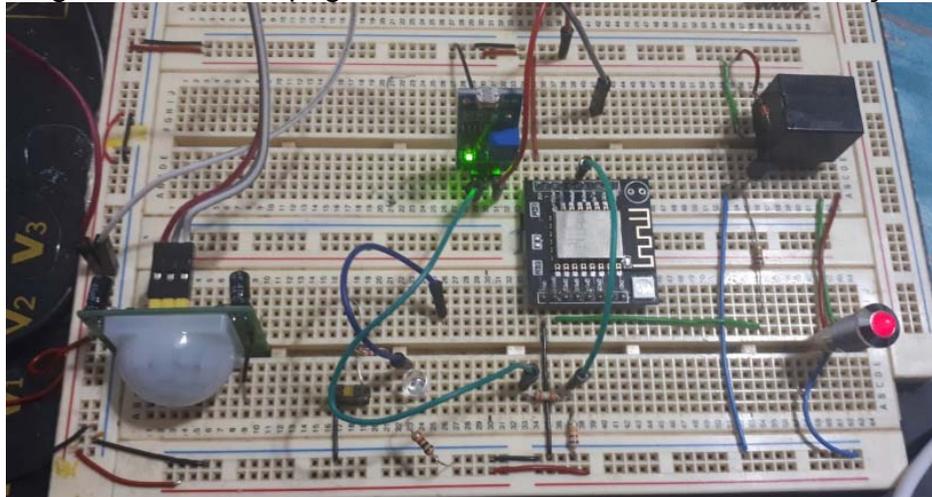
Foi utilizado um sensor que detecta variações de radiação infravermelha conhecido como PIR (*Passive Infrared Sensor*), modelo HC-SR501. Essas variações acontecem quando uma pessoa acessa o perímetro coberto pelo sensor, que identifica a ocorrência de movimento. São de baixo custo e consomem pouca energia, ideais para aplicação neste trabalho.

A tensão de alimentação deste sensor varia entre 5 e 12 VDC e tem consumo menor que 1 mA. A distância de detecção pode ser ajustada entre 3 e 7 metros, com um ângulo de 110 °C. A saída digital é de 3,3 VDC, que pode ter um atraso no sinal de saída ajustado entre 3 segundos a 5 minutos.

3.2.2.2 Sensor de Luminosidade (LDR)

Para medição de luminosidade foi utilizado um módulo que utiliza um fotoresistor. O circuito possui saída digital e ajuste de sensibilidade, o que possibilitou a utilização da entrada digital para seu monitoramento. Os circuitos de controle de cortinas e de iluminação também tem esse sensor em seu circuito. Como a saída deste sensor é de 5 Volts foi necessário utilizar um divisor de tensão na entrada do módulo ESP8266. Na Fig. 14 é mostrada uma das montagens realizadas em matriz de contatos.

Figura 14 – Prototipagem do circuito de controle da iluminação



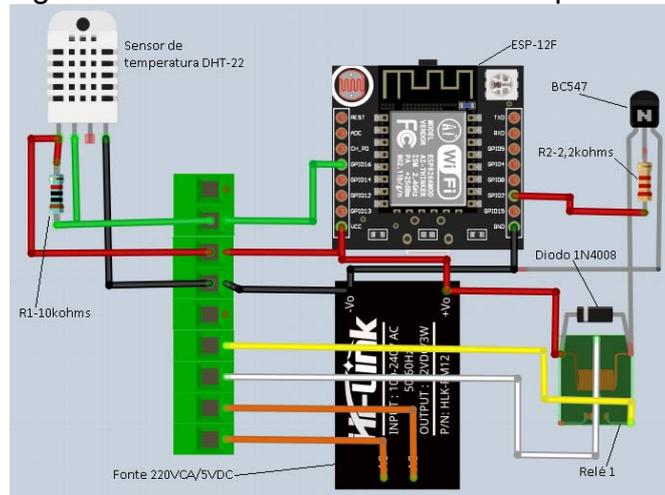
Fonte: Autor (2019)

3.2.2.3 Sensor de Temperatura e Umidade (DHT22)

O sensor de temperatura utilizado foi o modelo DHT22 que permite medir temperaturas entre -40 a +80 graus Celsius e umidade entre 0 a 100%. Também chamado de AM2302, este módulo integra um sensor de umidade e um sensor de temperatura no mesmo invólucro. O sensor DHT22 fornece temperatura e umidade do ar, de uma forma muito rápida, através da saída dados. Utiliza um sensor capacitivo de umidade e um termistor para medir o ar circundante, ambos conectados a um controlador de 8 bits que produz um sinal digital serial no pino de dados (DATA).

Para utilização deste sensor é necessária a instalação de uma biblioteca específica na IDE, além de um resistor de *pull-up* conectado entre os pinos DATA e a alimentação de tensão. A ligação deste sensor ao microcontrolador ESP8266 é mostrada na Fig. 15.

Figura 15 - Circuito de controle de temperatura



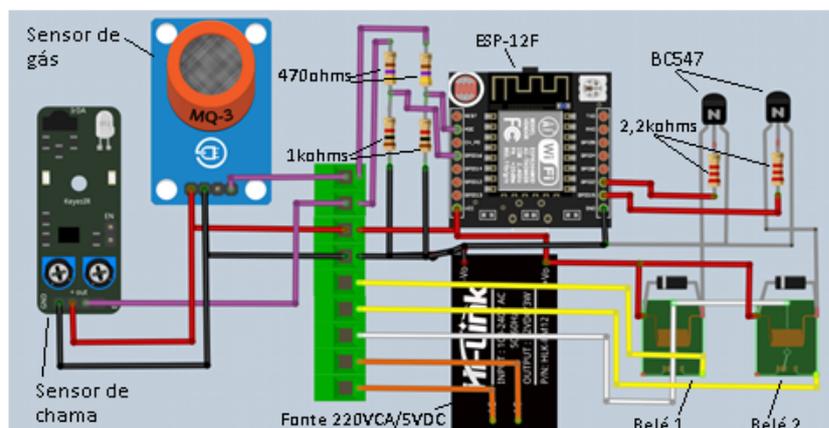
Fonte: Autor (2019)

3.2.2.4 Sensor de Gás e Chama

O sensor de gás utilizado no trabalho foi o MQ-2. Capaz de detectar a presença de vários gases, entre eles o butano, conhecido popularmente como gás de cozinha. Para a detecção de chama utilizou-se o sensor infravermelho, capaz de detectar chama ou outras fontes de calor que possuam tamanho de onda entre 760 a 1100nm.

Quando há ocorrência de chama a saída do sensor fica em estado baixo (0) e quando não há detecção, fica em estado alto (1). Estas informações também valem para o sensor de gás, que, para este trabalho, teve sua saída analógica utilizada. O desenho do circuito é mostrado na Fig. 16.

