

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ  
DEPARTAMENTO ACADÊMICO DE ELETROTÉCNICA  
CURSO DE ENGENHARIA INDUSTRIAL ELÉTRICA

CLEBER BENEDITO SAVI DA SILVA  
DANIEL RIBEIRO NARDOTO  
MARCELLO DE MELO JORGE

**DESENVOLVIMENTO DE UM SISTEMA MODULAR  
MICROCONTROLADO DE AUTOMAÇÃO RESIDENCIAL VIA WI-FI**

TRABALHO DE CONCLUSÃO DE CURSO

CURITIBA  
2016

CLEBER BENEDITO SAVI DA SILVA  
DANIEL RIBEIRO NARDOTO  
MARCELLO DE MELO JORGE

**DESENVOLVIMENTO DE UM SISTEMA MODULAR  
MICROCONTROLADO DE AUTOMAÇÃO RESIDENCIAL VIA WI-FI**

Trabalho de Conclusão de Curso, apresentado ao Curso Superior de Engenharia Industrial Elétrica, ênfase Eletrotécnica do Departamento Acadêmico de Eletrotécnica da Universidade Tecnológica Federal do Paraná – UTFPR, como requisito parcial para obtenção do título de Engenheiro Eletricista.

Orientador: Prof. Dr. Amauri Amorim Assef

CURITIBA

2016

**Cleber Benedito Savi da Silva**  
**Marcello de Melo Jorge**  
**Daniel Ribeiro Nardoto**

## **Desenvolvimento de um sistema modular microcontrolado de automação residencial via WI - FI**

Este Trabalho de Conclusão de Curso de Graduação foi julgado e aprovado como requisito parcial para a obtenção do Título de Engenheiro Eletricista, do curso de Engenharia Elétrica do Departamento Acadêmico de Eletrotécnica (DAELT) da Universidade Tecnológica Federal do Paraná (UTFPR).

Curitiba, 01 de dezembro de 2016.

---

Prof. Emerson Rigoni, Dr.  
Coordenador de Curso  
Engenharia Elétrica

---

Profa. Annemarlen Gehrke Castagna, Mestre  
Responsável pelos Trabalhos de Conclusão de Curso  
de Engenharia Elétrica do DAELT

### **ORIENTAÇÃO**

---

Amauri Amorin Assef, Dr.  
Universidade Tecnológica Federal do Paraná  
Orientador

### **BANCA EXAMINADORA**

---

Amauri Amorin Assef, Dr.  
Universidade Tecnológica Federal do Paraná

---

Mariana Antonia Aguiar Furucho, M.Sc.  
Universidade Tecnológica Federal do Paraná

---

Ismael Chiamenti, Dr.  
Universidade Tecnológica Federal do Paraná

A folha de aprovação assinada encontra-se na Coordenação do Curso de Engenharia Elétrica

## **AGRADECIMENTOS**

Primeiramente a Deus, pois permitiu que tudo isso acontecesse, ao longo de nossas vidas, e não somente nestes anos como universitários, mas que em todos os momentos, é o maior mestre que alguém pode conhecer.

A esta universidade, seu corpo docente, direção e administração que oportunizaram a janela que hoje vislumbramos um horizonte superior, eivado pela acendrada confiança no mérito e ética aqui presentes.

Ao Prof. Dr. Amauri Amorin Assef pela oportunidade e apoio na elaboração deste trabalho.

Aos nossos pais, pelo amor, incentivo e apoio incondicional.

A todos que direta ou indiretamente fizeram parte da nossa formação, o nosso muito obrigado.

## RESUMO

SILVA, Cleber Benedito Savi; NORDOTO, Daniel Ribeiro; JORGE, Marcello de Melo. Desenvolvimento de um sistema modular micro controlado de automação residencial via Wi-Fi. 2016. 57 f. Trabalho de Conclusão de Curso – Bacharelado em Engenharia Industrial Elétrica, Departamento Acadêmico de Eletrotécnica, Universidade Tecnológica Federal do Paraná. Curitiba, 2016.

O presente trabalho apresenta a implementação e especificação de um protótipo para controle de automação residencial à distância, utilizando um roteador Wi-Fi, aparelho celular com sistema operacional Androide e um sistema micro controlado Arduino. O conjunto de dispositivos utilizados tem como objetivo automatizar a iluminação de uma residência, apresentando facilidade de interação para usuários deficientes ou não. De uma forma muito moderna, o *smartphone* poderá controlar toda a iluminação de uma residência através de simples toques na tela utilizando a comunicação Wi-Fi. Para tanto, fizeram-se necessários um aplicativo com uma interface gráfica personalizada, o sistema de *hardware* (Arduino) aliado a placa de acionamento de carga e o *software* livre (*Processing*), que foi utilizado na leitura dos dados que serão enviados do celular pela rede Wi-Fi e recebidos pelo microcontrolador. Este, por sua vez, encaminhará as sequências de ações necessárias para a realização das funções desejadas, ou seja, ligar ou desligar lâmpadas de um determinado cômodo de uma residência.

**Palavras-chave:** Automação Residencial, Comunicação Wi-Fi, Arduino.

## ABSTRACT

SILVA, Cleber Benedito Savi; NORDOTO, Daniel Ribeiro; JORGE, Marcello de Melo. Development of a modular microcontrolled system for home automation using Wi-Fi. 2016. 57 f. Trabalho de Conclusão de Curso – Bacharelado em Engenharia Industrial Elétrica, Departamento Acadêmico de Eletrotécnica, Universidade Tecnológica Federal do Paraná. Curitiba, 2016.

This research is about implementation and specification of a home automation prototype, remotely controlled using a Wi-Fi router, a cellphone with Android operational system and an Arduino microcontrolled system. The used device set the automation of system the lighting of a house, featuring ease of interaction to users who are disabled or not. In a very modern way, smartphones will be able to control the entire lighting of a house through simple touches on the screen, using Wi-Fi system. The apps required were customized with a personalized graphical interface, a hardware (Arduino) system together with a load drive, and the free software (Processing), used in the reading of data to be sent from the mobile to the Wi-Fi network and received in the microcontroller. This microcontroller will perform the required action sequence for perform the desired function, that is, the act of tuning or and off the lights of a certain room on a house.

**Keywords:** Home Automation, Wi-Fi Communication, Arduino.

## LISTA DE SIGLAS, ABREVIATURAS E ACRÔNIMOS

ANATEL	Agência Nacional de Telecomunicações
CLP	Controlador Lógico Programável
CPU	Unidade Central de Processamento
EAP	<i>Extensible Authentication Protocol</i>
FAW	<i>Future Audio Workshop</i>
HTML	<i>Hypertext Markup Language</i>
IA	Inteligência artificial
IBGE	Instituto Brasileiro de Geografia e Estatística
IDE	<i>Integrated Development Environment</i>
IEEE	<i>Institute of Electrical and Electronics Engineers</i>
MIDI	<i>Musical Instrument Digital Interface</i>
PC	<i>Personal Computer</i>
PLC	<i>Power Line Carrier</i>
PNE	Portador de necessidades especiais
PWM	<i>Pulse Width Modulation</i>
SD	<i>Secure Digital</i>
TCC	Trabalho de Conclusão de Curso
TKIP	<i>Temporal Key Integrity Protocol</i>
USB	<i>Universal Serial Bus</i>
WEP	<i>Wired Equivalent Privacy</i>
WI-FI	<i>Wireless Fidelity</i>
WLAN	<i>Wireless Local Area Network</i>
WPA	<i>Wi-Fi Protected Access</i>
WPAN	<i>Wireless Personal Area Network</i>
WWAN	<i>Wireless Wide Area Network</i>

## LISTA DE ILUSTRAÇÕES

FIGURA 1 – SISTEMA PROPOSTO. ....	12
FIGURA 2 – DIAGRAMA DE BLOCOS DA PROGRAMAÇÃO DO ARDUINO. ....	29
FIGURA 3 – PLACA ARDUINO UNO R3.....	30
FIGURA 4 – <i>SHIELD ARDUINO ETHERNET</i> . ....	33
FIGURA 5 – PLATAFORMA DE PROGRAMAÇÃO ARDUINO EM LINGUAGEM “C”.....	34
FIGURA 6 – INTERFACE TOUCHOSC. ....	35
FIGURA 7 – AMBIENTE DE PERSONALIZAÇÃO TOUCHOSC. ....	36
FIGURA 8 – APLICATIVO EM UM CELULAR ANDROID. ....	36
FIGURA 9 – PLACAS ARDUINO UNO COM O <i>SHIELD ARDUINO ETHERNET</i> . ....	37
FIGURA 10 – PLACAS ARDUINO UNO E <i>SHIELD ARDUINO ETHERNET</i> CONECTADAS. ....	38
FIGURA 11 – DIAGRAMA DE FUNCIONAMENTO DO RELÉ. ....	39
FIGURA 12 – ACIONAMENTO DO RELÉ. ....	39
FIGURA 13 – PLACA POTÊNCIA.....	40
FIGURA 14 – PLACA ADAPTADORA DE CONEXÕES. ....	41
FIGURA 15 – PLACA DE ATUADORES. ....	42
FIGURA 16 – ADAPTADOR.....	43
FIGURA 17 – DIAGRAMA EM BLOCOS DO MODELO DE COMANDOS. ....	44
FIGURA 18 – DIAGRAMA EM BLOCOS DA COMUNICAÇÃO USUÁRIO E PLACA ARDUINO. ...	44
FIGURA 19 – DIAGRAMA DE FUNCIONAMENTO.....	47
FIGURA 20 – QUADROS DO PROGRAMA TOUCHOSC.....	48
FIGURA 21 – DIAGRAMA DE ACIONAMENTO DE RELÉ.....	50



## LISTA DE TABELAS

TABELA 1 – COMPARATIVO DE AUTOMAÇÃO .....	15
TABELA 2 – GERAÇÕES DA TELEFONA CELULAR.....	21
TABELA 3 – COMPARATIVOS ENTRE DIFERENTES MICROCONTROLADORES ARDUINO.....	30

# SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	11
1.1	TEMA	12
1.1.1	DELIMITAÇÃO DO TEMA	13
1.2	PROBLEMAS E PREMISSAS	13
1.3	OBJETIVOS	14
1.3.1	OBJETIVO GERAL	14
1.3.2	OBJETIVOS ESPECÍFICOS	14
1.5	ESTRUTURA DO TRABALHO	16
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA</b>	17
2.1	AUTOMAÇÃO DE RESIDÊNCIAS	17
2.2	CONCEITOS DE AUTOMAÇÃO	18
2.3	CONCEITOS TECNOLÓGICOS	19
2.4	CONCEITOS DE COMPUTAÇÃO FÍSICA	20
2.5	CONCEITOS DE SMARTPHONE	21
2.6	CONCEITOS DE ROTEADOR WI-FI	22
2.6.1	REDES WI-FI	23
2.6.2	SEGURANÇA	25
2.7	CONCEITO DE SISTEMAS OPERACIONAIS	26
2.7.1	ANDROID	26
2.7.2	IOS	27
<b>3</b>	<b>PLATAFORMAS, PROGRAMAÇÃO E DESENVOLVIMENTO</b>	28
3.1	PLATAFORMA ARDUINO	28
3.1	DEFINIÇÃO DE SHIELDS	32
3.2	<i>SOFTWARE ARDUINO DEVELOPMENT ENVIRONMENT</i>	33
3.4	<i>SOFTWARE TOUCHOSC INTERFACE EDITOR</i>	34
3.5	INSTALAÇÃO	37
3.6	INSTALAÇÕES ELÉTRICAS E COMPONENTES	38
<b>4</b>	<b>ARQUITETURA DO SISTEMA</b>	45
4.1	PLANEJAMENTO	45
4.2	CAMADA DE COMPONENTES FÍSICOS	45

4.3	CAMADA DE TRANSDUTORES E COMUNICAÇÃO.....	46
4.4	DESCRIÇÃO DA PROGRAMAÇÃO DO ARDUINO.....	46
4.5	DESCRIÇÃO FUNCIONAL .....	46
4.6	COMPONENTES E CARACTERÍSTICAS DO PROJETO .....	48
4.7	TESTE EM LABORATÓRIO .....	50
<b>5</b>	<b>CONCLUSÕES</b> .....	<b>52</b>
5.1	TRABALHOS FUTUROS .....	53
	<b>REFERÊNCIAS</b> .....	<b>54</b>

## 1 INTRODUÇÃO

De acordo com Muratori (2011), a automação residencial é um conceito muito recente, sendo uma área que apresentou grande crescimento. As primeiras utilizações desta tecnologia datam do final da década de 70, data a qual surgiram nos Estados Unidos os primeiros controladores inteligentes, onde os comandos eram feitos pela própria rede elétrica da residência, no conceito de PLC (*Power Line Carrier*). Dentro da automação residencial, apresenta-se o conceito de Domótica, palavra que provém da junção do termo em latim *domus*, que significa casa e da palavra robótica, sendo seu objetivo o melhor controle e praticidade das rotinas e tarefas residenciais. A Domótica é um conceito de tecnologia que envolve um sistema programável integrado capaz de controlar todos os ambientes de uma residência através de um só equipamento, como, por exemplo, a luminosidade, o som, a segurança e temperatura, entre outros (BOLZANI, 2004).

Segundo González et al. (2000), não há como negar que as tecnologias condicionam o tipo de sociedade em que se vive e podem influenciar na configuração e transformação da forma de vida moderna. Ainda segundo o autor, a autonomia de um projeto pode ser entendida através de uma explicação “frankensteiniana”, quando ocorre a perda do controle da invenção pelo inventor a partir do momento que é disponibilizada a um público. Contudo, essa falta de controle por parte do criador, não transforma o produto em algo autônomo, apenas mostra que seu desenvolvimento integra sistemas com diversas configurações e possibilidades. Neste sentido, o projeto de automação desenvolvido neste Trabalho de Conclusão de Curso (TCC) visou montar uma central que poderia acionar vários equipamentos, utilizando Wi-Fi, aliada a um aplicativo previamente instalado e ajustado em um *smartphone*. No entanto, neste trabalho foi implementado o acionamento de lâmpadas. Dessa forma, é possível verificar os principais elementos que compõem o sistema no esquemático da Figura 1, formado por: um *smartphone*, *modem*, sistema Arduino, placa de potência e lâmpadas.

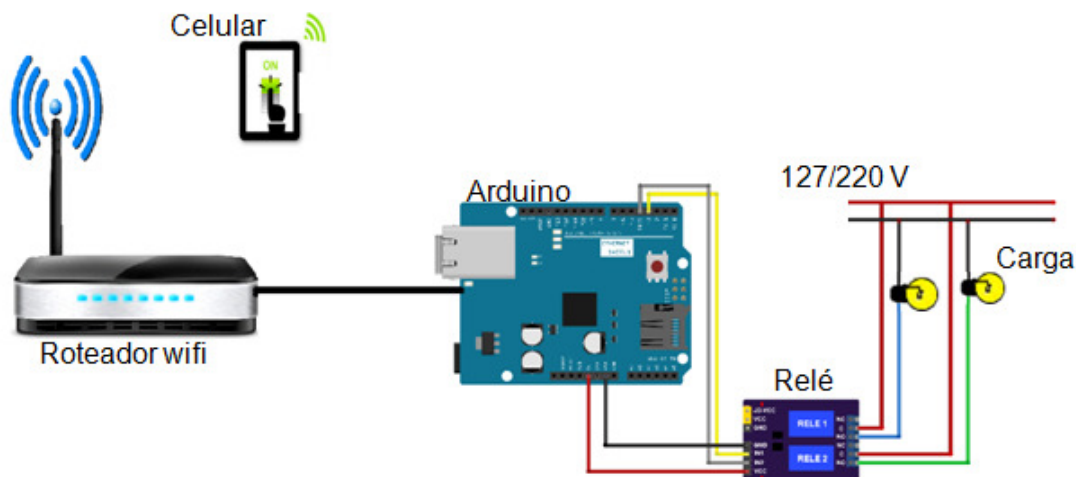


Figura 1 – Sistema proposto.

Fonte: Autoria própria.

## 1.1 TEMA

Em 2007, a facilidade de mandar um e-mail ou postar fotos nas redes sociais em qualquer lugar era algo acessível apenas para pessoas com alto poder aquisitivo. Segundo a ANATEL (Agência Nacional de Telecomunicações) (2013), apenas 1,692 milhões de aparelhos tinham o acesso à internet móvel. Hoje em dia, com a facilidade e o custo para se ter esta tecnologia, pode-se constatar que a internet móvel se popularizou. Segundo dados do IBGE (Instituto Brasileiro de Geografia e Estatística), (2013), cerca de 162,21 milhões de aparelhos já tinham acesso à internet via aparelhos 3G em 2013.

A ideia de uma casa moderna e automatizada foi durante muito tempo associada a algo caro, complicado, pouco acessível e de difícil implantação. Todos estes motivos criam uma barreira entre o público da classe média e a aquisição de tecnologia para automatizar o ambiente ao seu redor (RAMOS & SANTOS, 2015).

### 1.1.1 DELIMITAÇÃO DO TEMA

Neste trabalho está sendo abordado o tema automação residencial utilizando um sistema micro controlado baseado em Arduino, aliado a um aplicativo de *smartphone* com comunicação via rede sem fio Wi-Fi. Para isso, foi desenvolvido um aplicativo voltado a atender as necessidades do usuário, além da montagem e teste de um protótipo para o acionamento de lâmpadas.