Figura 16 - Esquemático do circuito de detecção de gás e chama



Fonte: Autor (2019)

3.2.2.5 Sensores de Chuva e Umidade do solo

Por se tratarem de dois sensores com características muito parecidas, os sensores de chuva e umidade do solo (Modelo HL-69) serão abordados neste mesmo tópico.

Estes sensores possuem uma característica comum que é de fazer a leitura da resistência entre dois eletrodos e posteriormente transformar em sinal digital através de um circuito intermediário que utiliza um comparador LM393. Ambos possuem uma placa de controle com *leds* que indicam alimentação de tensão e saída digital, além de um potenciômetro para ajuste da sensibilidade e saída analógica.

Sem a presença de umidade ou de chuva, o sinal de saída fica em estado alto (1). O sensor de presença de chuva e o sensor de umidade do solo foram utilizados nos controles de abertura da cortina e no controle de irrigação do solo respectivamente.

3.3 APLICATIVO

A elaboração do aplicativo para o *smartphone* foi implementada com base em um diagrama de ligação e comunicação dos dispositivos utilizados, ilustrado na Fig. 9. Assim, a partir do site onde os aplicativos são criados (<http://ai2.appinventor.mit.edu>), as telas de interação com o usuário foram desenvolvidas, de maneira que fosse possível realizar os comandos e leituras de estados dos dispositivos nas mesmas, configuração do IP de cada dispositivo, bem como o acionamento das saídas e seleção de cada controle em modo manual e em automático.

Paralelamente a criação das telas, iniciou-se a programação dos microcontroladores, uma vez que era necessário realizar testes de comunicação entre o app e os equipamentos. Tal medida foi tomada a fim de garantir que os mesmos estivessem conectados à mesma rede *Wi-Fi*.

Além disso, foram definidas as entradas e saídas dos microcontroladores, bem como as funções de controle e leitura, com o intuito de assegurar a troca de informações entre o aplicativo e os controladores. Portanto, sempre que ocorresse

uma solicitação na tela do *smartphone*, a mesma deveria ser enviada e realizada no respectivo equipamento.

Por fim, com o aplicativo e os *hardwares* devidamente programados, os mesmos foram testados e os seus resultados foram analisados e avaliados.

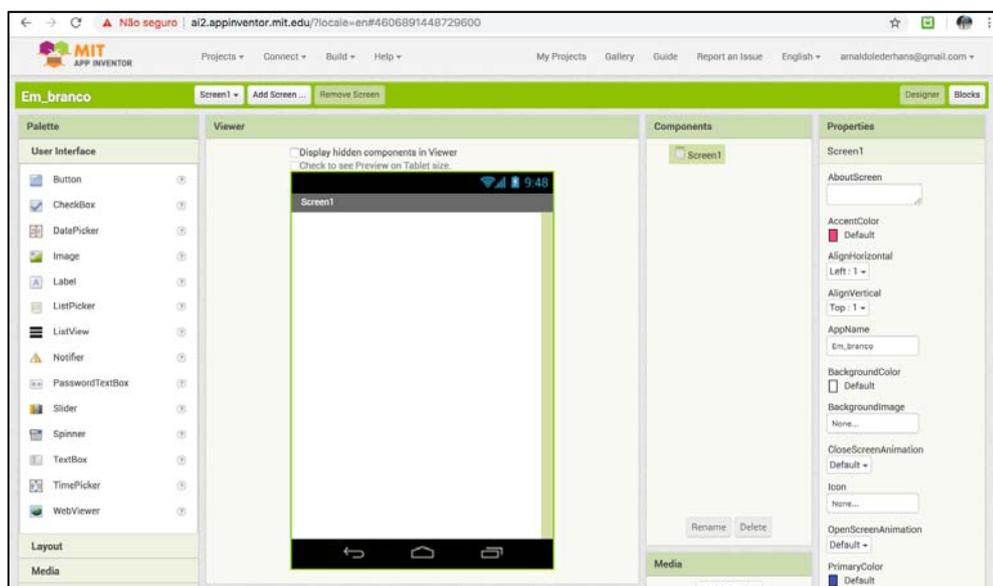
3.3.1 Telas do aplicativo

Para o desenvolvimento das telas, o site fornece um visualizador no centro da página, mostrado na Fig. 17, onde é possível simular a tela de um *smartphone* em tempo real. Neste campo, inicialmente em branco, são inseridos objetos, strings, botões, imagens, entre outros, que dão origem as telas do aplicativo posteriormente.

Como o site de criação do aplicativo só permite a visualização de uma tela por vez, para visualização das demais telas existe um botão com caixa de seleção, onde é possível escolher a tela desejada. Além disso, no lado direito da página, são definidas as principais características da tela no campo *Properties*, como nome, cor de fundo, *layout*, organização, entre outras definições.

Ainda no lado direito, na área *Components*, é possível acompanhar todos os componentes selecionados e que estão sendo utilizados na tela. Isso permite visualizar de forma geral a organização dos objetos e de que forma eles estão conectados.

Figura 17 – Local onde são desenvolvidas as telas

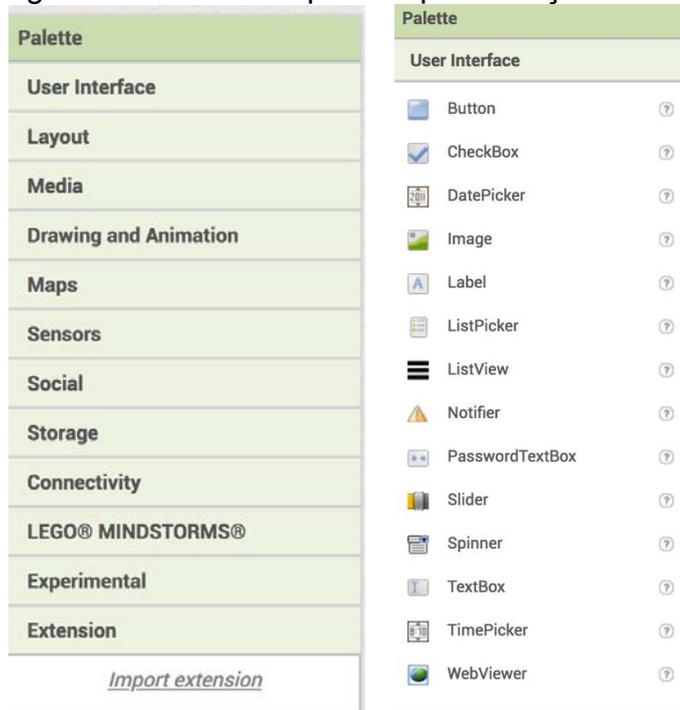


Fonte: Autor (2018)

Abaixo da opção *Components*, existe o campo *Media*, onde todos os arquivos de mídia (foto, áudio ou vídeo) que foram inseridos no projeto (através de upload no site) ficam dispostos para leitura. Ou seja, tal espaço auxilia no acompanhamento dos arquivos que compõem o projeto. No lado esquerdo da página esta a opção *Palette*, que nada mais é que um índice, onde é possível selecionar o tipo de objeto que se deseja utilizar e implementar na tela. Para criar um botão, por exemplo, basta selecionar o *User Interface*, no qual é aberto uma extensa lista de componentes disponíveis para seleção, e pressionar o campo *Button*. Em seguida, o mesmo é arrastado através do mouse até a tela de trabalho (onde ocorre a visualização).