### 1.2 PROBLEMAS E PREMISAS

O problema principal encontrado para justificar o desenvolvimento deste trabalho é que, apesar do grande número de opções de automação no mercado a maioria está sendo oferecida a valores altos, não permitindo que essa tecnologia seja de fácil acesso. Tais sistemas, têm o objetivo de facilitar as tarefas, ou mesmo melhorar a harmonia e a interação das pessoas com o ambiente. No caso especial de algumas pessoas portadoras de necessidades especiais (PNEs), o sistema poderia facilitar o acionamento de lâmpadas.

Inicialmente este projeto visa apresentar um mecanismo simples e barato, com o intuito de popularizar a tecnologia da automação residencial apenas com o acionamento de lâmpadas. Porém, existe o interesse dos autores em expandir as aplicações utilizando mais módulos para acionamento de janelas, portas e sensores.

## 1.3 OBJETIVOS

### 1.3.1 OBJETIVO GERAL

A proposta deste projeto é o desenvolvimento e a implantação de um sistema de automação residencial com custo reduzido de instalação, onde o *software open source* e o *hardware* baseado na arquitetura Arduino foram especificados para customizar o produto final.

### 1.3.2 OBJETIVOS ESPECÍFICOS

- Utilizar um módulo micro controlado Arduino e uma placa de rede para conexão a um roteador de Wi-Fi sem o auxílio de PC (*Personal Computer*);
- Configurar uma versão com acesso local de um servidor interno no Arduino, que cria uma porta para acessar e receber comandos de um celular, além de habilitar ou desabilitar portas com relés. Esta versão ficará disponível apenas na rede interna;
- Apresentar um circuito elétrico de comando do Arduino e contadores;
- Instalar o protótipo em uma residência, além de apresentar e avaliar o seu funcionamento prático.

## 1.4 JUSTIFICATIVA

Silva & Ferreira (2003) destacam dois marcos fundamentais da luta por cidadania das pessoas com deficiência: o Ano Internacional da Pessoa com

Deficiência no mundo e a Assembleia Constituinte no Brasil. Para o autor, o Ano Internacional da Pessoa com Deficiência teve grande importância, porque o tema chegou até à imprensa e foram realizados vários programas de televisão sobre o assunto. Hoje, as pessoas com deficiência contam com inúmeras leis que defendem seus direitos e primam pela questão da acessibilidade.

Como descrito anteriormente, a automação é uma área ainda pouco explorada em determinados segmentos da sociedade. Logo, o desenvolvimento de mecanismos de automação residencial que poderá ser acessível às pessoas em geral e PNEs torna o projeto adequado e interessante para uma parcela significativa da população.

Como o presente projeto busca, especificamente, atender as necessidades de automatização de residências para todas as pessoas, foi elaborado um protótipo que ilustra os conceitos abordados neste trabalho de conclusão de curso. Assim, como principais vantagens pode-se citar facilidade no manuseio e interação com o usuário final, sendo capaz de controlar pequenas atividades do cotidiano do homem, como ligar e desligar lâmpadas, com um custo abaixo do mercado (< R\$ 4.000,00) em relação outros projetos oferecidos no mercado em 2016 (ver Tabela 1 – Comparativo de Automação). Desta forma, este projeto é justificado a partir do crescimento e emersão constante da sociedade, combinada com as novas tecnologias e recursos *de software* livre, bem como fontes de conhecimento distribuídas pela rede de internet e o custo de implantação e operação reduzidos em relação ao mercado.

Tabela 1 – Comparativo de Automação

Sondagem no mercado de automação residencial	
Empresas	Faixa de preços
OptiArt	R\$15 mil – R\$ 95 mil
Engetax	R\$10mil – R\$90 mil
Grupo Feliza	R\$ 10ml – 90 mil
iHouse	acima de R\$ 15mil

Fonte: Autoria própria



Demonstrar a facilidade de se controlar, remotamente, estes elementos de acordo com as necessidades do usuário também colaboram para fundamentação deste trabalho. Nesse sentido, o sistema apresenta uma interface simples e amigável, para que qualquer usuário possa ligar e desligar lâmpadas em sua residência.

## 1.5 ESTRUTURA DO TRABALHO

A estrutura deste trabalho apresenta-se da seguinte forma: No Capítulo 1 é apresentada a introdução, objetivos e justificativa do trabalho proposto. No Capítulo 2 é feita uma abordagem sobre a fundamentação teórica das tecnologias envolvidas para o pleno funcionamento do sistema proposto. No Capítulo 3 são tratadas as plataformas e programas envolvidos com os métodos de programação, a descrição técnica do componente Arduino e as conexões referentes à instalação elétrica. No Capítulo 4 é abordada a arquitetura do sistema, suas limitações e vantagens. Por fim, no Capítulo 5 é apresentada a conclusão deste TCC, no Capítulo 6 sugestões de implementações futuras.

## 2 FUNDAMENTAÇÃO TEÓRICA

### 2.1 AUTOMAÇÃO DE RESIDÊNCIAS

A pesquisa sobre a integração de uma residência "inteligente" para pessoas com deficiência se faz necessária devido à necessidade de possibilitar maneiras mais vantajosas a interação com aparelhos elétricos, facilitando nas atividades realizadas na residência, bem como, possibilitando conforto e segurança. Com o tempo e desenvolvimento, pode ser possível monitorar os equipamentos e criar sugestões de uso afim de por exemplo economizar energia.

A história mostra que a primeira possibilidade de se instalar uma casa inteligente foi associada ao conforto do lar, com a energia elétrica movimentando tudo e realizando todos os afazeres. Acreditava-se que, com um simples toque em um botão, todas as tarefas rotineiras seriam realizadas automaticamente (BRUCHER,1939).

O surgimento de residências automatizadas teve início em meados do século XX, por pessoas com interesses em eletricidade, como por exemplo, Jack Fletcher, em 1954 (STIMSON, 1954). Em 1970, foram desenvolvidos, por algumas empresas, equipamentos com funções limitadas de automação residencial, como controle de iluminação e de temperatura. O mesmo princípio está sendo oferecido neste projeto, porém de uma forma simplificada.

A indústria tem mobilizado, a vários anos, tentativas de padronizar as redes domésticas e um protocolo de troca de mensagens entre aparelhos eletrônicos em uma casa. Segundo Abowd e Sterbenz (2000), ainda não existia, no ano de 2000, um acordo se essa normatização seria positiva em curto prazo. Alguns são a favor para desenvolver o mercado, outros creem que este tipo de norma poderá atrapalhar novos projetos.

A história tem mostrado que muitas das tecnologias descobertas não são desenvolvidas de imediato para o mercado, pois não têm uma aplicação viável que permita sua utilização (MELOSI, 1990). Como exemplo, no final de 1870, a

lâmpada inventada por Thomas Edson alavancou todo o setor industrial na parte de iluminação e equipamento elétricos, abrindo espaço para novas iniciativas na área.

O desenvolvimento de tecnologias microcontroladas com a utilização de placas pequenas e facilidade de instalação, são de suma importância para a popularização dos sistemas automatizados em residências. Com isso, este TCC poderá ser um projeto de relevância para facilitar a vida das pessoas com dificuldade de locomoção, dando mais segurança em seus lares e possibilitando um estudo de mercado sobre a comunicação de equipamentos via rede sem fio.

## 2.2 CONCEITOS DE AUTOMAÇÃO

O desenvolvimento de aplicações e serviços em residências não é o mesmo que desenvolver no setor comercial ou industrial, pois em uma indústria ficam bem definidos os padrões esperados quanto a fins lucrativos ou comerciais. Em contrapartida, em uma casa, cada indivíduo tem uma rotina e um padrão que tem que ser monitorado.

No ambiente residencial, os comandos estão disponíveis para serem utilizados no momento que o usuário solicitar. As ações realizadas pelo sistema ao longo do dia vêm se desconectando dos locais físicos impostos para realizar as atividades. Por exemplo, uma pessoa deitada na cama pode acionar uma máquina de lavar roupas na lavanderia ou uma lava louça na cozinha. Cada cômodo pode ser uma central de informações e entretenimento. Kidd, Orr e Abowd (1990) definem a casa como "Ambientes de livre escolha", onde não se tem imposição quanto, onde e nem como os recursos são solicitados e utilizados.

As divisões físicas dos cômodos das casas já não impõem mais quais atividades devem ser realizadas em cada ambiente. Com a tecnologia é possível que várias atividades ocorram dentro de um mesmo cômodo ou ainda que, em toda casa, sejam desenvolvidas estas mesmas atividades, como por exemplo,

ligar o sistema de irrigação do jardim estando sentado no sofá da sala ou no quarto. No caso de chuva, a casa poderia ter um conjunto de sensores para detectar que está chovendo e, através de um sistema inteligente, determinar ações como, avisar ao usuário com dificuldades de escuta ou visão para fechar as janelas, ou o próprio sistema realizar este comando.

### 2.3 CONCEITOS TECNOLÓGICOS

Na década de 80, teve início as utilizações maciças de semicondutores em eletrodomésticos, surgindo diversos equipamentos com sensores e processadores no intuito de facilitar os afazeres da residência. Fletcher (1985) cita um forno da Panasonic dos quais não constava botões de ajustes para temperatura ou tempo para cozinhar. Ao ligá-lo, por meio de sensores, o mesmo sabia quando a comida estava pronta. Segundo Bolzani (2010), o custo para implantação destes componentes nos eletrodomésticos inibiu a fabricação destes eletrodomésticos. Atualmente, mesmo com vários equipamentos vindo com sensores e atuadores, seu uso sempre foi moderado se comparado com a expectativa da época. Além disso, o alto custo da implantação dessas tecnologias ainda é um fator que inviabilizar sua popularização. Os indicadores de desenvolvimento da indústria de semicondutores, relativa a capacidade de processamento computacional, comunicação, armazenamento, integração e dissipação de potência, mostra que é possível instalar dispositivos eletrônicos em qualquer objeto físico, como roupas, móveis, carros e casas. A viabilidade de adicionar estes dispositivos a objetos comuns tem mudado a forma como os indivíduos interagem com esses objetos em seu meio. No caso de um celular ou um *Smartwatches* (relógios inteligentes), os acessórios podem servir como uma base onde o usuário pode comandar equipamentos através de um aplicativo instalado.

Algumas alternativas de controle residencial são encontradas na literatura, como as citadas por Harper (2003), utilizando computadores pessoais como base

de gerenciamento de equipamentos. Mas estes métodos de gerenciamento de equipamentos, utilizando PCs como central de comandos, com teclado, *mouse* e monitor, têm sido substituídos por versões de computação Ubíqua, proposto por Weiser (1991), um dos primeiros desenvolvedores de residência inteligente.

A central de comandos deve ser conectada a um roteador Wi-Fi, que posteriormente passarão a receber sinais via rede que são enviados de qualquer lugar da casa com o auxílio de celular, relógios, televisores ou outro equipamento com tecnologia Wi-Fi e o aplicativo de controle instalado. A computação Ubíqua é responsável por unir o modo de vida com a tecnologia, na forma de pequenos relés distribuídos pela casa, que podem ser acionados por múltiplos usuários. Este método é mais apropriado do que computadores monolíticos, tendo em vista que as necessidades de execução de tarefas são espalhadas pela casa, tanto no tempo quanto no espaço (MATEAS et al., 1996).

## 2.4 CONCEITOS DE COMPUTAÇÃO FÍSICA

A computação física é a parte física dos computadores, seus componentes eletrônicos, por outro lado a computação virtual são os programas de computadores, e esta por sua vez apresenta um grande número de e estudiosos que tem na sua formação um bom entendimento em programação (BANZI, 2011).

A computação física envolve o projeto de objetos interativos que podem se comunicar com humanos utilizando sensores e atuadores controlados por um comportamento implementado como software, executado dentro de um microcontrolador (um pequeno computador em um único *chip*) (BANZI, 2011).

Partindo desse conceito, os criadores do sistema Arduino passaram a utilizar esse protocolo em seus projetos de criação, anteriormente utilizados somente por profissionais. Esta evolução permitiu o acesso de profissionais que atuam em outras áreas como arquitetos, *designers*, entre outros (OLIVEIRA NETO, 2011).

## 2.5 CONCEITOS DE SMARTPHONE

A crescente demanda por dispositivos móveis, tais como *tablets* e *smartphones*, e a divulgação de novas mídias interativas através destes aparelhos, revelam um novo cenário no acesso à informação e ao conhecimento. Esses dispositivos apresentam características próprias, com desafios referentes à acessibilidade que se mostram presentes somente nestas plataformas. Originalmente, telefones celulares surgiram como dispositivos para conversação por voz, exclusivamente. Porém, com o avanço da tecnologia e a evolução das gerações da telefonia celular (ver Tabela 1), esses dispositivos adquiriram também capacidade de processamento e comunicação através da integração da rede celular com rede de dados, em especial a internet (MORIMOTO, 2010).

Tabela 2 – Gerações da telefonia celular.

Geração	1G	2G	2,xG	3G	4G
Características	<ul style="list-style-type: none"> <li>- Transmissão de Dados Analógica (AMPS);</li> <li>- Taxas de 9600bps</li> </ul>	<ul style="list-style-type: none"> <li>- Transmissão digital de dados (TDMA, CDMA e GSM);</li> <li>- Taxas de 9600bps a 14400bps;</li> <li>- Surgimento de aplicações WAP.</li> </ul>	<ul style="list-style-type: none"> <li>- Disponibilização de aplicações pré-3G.</li> </ul>	<ul style="list-style-type: none"> <li>- Evolução CDMA e GSM;</li> <li>- Taxas de até 2Mbps;</li> <li>- Surgimento de aplicações multimídia.</li> </ul>	<ul style="list-style-type: none"> <li>- Elevação das taxas de transmissão de dados;</li> <li>- Tecnologias e aplicações ainda em discussão.</li> </ul>

Fonte: Morimoto, 2010.

Um celular com uma bateria Li-Ion de 860 mAh dispõe de pouco mais de 3 watts de energia (que correspondem ao que um notebook mediano consome em apenas 5 minutos) para realizar todas as suas funções até a próxima recarga. Entretanto, diferente do que se tem nos notebooks, a autonomia dos *smartphones* precisa ser medida em dias e não em horas (MORIMOTO, 2010).

As restrições com relação ao tamanho e ao consumo fez com que o *hardware* dos *smartphones* evoluísse em um caminho bem diferente dos PCs,

com o uso de processadores de baixo consumo e *chips* altamente integrados. A mudança mais notável é o uso de processadores ARM. Os *chips* ARM são processadores que apresentam uma arquitetura extremamente otimizada, com poucos transistores e um consumo elétrico extremamente baixo. Tal processador é baseado em um processador ARM11. Entre o conjunto de componentes auxiliares estão um acelerador de vídeo, que se encarrega da decodificação de vídeos em diversos formatos (desafogando o processador principal e ajudando a reduzir o consumo), um acelerador 3D otimizado para jogos e aplicativos escritos em Java, um processador ARM9 auxiliar, para o processamento dos sinais da rede 3G e um *chip* Qualcomm gpsOne, além de um receptor GPS de 20 canais (MORIMOTO, 2010).

Neste projeto, utilizou-se o *smartphone* GALAXY S4 da Samsung com processador Qualcomm APQ8064T Snapdragon 600, 1,9 GHz Quad Core, 2GB de memória RAM, com conexão Wi-Fi 802.11 a/b/g/n/ac e velocidade de 100 Mbps. O equipamento atendeu bem quanto as condições de programação, processamento e comunicação, fazendo com que a operação de acionamento da lâmpada fosse finalizada em um tempo normal dentro de ambiente fechado com sinal Wi-Fi pleno. A escolha do aparelho deveu-se apenas ao fato da disponibilidade do mesmo, uma vez que é de propriedade de um dos membros da equipe deste TCC. Entretanto, o projeto é aplicável aos principais *smartphones* comercialmente disponíveis.

## 2.6 CONCEITOS DE ROTEADOR WI-FI

O roteador é um aparelho usado em redes de computadores para o encaminhamento das informações acondicionadas em pacotes de dados, proporcionando conectividade entre os dispositivos como computadores, *smartphones* e *tablets*, em redes LAN com a internet. Além disso, o roteador possui uma característica específica: buscar as melhores rotas para enviar e receber dados, podendo priorizar não só as transmissões mais curtas, como também as menos congestionadas (RIBEIRO, 2013).

Para conectar aparelhos que utilizam transmissão de dados via *wireless* (sem fio) e eliminar a necessidade de um computador estar conectado a um computador principal, a maior parte dos modernos roteadores possui potentes antenas para enviar e receber suas transmissões de pacotes de dados por redes sem fio e se conectam diretamente ao modem da internet de banda larga.