Para que os elementos selecionados fiquem organizados na tela, o site disponibiliza um campo *layout*, no qual é possível utilizar diferentes arranjos que configuram a distribuição dos objetos. Tal ferramenta só permite a montagem da tela na horizontal ou vertical. A Fig. 18 ilustra o índice de opções criação das telas.

Figura 18 – Índice disponível para criação da tela



Fonte: Autor (2019)

3.3.1.1 Tela do menu inicial

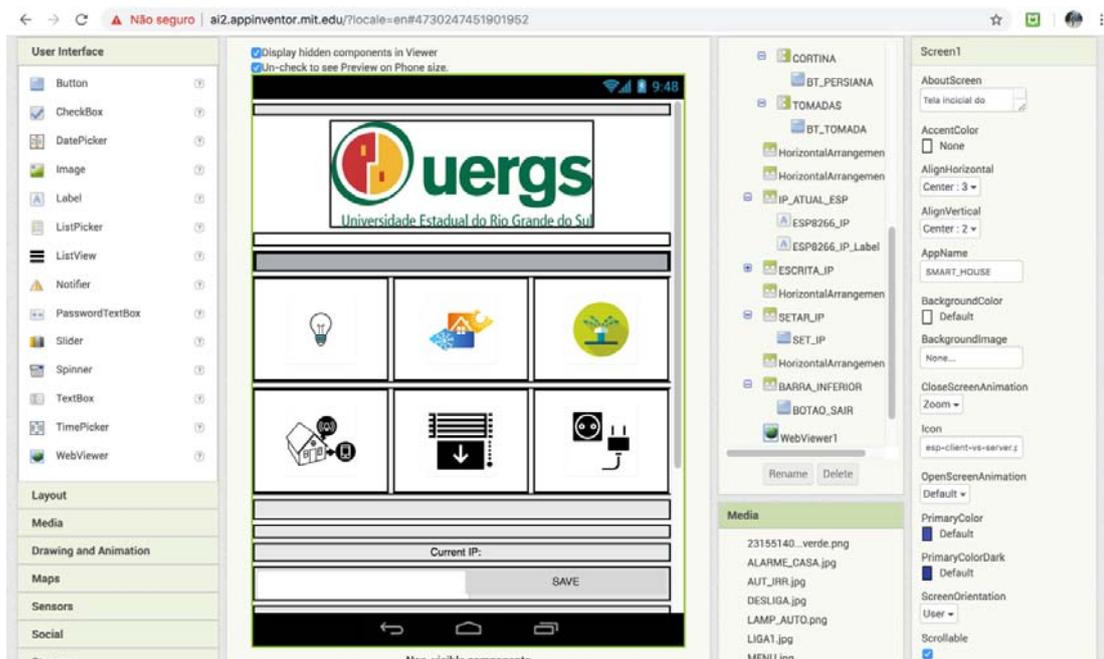
A tela inicial foi desenvolvida de forma que o usuário tenha um menu de seleção intuitivo e de fácil uso. Para isso, foram utilizadas imagens na mesma, de

fácil interpretação, que representam diferentes finalidades, conforme ilustrado na Fig. 19. Como exemplo, há uma lâmpada que ao ser pressionada leva o usuário à outra tela, onde é possível ligar e desligar distintos pontos de iluminação.

Além da imagem da lâmpada, estão dispostos outros atalhos que levam as seguintes telas:

- a) Tela de controle de iluminação;
- b) Tela de controle do sistema de refrigeração;
- c) Tela do sistema de irrigação do jardim;
- d) Tela do sistema de alarme;
- e) Tela de controle das cortinas;
- f) Tela de controle de tomadas.

Figura 19 – Tela inicial do aplicativo



Fonte: Autor (2019)

3.3.1.2 Tela para controle da iluminação

A tela do sistema de iluminação foi desenvolvida para controlar cinco diferentes ambientes de uma residência: sala, quarto, cozinha, banheiro e área

externa. Nela, é possível ligar ou desligar individualmente cada lâmpada e verificar o respectivo status, ou seja, o último comando enviado ao dispositivo. Também é possível programar intervalos para que o sistema ative e desative a iluminação. A tela de controle da iluminação é mostrada na Fig. 20.

Figura 20 – Tela do sistema de iluminação



Fonte: Autor (2018)

3.3.1.3 Tela para controle do sistema de temperatura

Na tela de controle de temperatura, mostrada na Fig. 21, foi inserido um botão com controle deslizante, ou seja, para escolher a temperatura desejada basta deslocar o botão da extremidade da esquerda (frio) para a da direita (quente) ou o contrário. A temperatura instantânea (medida em campo) é mostrada na caixa *Temperatura atual*.

Além desse controle, o usuário também pode inserir o valor de temperatura desejada manualmente (em °C). Para isso, é necessário apenas inserir o valor desejado na caixa *Setpoint de temperatura*.

Figura 21 – Tela do sistema de controle de temperatura



Fonte: Autor (2019)

3.3.1.4 Tela para controle do sistema de irrigação do jardim

Esta tela permite o usuário optar em controlar a irrigação no modo automático ou através do acionamento manual. No modo automático a bomba ligará quando o sensor de umidade do solo emitir sinal para o controlador. Caso o sistema esteja em modo automático é possível saber o status da bomba de água. A tela de controle da irrigação é mostrada na Fig. 22.

Figura 22 – Tela do sistema de irrigação

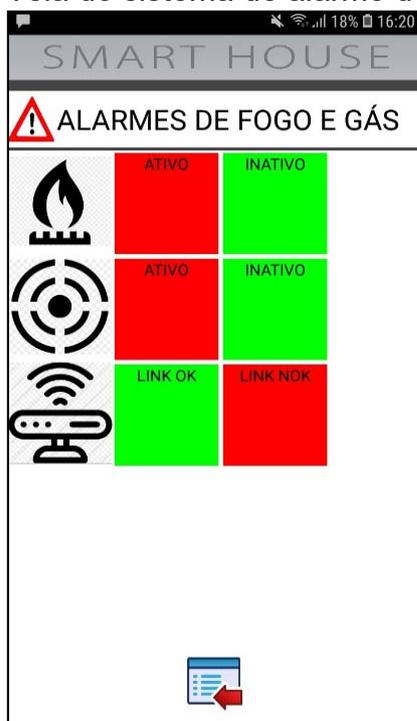


Fonte: Autor (2019)

3.3.1.5 Tela do sistema de alarme de presença de gás ou fogo

Através desta tela é possível acessar o status dos alarmes de gás e de chama. Além disso, o usuário poderá verificar se o link de comunicação *Wi-Fi* com o dispositivo está funcionando. Essa função tornou-se necessária devido a se tratar de um sistema de segurança. Para isso o aplicativo, através de uma rotina de *clock*, envia requisições constantes ao servidor, que retorna com o status dos alarmes e da rede. A tela é ilustrada na Fig. 23.