Porém, diferente de um *access point*, o roteador não é só um equipamento para a distribuição de conexões locais, realizando também o processamento das melhores rotas para transmissão de pacotes de dados, garantindo assim que seus sinais cheguem ao seu destino de maneira mais rápida possível, seguindo as melhores rotas para difundir os dados a ele encaminhado. Os roteadores mais modernos funcionam conectados ao cabo de banda larga e possuem mais quatro saídas de cabos para computadores e uma antena, que é a parte fundamental para fazer funcionar a transmissão do sinal para a conectividade sem fio, alcançando assim dispositivos com conexão via *wireless*, como smartphones, tablets e notebooks. Informação que é importante conhecer é o padrão de conectividade da rede sem fio, que indica a frequência da transmissão de dados entre seus dispositivos e o roteador e é caracterizado pelos parâmetros 802.11b, 802.11g e 802.11n, sendo o primeiro mais antigo, enquanto os últimos são as frequências mais modernas (Ribeiro, 2013).

Neste projeto, utilizou-se o Roteador Wireless N 150Mbps da TL-WR741ND da empresa TP-Link. No ambiente de instalação, uma das entradas fica a disposição para conexão do cabo que vai para o módulo Arduino.

### 2.6.1 REDES WI-FI

A tecnologia *wireless* permite a conexão entre diferentes pontos sem a necessidade do uso de cabos (nem de telefonia, nem de TV a cabo, nem de fibra óptica), através da instalação de uma antena e de um rádio de transmissão.

Com base no trabalho de Tanenbaum (2003), são descritos, resumidamente, os principais conceitos relacionados a tecnologia Wi-Fi:



- IEEE 802.11: Fornece o serviço de segurança dos dados através de dois métodos: autenticação e criptografia. É o padrão específico do IEEE (*Institute of Electrical and Electronics Engineers*) que cuida da segurança das redes sem fio que também são conhecidas como redes Wi-Fi ou Wireless.
- Wi-Fi: O padrão Wi-Fi opera em faixas de frequências que não necessitam de licença para instalação e/ou operação. Este fato as torna atrativas. No entanto, para uso comercial no Brasil é necessária licença da ANATEL. Para se ter acesso à internet através de rede Wi-Fi deve-se estar no raio de ação de um ponto de acesso (normalmente conhecido por *hotspot*) ou local público onde opere rede sem fios e usar dispositivo móvel, como laptop.
- Funcionamento: Através da utilização de portadoras de rádio ou infravermelho, as redes *wireless* estabelecem a comunicação de dados entre os pontos da rede. Os dados são modulados na portadora de rádio e transmitidos através de ondas eletromagnéticas com alcance de até 300 m, em locais abertos, e 50 m em locais com paredes. Múltiplas portadoras de rádio podem coexistir num mesmo meio, sem que uma interfira na outra. Para extrair os dados, o receptor sintoniza numa frequência específica e rejeita as outras portadoras de frequências diferentes.
- Classificação das redes sem fios:
  - WPAN: *Wireless Personal Area Network* ou rede pessoal sem fio. Normalmente utilizada para interligar dispositivos eletrônicos fisicamente próximos.
  - WLAN: *Wireless Local Area Network*. WLAN já é muito importante como opção de conexão em muitas áreas de negócio.
  - WMAN (Wi-Max): Trata-se de uma tecnologia de banda larga sem-fio, capaz de atuar como alternativa a tecnologias como cabo e DSL na construção de redes comunitárias e provimento de acesso de última milha.

- WWAN: *Wireless Wide Area Network*. É a rede geograficamente infinita via comunicação sem fio. Hoje encontram-se diversas empresas que comercializam a internet sem fio.
- WMA: Tipo de rede WLAN mais comumente encontrada em roteadores simples comercializados.

### 2.6.2 SEGURANÇA

Na criptografia faz-se o uso de uma chave (KEY) para codificar os dados que são transferidos. A chave é determinada pelo tamanho, por exemplo 64, 128 e 256 *bits*. Quanto maior a combinação de *bits*, mais forte fica a criptografia, e, conseqüentemente, a segurança.

A seguir são apresentados os protocolos de segurança mais utilizados:

- WEP: Significa *Wired Equivalent Privacy*. O WEP se encarrega de “encriptar” os dados transmitidos através da rede. Existem dois padrões WEP: 64 e 128 *bits*. O primeiro é suportado por qualquer ponto de acesso ou interface que siga o padrão Wi-Fi, o que engloba todos os produtos comercializados atualmente. O padrão de 128 *bits*, por sua vez, não é suportado por todos os produtos. Para habilitá-lo é preciso que todos os componentes usados na rede suportem o padrão. Caso contrário, os nós que suportarem apenas o padrão de 64 *bits* ficarão fora da rede. É muito inseguro devido a sua arquitetura.
- WPA: Significa *Wi-Fi Protected Access*. Também chamado de WEP2, ou TKIP (*Temporal Key Integrity Protocol*), surgiu de um esforço conjunto de membros da Wi-Fi Aliança e de membros do IEEE, empenhados em aumentar o nível de segurança das redes sem fio, ainda no ano de 2003, combatendo algumas das vulnerabilidades do WEP. Com a substituição do WEP pelo WPA, tem-se como vantagem melhoria na criptografia dos dados

ao utilizar um protocolo de chave temporária (TKIP) que possibilita a criação de chaves por pacotes, além de possuir função detectora de erros, um vetor de inicialização de 48 *bits*, ao invés de 24 como no WEP, e um mecanismo de distribuição de chaves. Além disso, outra vantagem é a melhoria no processo de verificação de usuários. Através do 802.11x e do EAP (*Extensible Authentication Protocol*) é possível que o servidor de autenticação realize a avaliação de cada usuário antes de acessar a rede.

- WPA2: É o Avanço do WPA. O WPA2 é a modalidade de segurança sem fio mais forte.

Para este projeto, o WAP2 ou WAP fazem parte dos padrões adotados para conectar o sistema de automação. Com isso, tem-se um dos melhores padrões de segurança oferecido por redes Wi-Fi simples.

## 2.7 CONCEITO DE SISTEMAS OPERACIONAIS

### 2.7.1 ANDROID

O Sistema Operacional Androide é um sistema desenvolvido pela Google e outras empresas que formam o OpenHandset Alliance. O desenvolvimento da plataforma é um projeto aberto onde vários fabricantes de dispositivos móveis podem modificar o sistema operacional. O Androide pode ser utilizado por diversos dispositivos como, celulares, *tablets* relógios, televisores, geladeiras e centrais multimídia. À princípio ele foi criado para tirar proveito de telas sensíveis ao toque, com grande capacidade de processamento, boa conectividade e interação com dispositivos atuais. Com isso, é possível criar aplicativos tão complexos quanto os destinados a computadores pessoais (DE MELO EUZÉBIO, 2011).

O programa utilizado na automação proposta neste TCC é o TouchOSC, sendo compatível com Androide 1.6 ou superior. O aplicativo é apresentado com maiores detalhes na seção 3.4.

### 2.7.2 IOS

O iOS é o sistema operacional para dispositivos móveis da empresa Apple. Foi desenvolvido originalmente para o iPhone, no entanto, hoje, também é usado em equipamentos iPod Touch, iPad e Apple TV. A Apple não permite o sistema operacional rodar em *hardware* de terceiros. A interface do usuário do iOS é baseada no conceito de manipulação direta, utilizando gestos em multitoque. A interação com o sistema operacional inclui gestos como apenas tocar na tela, deslizar o dedo, e o movimento de "pinça" utilizado para se ampliar ou reduzir a imagem. Em relação ao *hardware*, conta com um acelerômetro que aplicado com a programação iOS pode realizar diferentes funções. São usados por alguns aplicativos para responder à agitação do aparelho (um resultado comum é o comando de desfazer) ou rodá-la em três dimensões (um resultado comum é a mudança do modo retrato para modo paisagem).

Elementos de controle de Interface consistem de barras, chaves e botões. A resposta à entrada do usuário é imediata e oferece uma interface bem simples. A interação com o sistema operacional inclui gestos, tais como tocar e apertar, modificar o tipo de ação para a manipulação do equipamento que tem definições específicas dentro do contexto do sistema operacional iOS e sua interface multitoque (APPLE, 2008).

Neste projeto, não foram feitos testes com o sistema operacional iOS. Entretanto, tecnicamente, o programa é compatível com iOS 4.3 ou superior, sendo viável sua utilização devido às respectivas características tecnológicas.

### 3 PLATAFORMAS, PROGRAMAÇÃO E DESENVOLVIMENTO

#### 3.1 PLATAFORMA ARDUINO

O Arduino foi desenvolvido inicialmente na Itália, projetado como um instrumento para auxiliar alunos do ensino. No entanto, em 2005 passou a ser comercializado por Massimo Banzi e David Cuartielles, vindo a ser um produto bem-sucedido entre fabricantes e estudantes, por apresentar uma fácil utilização e boa durabilidade (MONK, 2013).

A meta dos professores foi a de desenvolver um *hardware* com facilidade de programação e a um custo reduzido. O projeto do Arduino foi concebido como *open source*, onde todo seu projeto, código fonte e *hardware* são de domínio público (MONK, 2013).

O sistema Arduino apresenta dois componentes básicos: a placa Arduino, que é um elemento físico (*hardware*) utilizado para construir seus objetos, e a IDE (*Integrated Development Environment*), que é um programa de computador onde é gerado o código que será executado na placa Arduino (BANZI, 2015).

Na Figura 2 é apresentado um diagrama de blocos genérico que apresenta os principais passos na programação do Arduino.

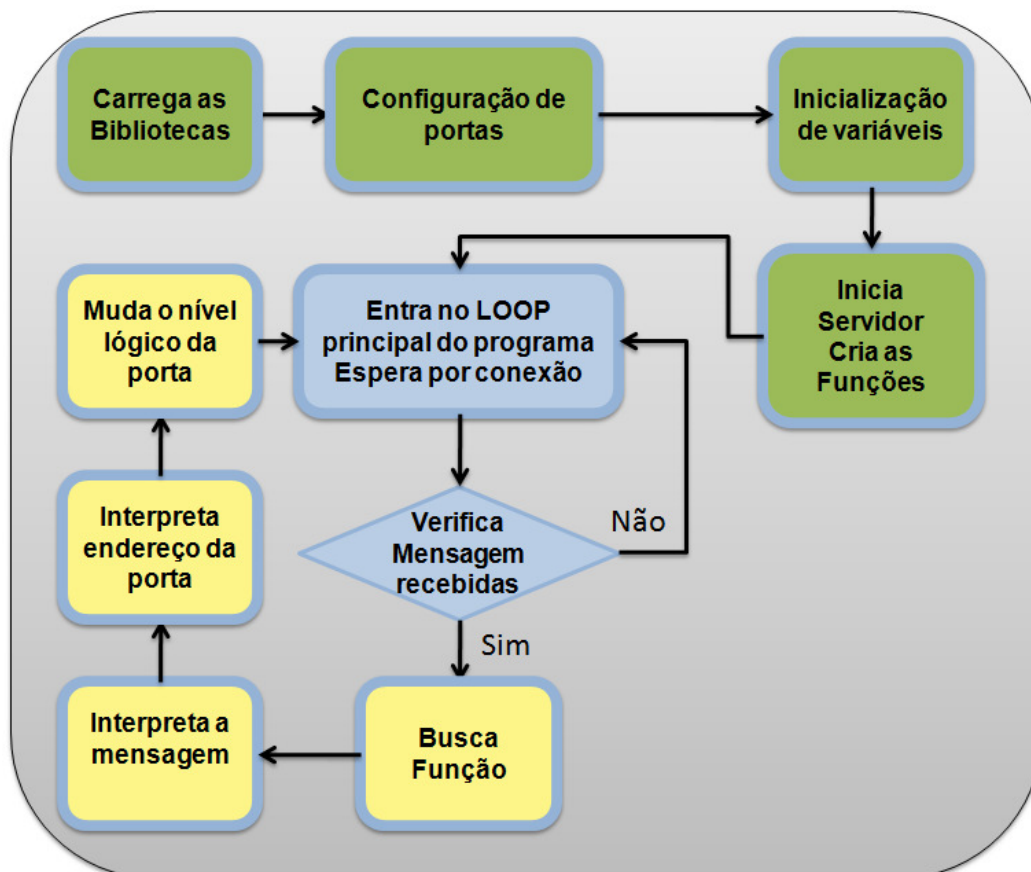


Figura 2 – Diagrama de blocos da programação do Arduino.

Fonte: Autoria própria.

A base do Arduino é o microcontrolador AVR® da Atmel e existem diversas variações e derivações destes *chips* com mais capacidade de armazenamento, como o ATmega8 e o ATmega256 de 256k de memória de programa Flash. As principais diferenças dos Arduinos comparando com o microcontrolador que os integram, são suas derivações, que é mostrada na Tabela 3, e as variedades de microcontroladores que podem acompanhar o Arduino (MONK, 2013).

Através de uma análise da Tabela 3 é possível evidenciar que o Arduino com o microcontrolador Mega2560 apresenta uma capacidade superior de memória Flash, além de grandes números de portas digitais e saídas PWM (*Pulse Width Modulation*). No entanto este destaque reflete no aumento de seu custo. Frente as necessidades deste projeto optaram-se pelo Atmega328, mostrado na Tabela 2, coluna 3, que apresenta baixo custo e atende as necessidades, como ter 14 pinos que poderiam conectar até 14 lâmpadas.

Tabela 3 – Comparativos entre diferentes microcontroladores Arduino.

ARDUINO	Diecimila	Duemilanove168	Duemilanove328	Mega1280	Mega2560
Processador	ATmega8	ATmega168	ATmega328	ATmega1280	ATmega2560
Memória Flash	8K	16K	32K	128K	256K
Memória RAM	1K	1K	2K	8K	8K
Memória EEPROM	512 bytes	512 bytes	1K	4K	4K
Pinos digitais	14	14	14	54	54
Pinos Analógicos	6	6	6	16	16
Saídas PWM	3	6	6	14	14

Fonte: Andrade, 2013.

Na Figura 3 é apresentado o *layout* de uma placa Arduino Uno R3 (valor de R\$ 70,00 em dezembro de 2016) empregada neste trabalho e algumas descrições importantes.

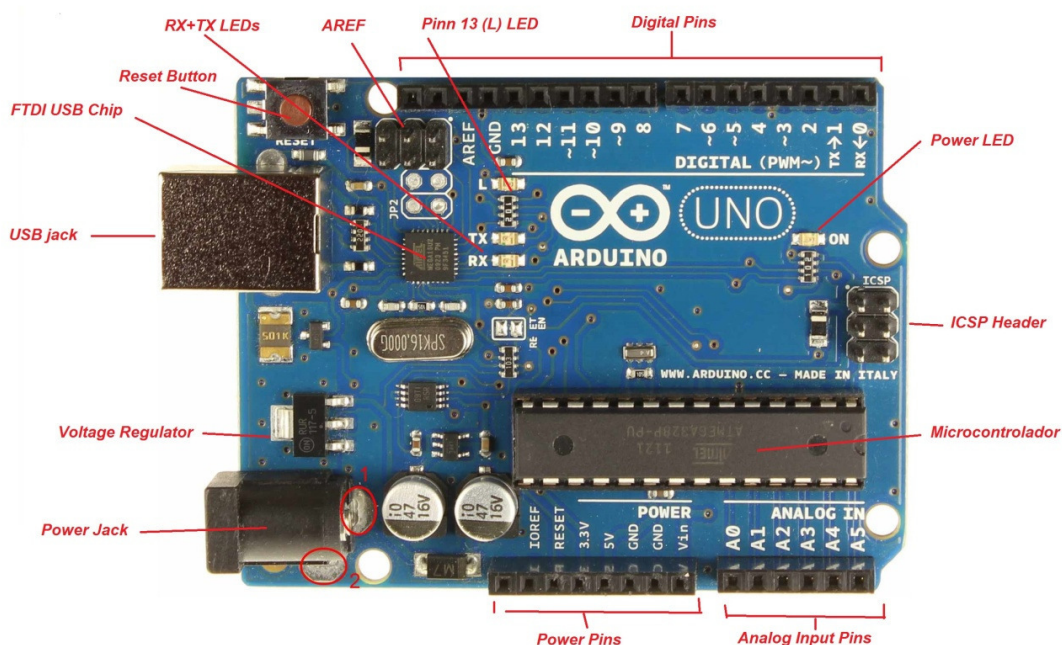


Figura 3 – Placa Arduino Uno R3.

Fonte: Arduino, 2009.

A seguir são descritos os principais itens que compõem o módulo Arduino:

- USB Jack: Permite o envio do código fonte para o microcontrolador Atmega;
- FTDI USB Chip: Circuito integrado que faz uma conversão de protocolos de comunicação;
- Pin 13(L) Led: O pino 13 é acionado quando o programa inserido no microcontrolador é executado e ainda verifica se está conectado à porta USB do computador, que quando o Arduino está instalado corretamente emite um sinal luminoso (piscando);
- Digital Pins: São portas digitais de entradas e saídas;
- Power Led: indica que o Arduino está energizado;
- Reset Button: Este botão reinicia a aplicação inserida no microcontrolador;
- Microcontrolador: É um computador com um conjunto de instruções aritméticas reduzida;
- Analog Input Pins: Pinos de entrada de sinais analógicos;
- Power Jack: Através desse conector é feita a alimentação externa do Arduino que pode variar entre os níveis de 6 a 20 V, ou ainda existe a possibilidade de ser alimentado pelo cabo USB;
- Voltage Regulator: responsável por filtrar a tensão de entrada proveniente do Power Jack, possibilitando a alimentação do Arduino nas tensões entre 6 a 20 V;
- AREF: É uma Referência de nível de tensão das entradas analógicas, utilizadas para a programação, que podem ser modificadas através do código. Caso haja a necessidade de trabalhar com tensões diferentes de 3,3 V e 5 V, utilizam-se esses pinos, sem ultrapassar os limites de tensão entre 0 a 5 V.