Figura 23 - Tela do sistema de alarme de gás e fogo

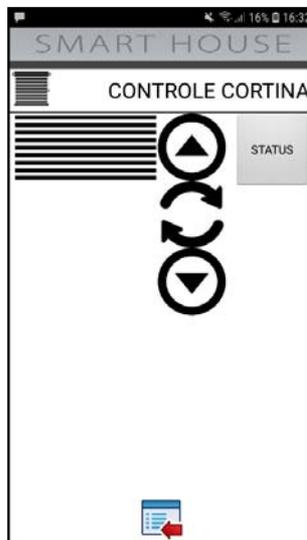


Fonte: Autor (2019)

3.3.1.6 Tela para controle de abertura da cortina

Para a tela de controle da cortina, o usuário dispõe de três botões: para cima, para baixo e confirma. Seu funcionamento é bastante simples, uma vez que o principal interesse deste controlador é de garantir a posição das cortinas. Também é possível selecionar o controle para o modo automático. O modo escolhido será mostrado na caixa *status*. A Fig. 24 mostra a tela de controle da cortina.

Figura 24 – Tela do controle das cortinas



Fonte: Autor (2019)

3.3.1.7 Tela para controle das tomadas

Nesta tela o usuário poderá ativar ou desativar a tensão na tomada de energia elétrica no modo manual ou programar um intervalo para que a mesma fique ativada. A Fig. 25 ilustra a tela para controle da tomada.

Figura 25 – Tela do sistema de tomada inteligente



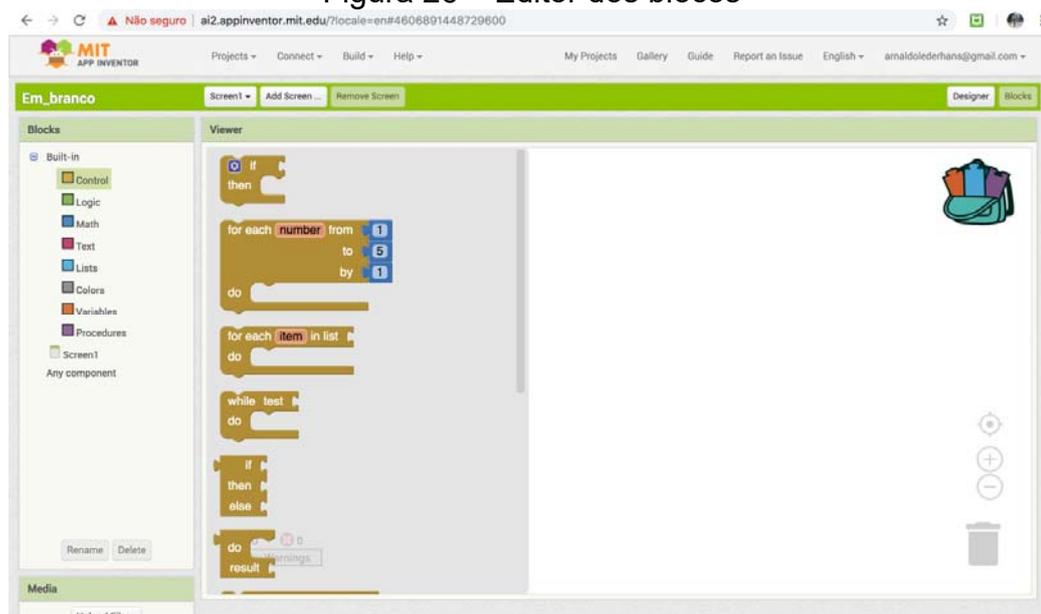
Fonte: Autor (2019)

3.3.2 Editor de blocos

O site do MIT APP INVENTOR conta com um editor de blocos, mostrado na Fig. 26, no qual é possível informar ao app como ele deve se comportar. Para isso, existe o botão *blocks* na parte superior da página que direciona o programador ao editor. Por outro lado, o botão *Designer* mantém o desenvolvedor na parte de construção gráfica do sistema.

Para cada tela foi criado um respectivo diagrama de blocos, onde cada elemento utilizado (um botão, por exemplo) é configurado e, assim, o aplicativo é capaz de interpretar e executar tais funções. Em outras palavras, esta é etapa do controle lógico do app.

Figura 26 – Editor dos blocos

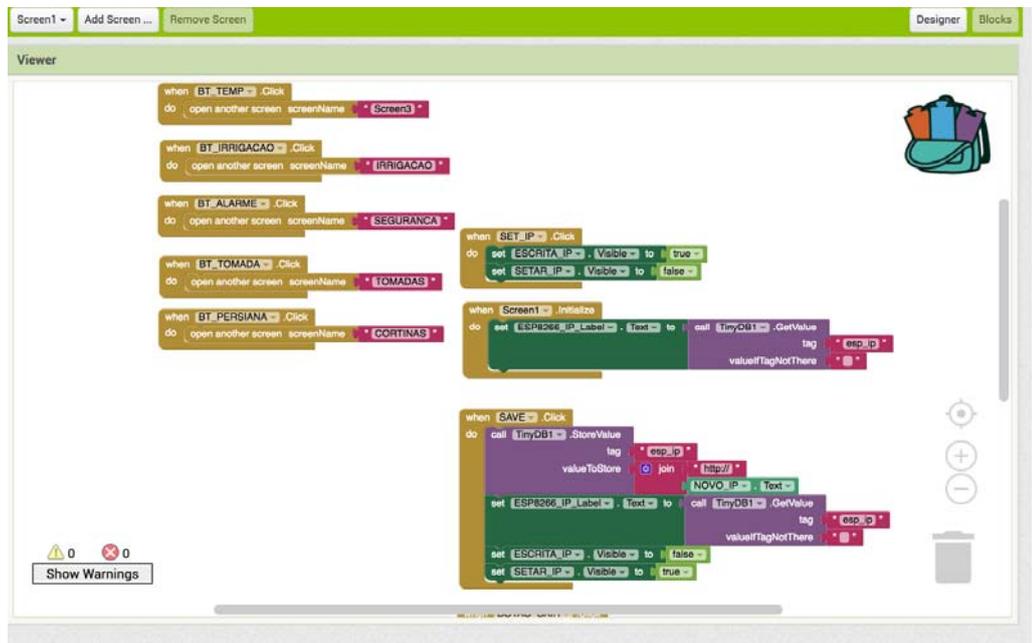


Fonte: Autor (2019)

Da mesma forma que a página de construção gráfica conta com diversas ferramentas, o editor de blocos também disponibiliza diferentes recursos para uso. Desta forma, o editor é dividido nas seguintes áreas: *Control*, *Logic*, *Math*, *Lists*, *Colors*, *Variables* e *Procedures*.

Cada componente do projeto tem seu próprio conjunto de blocos específicos para seus próprios eventos, métodos e propriedades. Na Fig. 27 é possível verificar um trecho dos blocos programados para a tela inicial do aplicativo.

Figura 27 – Ilustração de um trecho do programa da tela inicial



Fonte: Autor (2019)

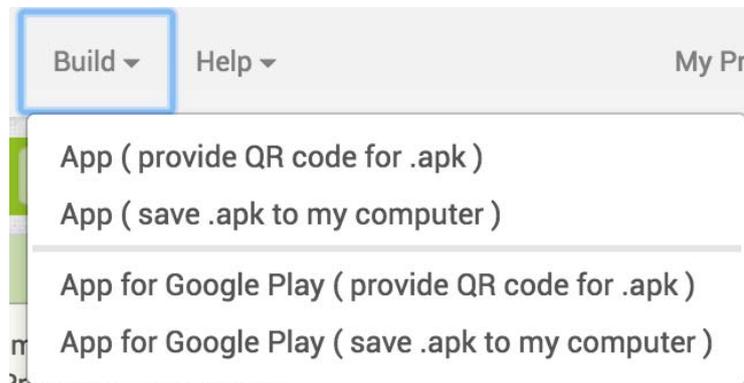
O Arduino utiliza linguagem C++ para programação das rotinas lógicas, já no MIT APP INVENTOR 2, a programação é feita através de blocos, mesmo que a função seja semelhante. Por exemplo: ambas as linguagens possuem o comando *when-do*. No entanto, no IDE da Google, a mesma é feita através de um bloco, no qual é possível selecionar a variável desejada e a subsequente ação.

3.3.3 Baixando o app para o celular

Após a programação definida, uma das etapas consistia em transferir o aplicativo em desenvolvimento no site para o *smartphone*. Para isso, foi necessário baixar dois apps na Google Play para o *smartphone*: o próprio da MIT e um visualizador de QR Code.

Na página do projeto, há o botão *Build* na parte superior da tela. Ao pressioná-lo, aparecem diversas opções para se compilar e gerar um executável no celular. Fica a critério do desenvolvedor a escolha da alternativa que melhor atenda as suas necessidades, mas todas têm a mesma finalidade e funcionam corretamente. As opções de compilação do aplicativo são mostradas na Fig. 28.

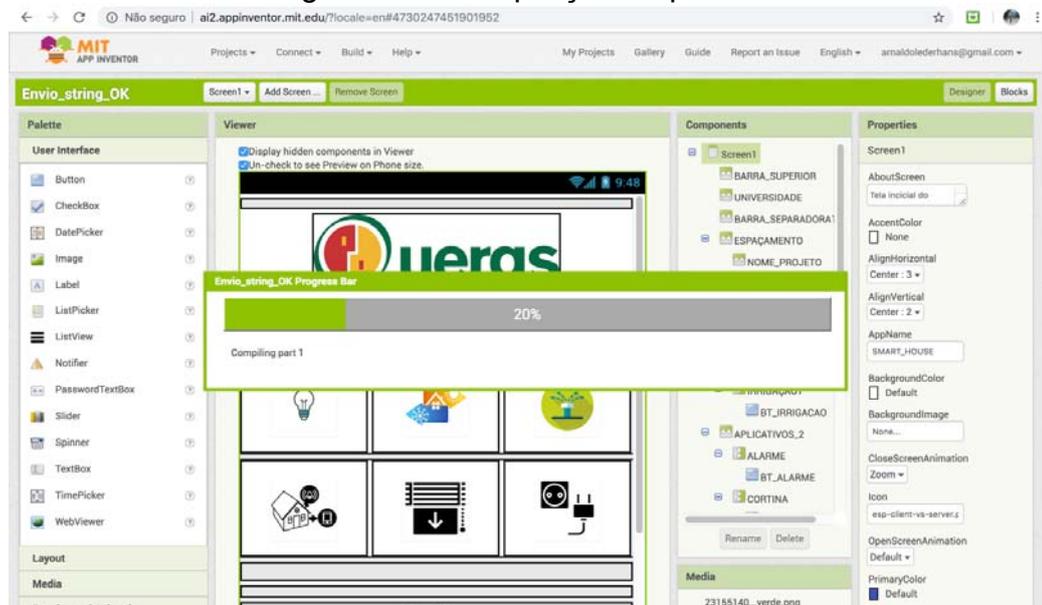
Figura 28 – Opções de compilação do aplicativo



Fonte: Autor (2019)

No presente projeto, optou-se por gerar um QR Code e, a partir de um visualizador no *smartphone*, transferir o programa para o celular. Tal medida é bastante prática e simples, no entanto, pode levar alguns minutos para processar conforme o tamanho e a complexidade do projeto. O andamento do processo de transferência é mostrado na Fig. 29.

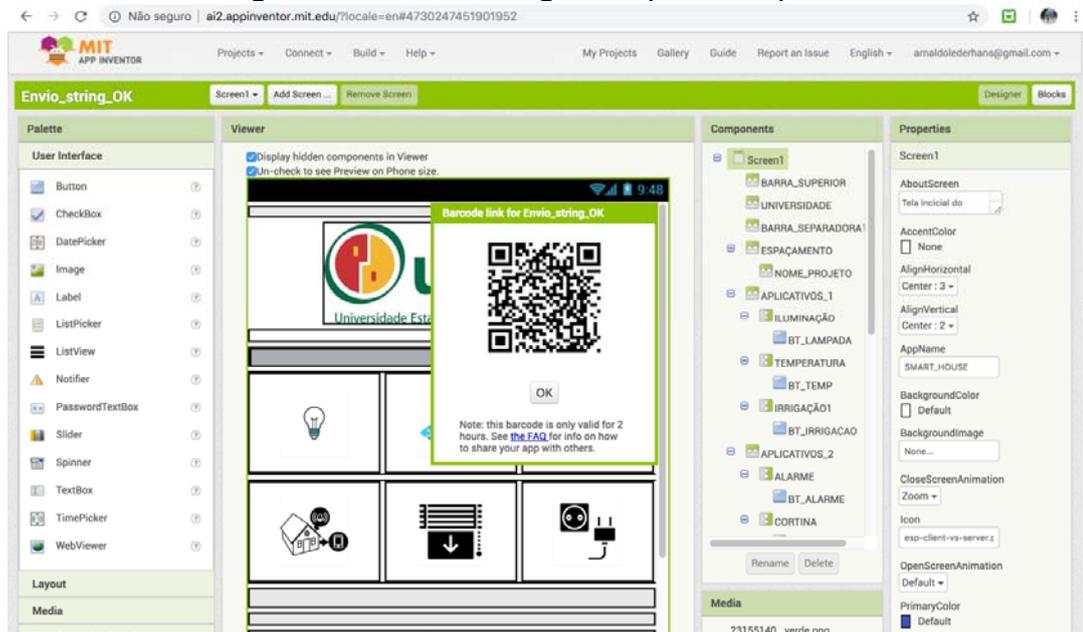
Figura 29 – Compilação o aplicativo



Fonte: Autor (2019)

Vale ressaltar que para concluir a transferência do app para o dispositivo móvel é necessário possuir um aplicativo desenvolvido pela MIT ou possuir um leitor de QR Code no celular. A Fig. 30 ilustra o código QR gerado pelo compilador.