O custo da placa Arduino Uno R3, no mês de novembro de 2016, variou entre R\$ 35,00 e R\$ 70,00, incluindo o cabo USB para gravação e depuração.



### 3.1 DEFINIÇÃO DE SHIELDS

Os *shields* são módulos que podem ser facilmente conectados, sobrepostos a uma placa Arduino, tendo como finalidade estender as capacidades internas da placa principal. Há no mercado uma variedade de *shields* que permitem este acoplamento ao Arduino. Algumas são utilizadas para conexão com a internet como a tecnologia Zigbee, Xbee e placa de rede com um conector RJ-45, utilizando módulos com recurso de GPS, GSM, controladores de ponte-H, produzidos em escala industrial. Há ainda aqueles criados por hobistas do Arduino que ao adquirirem a liberação de sua fonte e arquitetura podem ser reproduzidos através de uma licença *Open Source*, como os Supervisores de Arduino, acelerômetros e Wi-Fi com cartão SD (*Secure Digital*). A grande vantagem dessas placas está em permitir que se façam os próprios *shields*, com um custo reduzido para os usuários (OLIVEIRA NETO, 2011).

Na Figura 4 são mostradas duas fotos da placa *shield* Arduino Ethernet W5100, utilizada neste TCC.

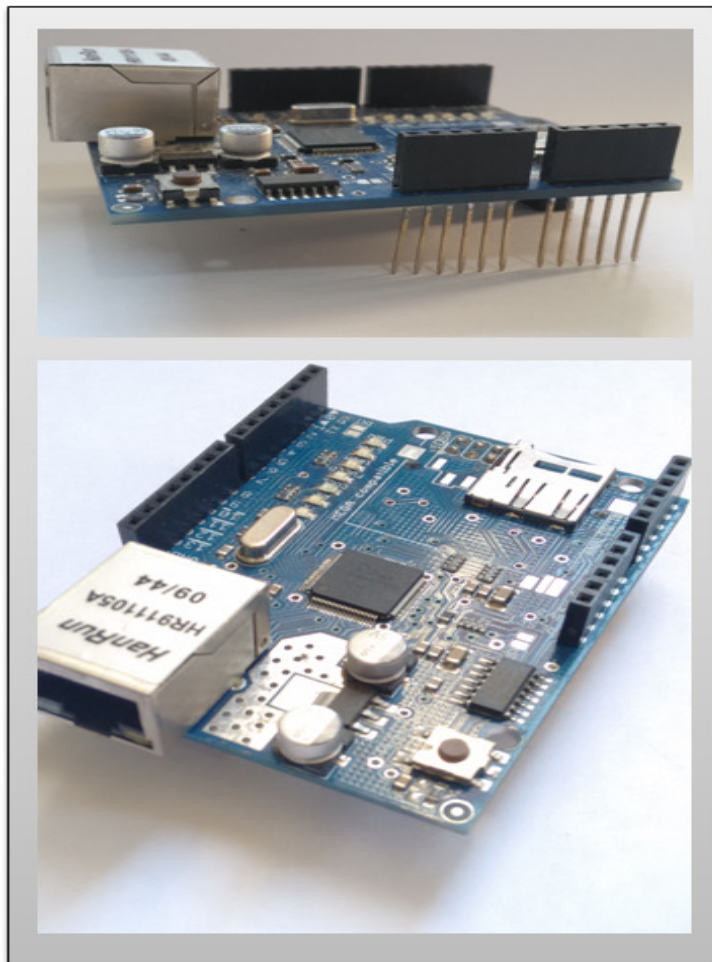


Figura 4 – *Shield Arduino Ethernet.*

Fonte: Autoria própria.

O custo do Arduino Ethernet *shield* W5100, no mês de novembro de 2016, variou entre R\$ 50,00 e R\$ 60,00, nos principais *sites* de compra nacionais.

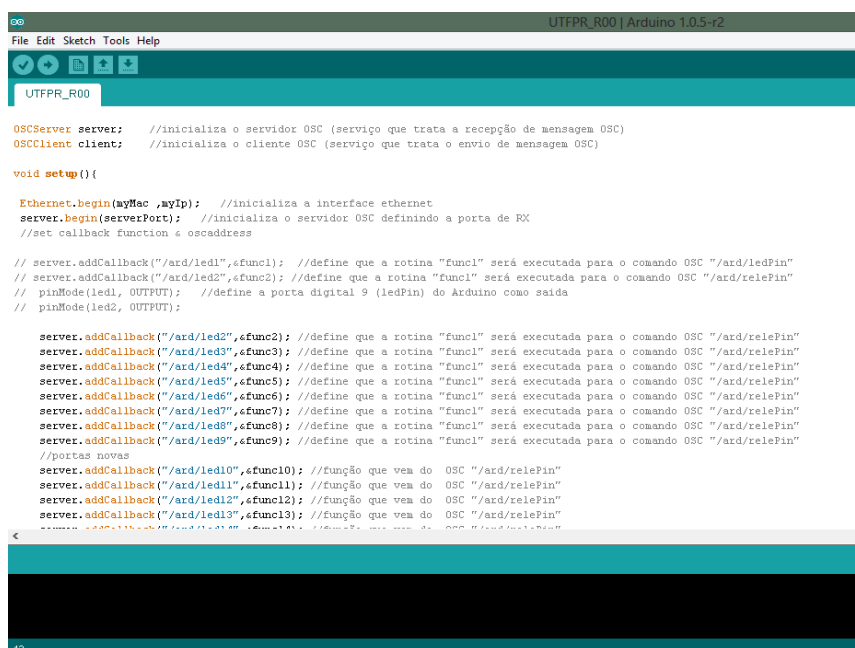
### 3.2 SOFTWARE ARDUINO DEVELOPMENT ENVIRONMENT

No desenvolvimento do código de programação, *upload* e *debug* para o *hardware* do Arduino foi preciso utilizar a plataforma de desenvolvimento de *software* livre Arduino *Development Enviroment*. Nesta plataforma é necessário usar a linguagem de programação Wiring que é derivada das linguagens C e C++. O aplicativo dispõe de bibliotecas específicas e por se tratar de *software* livre, é

possível implementar e criar rotinas, segundo as necessidades encontradas na sua utilização.

O Arduino IDE apresenta a possibilidade do reconhecimento de todas as estruturas da linguagem C e ainda os recursos da linguagem C++. Com o código pronto e compilado é realizado o “*upload*” para o *hardware* (placa Arduino). Como procedimento, existe a verificação do código para que este obedeça às regras da linguagem e, caso nenhum erro seja verificado, é criado o código binário correspondente ao que foi construído. Finalmente, o código é gravado no microcontrolador, para a sua utilização no processo (ANDRADE, 2013).