Figura 30 – QR Code gerado pelo compilador



Fonte: Autor (2019)

3.4 CONEXÃO DOS SISTEMAS

Para que o Microcontrolador fosse conectado a rede *Wi-Fi*, todos os dispositivos ESP8266 foram programados e informados do nome da rede e a respectiva senha nas quais estavam inseridos.

Caso os dispositivos fossem instalados em outra rede, tais dados deveriam ser atualizados conforme a mesma. É possível, através da instalação de bibliotecas e rotinas de *software* a execução de atualizações do programa previamente compilado e carregado, bem como a alteração de usuário e senha da rede *Wi-Fi* utilizando a própria rede sem fio. Essas rotinas facilitam para o usuário, uma vez que não é necessário desinstalar os controladores das caixas elétricas em caso de uma eventual atualização. A Fig. 31 mostra as configurações de IP na IDE do Arduino.

Figura 31 – Configurações do Arduino



```

ENVIO_STRING_APPINVENTOR_OK_DHT22
//ARNALDO LEDERHANS
//PROGRAMA PARA SERVER E CLIENT ESP8266 NODEMCU COM APPINVENTOR
//AMOSTRAGEM DE TEMPERATURA COM DHT22

#include <DHT.h>
#include <Adafruit_AM2320.h>
#define DHTPIN 4 // what digital pin the DHT22 is connected to
#define DHTTYPE DHT22 // there are multiple kinds of DHT sensors

DHT dht(DHTPIN, DHTTYPE); //DO PINO USADO E DO TIPO DE DHT

#include <Crescer.h> //biblioteca para usar o temporizador sem delay
#include <ESP8266WiFi.h>

Tempora tempCon; // tempora é o tipo de variavel para o temporizador
Tempora Le_analogica;
//const char* ssid = "Fernanda";
//const char* password = "20630808";

const char* ssid = "Arnaldo";
const char* password = "20630808";

//const char* ssid = "VIVOFIBRA-057C";
//const char* password = "wE8FFRpvqK";

int analogica;

//PARA FORÇAR UM IP RETIRAR OS COMENTÁRIOS DAS LINHAS ABAIXO, CASO CONTRÁRIO, VERIFICAR O IP QUE SERÁ DESTINADO AO ESP12 NA SERIAL
//IPAddress ip(192, 168, 15, 31); //mudar o ultimo digito
//IPAddress wifi(192, 168, 15, 1);

```

Fonte: Autor (2019)

Para o caso de os dados da rede e da senha serem preenchidos erroneamente, o IDE não informa tal erro, uma vez que a compilação ocorre corretamente e o programa é transferido para o *hardware*. A falha só é percebida no momento em que se vai acionar uma saída ou realizar a leitura de uma variável e a resposta é nula. Neste caso, deve se verificar os dados inseridos para cada microcontrolador e corrigi-los para garantir o correto funcionamento do sistema.

3.4.1 Configurando Entradas e Saídas

Diversas funções e comandos foram utilizados nos controladores, uma vez que havia a necessidade de se realizar distintas aplicações. Como exemplo, podemos citar o uso do conversor AD (analógico para digital) para aferir a temperatura de um dos ambientes, conforme pode ser visualizado na Fig. 20.

Além disso, foram utilizadas diversas IO's dos microcontroladores: ora para ler o status de uma entrada, ora pra acionar uma saída. Como exemplo, tem-se o controle das lâmpadas da casa, bem como o estado das mesmas. Para facilitar a programação tanto na IDE do Arduino, quanto no App Inventor, foi construída a tabela 2, incluindo as entradas e saídas dos módulos de controle.

Tabela 2 - Descrição das entradas e saídas de cada controle

I/O	Tag IDE	Sinal sem atuação	Saída (VDC)	Tensão de trabalho	ENTRADA/SAÍDA UTILIZADA					
					Iluminação	Temperat	Cortina	Alarme	Tomada	Irrigação
Presença	PIR	HIGH	3.3	5VDC	A0					
Luminosidade	LDR	HIGH	5	5VDC	16		A0			
Umidade	UMI	HIGH	5	5VDC						A0
Chuva	CHU	HIGH	5	5VDC			16			
Temperatura	TMP	NA	NA	5VDC		16				
Gás	GAS	HIGH	5	5VDC				A0		
Chama	FOG	HIGH	5	5VDC				16		
Motor DC	MOT	OFF		5VDC			PONTE H			
Bomba d'água	BDA	OFF		5VDC						RELÉ 1
Buzzer	BUZ	OFF		5VDC				RELÉ 1		
Válvula	VAL	CLOSE		110VCA				RELÉ 2 *		
Lâmpada	LAM	OFF		110-220	RELÉ 1					
Micro ventilador	VEM	OFF		12VCD		RELÉ 1				
RELÉ 1	RL1	OPEN		5VDC	2	2		2	2	2
RELÉ 2	RL2	OPEN		5VDC	15	15		15	15	15

* SEM TENSÃO CONTATO ABERTO

Fonte: Autor (2019)

3.4.2 Endereços de IP dos dispositivos

Para que ocorresse a comunicação entre o Arduino e o aplicativo, foi necessário atribuir endereços fixos de IP para cada dispositivo do sistema, conforme mostrado na tabela 1. Assim, o aplicativo, que neste caso é tratado como cliente da rede, envia o comando específico para o controlador através de uma função chamada *WebView*. O servidor (controlador) envia a resposta que é processada pelo aplicativo através da função *WebViewString*. Um trecho do programa que realiza a conexão é mostrado na Fig. 32.

Figura 32 – Obtendo o endereço IP do Arduino

```

1 //UNIVERSIDADE ESTADUAL DO RIO GRANDE DO SUL - CURSO SUPERIOR DE TECNOLOGIA EM AUTOMAÇÃO INDUSTRIAL
2 //CÓDIGO FONTE PARA CONTROLE DE LÂMPADA - SALA
3
4 #include <ESP8266WiFi.h> //INCLUSÃO DA BIBLIOTECA NECESSÁRIA PARA FUNCIONAMENTO DO CÓDIGO
5
6 const char* ssid      = "Redmi";
7 const char* password = "123456";
8
9
10 //DEFINIÇÃO DE IP FIXO PARA O NODEMCU
11 IPAddress ip(192,168,0,25); //COLOQUE UMA FAIXA DE IP DISPONÍVEL DO SEU ROTEADOR. EX: 192.168.1.110 **** ISSO VARIA, NO MEU CASO É: 192.168.0.175
12 IPAddress gateway(192,168,0,1); //GATEWAY DE CONEXÃO (ALTERE PARA O GATEWAY DO SEU ROTEADOR)
13 IPAddress subnet(255,255,255,0); //MASCARA DE REDE
14
15 WiFiServer server(80); //CASO OCORRA PROBLEMAS COM A PORTA 80, UTILIZE OUTRA (EX:8082,8089) E A CHAMADA DA URL FICARÁ IP:PORTA(EX: 192.168.0.15:8082)
16
17 // Create an instance of the server
18 // specify the port to listen on as an argument
19
20 void setup() {
21   Serial.begin(115200);
22   delay(10);
23
24   // PREPARAÇÃO DOS PINOS DE ENTRADA E SAÍDA

```

Fonte: Autor (2019)

4 RESULTADOS

Finalizada a programação do aplicativo e dos dispositivos, chegou o momento de realizar a comunicação entre os mesmos e avaliar os resultados obtidos. Além disso, foi necessário analisar o desempenho individual de cada componente que compõe o presente sistema. Como o *software* MIT AI2 Companion foi instalado no celular, ao ler o código QR gerado no site, este aplicativo gerou um novo arquivo no qual continha toda a programação realizada no site da Google. O atalho do aplicativo pode ser constatado na Fig. 33, onde na parte superior, há o ícone *Smart House*.

Figura 33 – Smartphone com os aplicativos instalados



Fonte: Autor (2019)

Ao abrir o aplicativo *Smart House*, a tela principal do aplicativo é executada. Durante os testes todos os botões e atalhos foram testados e funcionaram corretamente. Além disso, a interação entre as telas também foi devidamente averiguada e o projeto se comportou conforme o esperado. A Fig. 34 mostra a tela que contém o menu principal do aplicativo.

Figura 34 – Tela do menu principal do aplicativo no celular



Fonte: Autor (2019)

Após realizada a análise do aplicativo, foi avaliada a comunicação entre o mesmo e o *hardware*. As rotinas de programação na IDE do Arduino e no aplicativo dos controles de iluminação, cortina, irrigação e tomadas são basicamente as mesmas, o que facilitou os testes. A integração do aplicativo com o microcontrolador ESP8266 para estes controles funcionou corretamente, tanto para o envio quanto para o recebimento de dados. Isso foi comprovado através de testes práticos com as montagens de todos os circuitos em matriz de contatos e acionamento do botões nas diferentes telas do aplicativo. Os dados enviados do aplicativo para o microcontrolador, bem como os dados recebidos, tratados e mostrados pelo aplicativo, podem ser visualizados na tabela 3.

Tabela 3 – Resumo dos testes realizados de forma efetiva

		Mensagem HTTP enviada			Mensagem (string) recebida			
Tela	IP	Botão Ligar	Botão Desligar	Botão automático	Status Ligado	Status Desligado	Status Automático	Status Automático
Controle da iluminação (Sala)	192.168.1.25	/RL1/ON	/RL1/OFF	/SET/AUT	ligadoON	ligadoOF	automaON	automaOF
Controle da iluminação (Quarto)	192.168.1.26	/RL1/ON	/RL1/OFF	/SET/AUT	ligadoON	ligadoOF	automaON	automaOF
Controle da iluminação (Cozinha)	192.168.1.27	/RL1/ON	/RL1/OFF	/SET/AUT	ligadoON	ligadoOF	automaON	automaOF
Controle da iluminação (Banheiro)	192.168.1.28	/RL1/ON	/RL1/OFF	/SET/AUT	ligadoON	ligadoOF	automaON	automaOF
Controle da cortina (Externa)	192.168.1.29	/RL1/ON	/RL1/OFF	/SET/AUT	ligadoON	ligadoOF	automaON	automaOF
Sistema de irrigação do jardim	192.168.1.31	/RL1/ON	/RL1/OFF	/SET/AUT	ligadoON	ligadoOF	automaON	automaOF
Controle da cortina	192.168.1.33	/RL1/ON	/RL1/OFF	/SET/AUT	ligadoON	ligadoOF	automaON	automaOF
Controle da tomada	192.168.1.34	/RL1/ON	/RL1/OFF	/SET/AUT	ligadoON	ligadoOF	automaON	automaOF
		Mensagem HTTP enviada			Mensagem (string) recebida			
Tela	IP	Botão Ativar	Botão Desativar	Ativo com alarme	Ativo sem alarme	Desativado		
Sistema de alarme de gás	192.168.1.32	/GAS/ON	/GAS/OFF	gasatiON	gasatiOF	gasdesat		
Sistema de alarme chama	192.168.1.32	/FOG/ON	/FOG/OFF	fogatiON	fogatiOF	fogdesat		
		Mensagem HTTP enviada			Mensagem (string) recebida			
Tela	IP	Timer ligar	Timer desligar	SetPoint	Status Ligado	Status Desligado	Status Automático	Temperatura
Controle de temperatura	192.168.1.30	/TMP/ON	/TMP/OFF	Valor de 20.0 a 3	ligadoON	ligadoOF	automaON	automaOF "float temperatura"

Fonte: Autor (2019)

O envio do valor de temperatura desejada (*setpoint*) do aplicativo para o controlador, bem como a atualização da temperatura atual na tela de controle de temperatura do aplicativo ocorriam somente quando a tela era ativada, ou seja, constatou-se um problema na amostragem dos valores reais. Para resolver tal erro, foi necessário a implementação de uma rotina de *clock* no aplicativo para atualizar o envio e recebimento das mensagens e um intervalo de 1 segundo.

Foram realizados testes de acionamento dos botões do aplicativo e simulações dos diferentes estados lógicos dos sensores e dos microcontroladores enviando dados para as telas do aplicativo, tendo o funcionamento dos circuitos comprovado de forma efetiva.

5 CONCLUSÃO

Diante de um cenário onde as pessoas estão cada vez mais conectadas e do surgimento de novas tecnologias, desenvolver um projeto que forneça algo relevante para o dia-a-dia do indivíduo é algo extremamente desafiador. O presente projeto surgiu primeiramente com o intuito de fornecer algo simples e prático no qual o usuário pudesse implementar o seu próprio sistema de automação residencial.

Para o desenvolvimento do trabalho, os desafios foram vários, principalmente quanto ao estabelecimento da comunicação entre o Arduino e o aplicativo, programados pelo autor, via rede *Wi-Fi*. O Arduino é uma plataforma amplamente conhecida e utilizado mundialmente, possuindo inúmeros exemplos e códigos na internet. No entanto, desenvolver um sistema de interface com um *Smartphone* é algo incomum nos dias atuais. Além disso, a grande maioria dos trabalhos e referências encontradas na internet utilizam a comunicação pelo Bluetooth.

Sendo assim, com base nos desafios citados anteriormente, da complexidade do projeto e da sua proposta de facilitar e trazer ganhos a vida do consumidor, pode se afirmar que os resultados foram extremamente satisfatórios. Desenvolver um projeto onde a pessoa possa interagir com os principais equipamentos da sua residência é algo bastante útil e prático nos dias atuais. Além disso, vale ressaltar a facilidade de se instalar os dispositivos sem alteração significativa na rede elétrica do imóvel, uma vez que toda a comunicação é feita sem fio.

O presente trabalho de conclusão de curso também permitiu a aplicação de importantes áreas estudadas no tecnólogo, como programação, eletrônica digital, eletrônica analógica, entre outras. Além disso, foi possível adquirir novos conhecimentos e uma nova plataforma de programação (desenvolvida pela Google), baseada em blocos, foi desenvolvida.

Por fim, ficam oportunidades para implementações futuras e possíveis melhorias, como:

- a) Utilização de outros microcontroladores;
- b) Implementação de interface via *bluetooth* e comparação de desempenho em relação a rede *Wi-Fi*;
- c) Desenvolvimento de placa padrão para uso na instalação;
- d) Desenvolvimento de aplicativo para outros sistemas operacionais, como

- iOs (da Apple) e Windows Phone (da Microsoft);
- e) Desenvolvimento de app a partir de outras plataformas de programação, como o Blynk;
- f) Aplicação de recursos de comunicação baseado em MQTT.

REFERÊNCIAS

- ALVES, Gilvani. **O emprego de técnicas de qualidade de serviço em redes sem fio aplicado à automação predial**. 2016. 102 f. Dissertação (Mestrado) - Curso de Engenharia Elétrica, Faculdade de Engenharia, Universidade Estadual Paulista, Ilha Solteira, 2016. Disponível em: <<https://repositorio.unesp.br/handle/11449/137898>>. Acesso em: 12 mar. 2019.
- APPINVENTOR (Estados Unidos). **Qualquer um pode criar aplicativos que impactam o mundo**. 2019. Disponível em: <<https://appinventor.mit.edu/explore/about-us.html>>. Acesso em: 16 abr. 2019.
- ARDUINO. **Arduino Software (IDE)**. 2019. Disponível em: <<https://www.arduino.cc/en/main/software>>. Acesso em: 08 mar. 2019.
- ARDUINO: The Documentary. Direção de Rodrigo Calvo. Produção de Raúl Alejos. New York: Laboral Centro de Arte y Creación Industrial, 2010. Son., color. Legendado. Disponível em: <<https://vimeo.com>>. Acesso em: 08 mar. 2019.
- AURESIDE (São Paulo). **Previsões para o mercado global de Automação Residencial**. 2016. Disponível em: <<http://aureside.com.br/>>. Acesso em: 11 mar. 2019.
- BATRINU, Catalin. **Projetos de Automação Residencial com ESP8266**. São Paulo: Novatec, 2018. 172 p. (9788575226803).
- BIONDO, Rodrigo Mingoelli. **Domótica: Sistemas e aplicabilidade**: Domótica: sistemas de aplicabilidade. 2011. 58 f. Monografia (Especialização) - Curso de Engenharia Elétrica, Escola de Engenharia, Universidade de São Paulo, São Carlos, 2011. Cap. 1. Disponível em: <http://www.tcc.sc.usp.br/tce/disponiveis/18/180450/tce-19102011-122815/?&lang=br>>. Acesso em: 12 ago. 2018.
- BOLZANI, Caio Augustus Moraes. **Análise de Arquiteturas e Desenvolvimento de uma Plataforma para Residências Inteligentes**. 2010. 155 f. Tese (Doutorado) - Curso de Engenharia de Sistemas Eletrônicos, Escola Politécnica, Universidade de São Paulo, São Paulo, 2010. Disponível em: <<https://www.teses.usp.br>>. Acesso em: 12 abr. 2019.
- BOLZANI, Caio Augustus Moraes. **Residências Inteligentes**. São Paulo: Livraria da Física, 2004. 332 p. (858832525X).
- COUTINHO, Gustavo Leuzinger. **A era dos Smartphones**: Um estudo Exploratório sobre o uso dos Smartphones no Brasil. 2014. 66 f. TCC (Graduação) - Curso de Comunicação Social, Faculdade de Comunicação, Universidade de Brasília, Brasília, 2014. Disponível em: <<http://bdm.unb.br/handle/10483/9405>>. Acesso em: 12 abr. 2019.
- DAVIS, Harold. **Absolute Beginner's Guide to Wi-Fi Wireless**

Networking. Indianápolis: Que Corp., 2004. 312 p.
MCROBERTS, Michael. **Arduino Básico.** São Paulo: Novatec, 2011. 456 p.
(9788575222744).

ESTADÃO: **Brasileiro gasta 200 minutos por dia em aplicativos, diz estudo.**
2018. Disponível em: <<https://exame.abril.com.br/tecnologia/brasileiro-gasta-200-minutos-por-dia-em-aplicativos-diz-estudo>>. Acesso em: 27 nov. 2018.

MIND COMMERCE (Estados Unidos) (Ed.). **WLAN 2012: Wireless Local Area Network Market Study & Business Overview.** Seattle: Mind Commerce, 2012. 100 p.
Disponível em: <<https://eds.a.ebscohost.com/eds/ebookviewer/ebook/>>. Acesso em: 21 maio 2019.

MONTICO, Matias (Ed.). **Guia avançado de redes wireless.** São Paulo: Digerati Books, 2009. 112 p. Disponível em: <<https://books.google.com.br>>. Acesso em: 12 mar. 2019.

MORAES, Alexandre Fernandes de. **Redes sem Fio: Instalação, Configuração e Segurança.** São Paulo: Érica, 2010. 284 p. (9788536503158).

MURATORI, José Roberto; BÓ, Paulo Henrique dal. **Automação Residencial - Conceitos e Aplicações.** 2. ed. Belo Horizonte: Educere, 2014. 200 p. (978-85-65641-02-9).

MURATORI, José Roberto; BÓ, Paulo Henrique dal. **Automação residencial: histórico, definições e conceitos.** 2011. Disponível em: <<https://www.osetoreletrico.com.br/capitulo-i-automacao-residencial-historico-definicoes-e-conceitos/>>. Acesso em: 26 nov. 2018.

OLIVEIRA, Sérgio. **Internet das Coisas com ESP8266, Arduino e Raspberry Pi.** São Paulo: Novatec, 2017. 240 p. (9788575225813).

PALMA, Luciano; PRATES, Rubens. **TCP/IP: Guia de Consulta Rápida.** São Paulo: Novatec, 2000. 128 p.