A Figura 5 representa a tela do ambiente de desenvolvimento do código. A plataforma é bastante amigável e conta com uma verificação antes de gravar o programa.



```

UTFPR_R00 | Arduino 1.0.5-r2
File Edit Sketch Tools Help
UTFPR_R00
OSCServer server; //inicializa o servidor OSC (serviço que trata a recepção de mensagens OSC)
OSCClient client; //inicializa o cliente OSC (serviço que trata o envio de mensagens OSC)

void setup() {

Ethernet.begin(myMac, myIp); //inicializa a interface ethernet
server.begin(serverPort); //inicializa o servidor OSC definindo a porta de RX
//set callback function e oscaddress

// server.addCallback("/ard/led1",efunc1); //define que a rotina "func1" será executada para o comando OSC "/ard/ledPin"
// server.addCallback("/ard/led2",efunc2); //define que a rotina "func1" será executada para o comando OSC "/ard/relePin"
// pinMode(led1, OUTPUT); //define a porta digital 9 (ledPin) do Arduino como saída
// pinMode(led2, OUTPUT);

server.addCallback("/ard/led2",efunc2); //define que a rotina "func1" será executada para o comando OSC "/ard/relePin"
server.addCallback("/ard/led3",efunc3); //define que a rotina "func1" será executada para o comando OSC "/ard/relePin"
server.addCallback("/ard/led4",efunc4); //define que a rotina "func1" será executada para o comando OSC "/ard/relePin"
server.addCallback("/ard/led5",efunc5); //define que a rotina "func1" será executada para o comando OSC "/ard/relePin"
server.addCallback("/ard/led6",efunc6); //define que a rotina "func1" será executada para o comando OSC "/ard/relePin"
server.addCallback("/ard/led7",efunc7); //define que a rotina "func1" será executada para o comando OSC "/ard/relePin"
server.addCallback("/ard/led8",efunc8); //define que a rotina "func1" será executada para o comando OSC "/ard/relePin"
server.addCallback("/ard/led9",efunc9); //define que a rotina "func1" será executada para o comando OSC "/ard/relePin"
//portas novas
server.addCallback("/ard/led10",efunc10); //função que vem do OSC "/ard/relePin"
server.addCallback("/ard/led11",efunc11); //função que vem do OSC "/ard/relePin"
server.addCallback("/ard/led12",efunc12); //função que vem do OSC "/ard/relePin"
server.addCallback("/ard/led13",efunc13); //função que vem do OSC "/ard/relePin"
server.addCallback("/ard/led14",efunc14); //função que vem do OSC "/ard/relePin"
server.addCallback("/ard/led15",efunc15); //função que vem do OSC "/ard/relePin"
server.addCallback("/ard/led16",efunc16); //função que vem do OSC "/ard/relePin"
server.addCallback("/ard/led17",efunc17); //função que vem do OSC "/ard/relePin"
server.addCallback("/ard/led18",efunc18); //função que vem do OSC "/ard/relePin"
server.addCallback("/ard/led19",efunc19); //função que vem do OSC "/ard/relePin"
server.addCallback("/ard/led20",efunc20); //função que vem do OSC "/ard/relePin"
server.addCallback("/ard/led21",efunc21); //função que vem do OSC "/ard/relePin"
server.addCallback("/ard/led22",efunc22); //função que vem do OSC "/ard/relePin"
server.addCallback("/ard/led23",efunc23); //função que vem do OSC "/ard/relePin"
server.addCallback("/ard/led24",efunc24); //função que vem do OSC "/ard/relePin"
server.addCallback("/ard/led25",efunc25); //função que vem do OSC "/ard/relePin"
server.addCallback("/ard/led26",efunc26); //função que vem do OSC "/ard/relePin"
server.addCallback("/ard/led27",efunc27); //função que vem do OSC "/ard/relePin"
server.addCallback("/ard/led28",efunc28); //função que vem do OSC "/ard/relePin"
server.addCallback("/ard/led29",efunc29); //função que vem do OSC "/ard/relePin"
server.addCallback("/ard/led30",efunc30); //função que vem do OSC "/ard/relePin"
server.addCallback("/ard/led31",efunc31); //função que vem do OSC "/ard/relePin"
server.addCallback("/ard/led32",efunc32); //função que vem do OSC "/ard/relePin"
server.addCallback("/ard/led33",efunc33); //função que vem do OSC "/ard/relePin"
server.addCallback("/ard/led34",efunc34); //função que vem do OSC "/ard/relePin"
server.addCallback("/ard/led35",efunc35); //função que vem do OSC "/ard/relePin"
server.addCallback("/ard/led36",efunc36); //função que vem do OSC "/ard/relePin"
server.addCallback("/ard/led37",efunc37); //função que vem do OSC "/ard/relePin"
server.addCallback("/ard/led38",efunc38); //função que vem do OSC "/ard/relePin"
server.addCallback("/ard/led39",efunc39); //função que vem do OSC "/ard/relePin"
server.addCallback("/ard/led40",efunc40); //função que vem do OSC "/ard/relePin"
server.addCallback("/ard/led41",efunc41); //função que vem do OSC "/ard/relePin"
server.addCallback("/ard/led42",efunc42); //função que vem do OSC "/ard/relePin"
server.addCallback("/ard/led43",efunc43); //função que vem do OSC "/ard/relePin"
server.addCallback("/ard/led44",efunc44); //função que vem do OSC "/ard/relePin"
server.addCallback("/ard/led45",efunc45); //função que vem do OSC "/ard/relePin"
server.addCallback("/ard/led46",efunc46); //função que vem do OSC "/ard/relePin"
server.addCallback("/ard/led47",efunc47); //função que vem do OSC "/ard/relePin"
server.addCallback("/ard/led48",efunc48); //função que vem do OSC "/ard/relePin"
server.addCallback("/ard/led49",efunc49); //função que vem do OSC "/ard/relePin"
server.addCallback("/ard/led50",efunc50); //função que vem do OSC "/ard/relePin"
server.addCallback("/ard/led51",efunc51); //função que vem do OSC "/ard/relePin"
server.addCallback("/ard/led52",efunc52); //função que vem do OSC "/ard/relePin"
server.addCallback("/ard/led53",efunc53); //função que vem do OSC "/ard/relePin"
server.addCallback("/ard/led54",efunc54); //função que vem do OSC "/ard/relePin"
server.addCallback("/ard/led55",efunc55); //função que vem do OSC "/ard/relePin"
server.addCallback("/ard/led56",efunc56); //função que vem do OSC "/ard/relePin"
server.addCallback("/ard/led57",efunc57); //função que vem do OSC "/ard/relePin"
server.addCallback("/ard/led58",efunc58); //função que vem do OSC "/ard/relePin"
server.addCallback("/ard/led59",efunc59); //função que vem do OSC "/ard/relePin"
server.addCallback("/ard/led60",efunc60); //função que vem do OSC "/ard/relePin"
server.addCallback("/ard/led61",efunc61); //função que vem do OSC "/ard/relePin"
server.addCallback("/ard/led62",efunc62); //função que vem do OSC "/ard/relePin"
server.addCallback("/ard/led63",efunc63); //função que vem do OSC "/ard/relePin"
server.addCallback("/ard/led64",efunc64); //função que vem do OSC "/ard/relePin"
server.addCallback("/ard/led65",efunc65); //função que vem do OSC "/ard/relePin"
server.addCallback("/ard/led66",efunc66); //função que vem do OSC "/ard/relePin"
server.addCallback("/ard/led67",efunc67); //função que vem do OSC "/ard/relePin"
server.addCallback("/ard/led68",efunc68); //função que vem do OSC "/ard/relePin"
server.addCallback("/ard/led69",efunc69); //função que vem do OSC "/ard/relePin"
server.addCallback("/ard/led70",efunc70); //função que vem do OSC "/ard/relePin"
server.addCallback("/ard/led71",efunc71); //função que vem do OSC "/ard/relePin"
server.addCallback("/ard/led72",efunc72); //função que vem do OSC "/ard/relePin"
server.addCallback("/ard/led73",efunc73); //função que vem do OSC "/ard/relePin"
server.addCallback("/ard/led74",efunc74); //função que vem do OSC "/ard/relePin"
server.addCallback("/ard/led75",efunc75); //função que vem do OSC "/ard/relePin"
server.addCallback("/ard/led76",efunc76); //função que vem do OSC "/ard/relePin"
server.addCallback("/ard/led77",efunc77); //função que vem do OSC "/ard/relePin"
server.addCallback("/ard/led78",efunc78); //função que vem do OSC "/ard/relePin"
server.addCallback("/ard/led79",efunc79); //função que vem do OSC "/ard/relePin"
server.addCallback("/ard/led80",efunc80); //função que vem do OSC "/ard/relePin"
server.addCallback("/ard/led81",efunc81); //função que vem do OSC "/ard/relePin"
server.addCallback("/ard/led82",efunc82); //função que vem do OSC "/ard/relePin"
server.addCallback("/ard/led83",efunc83); //função que vem do OSC "/ard/relePin"
server.addCallback("/ard/led84",efunc84); //função que vem do OSC "/ard/relePin"
server.addCallback("/ard/led85",efunc85); //função que vem do OSC "/ard/relePin"
server.addCallback("/ard/led86",efunc86); //função que vem do OSC "/ard/relePin"
server.addCallback("/ard/led87",efunc87); //função que vem do OSC "/ard/relePin"
server.addCallback("/ard/led88",efunc88); //função que vem do OSC "/ard/relePin"
server.addCallback("/ard/led89",efunc89); //função que vem do OSC "/ard/relePin"
server.addCallback("/ard/led90",efunc90); //função que vem do OSC "/ard/relePin"
server.addCallback("/ard/led91",efunc91); //função que vem do OSC "/ard/relePin"
server.addCallback("/ard/led92",efunc92); //função que vem do OSC "/ard/relePin"
server.addCallback("/ard/led93",efunc93); //função que vem do OSC "/ard/relePin"
server.addCallback("/ard/led94",efunc94); //função que vem do OSC "/ard/relePin"
server.addCallback("/ard/led95",efunc95); //função que vem do OSC "/ard/relePin"
server.addCallback("/ard/led96",efunc96); //função que vem do OSC "/ard/relePin"
server.addCallback("/ard/led97",efunc97); //função que vem do OSC "/ard/relePin"
server.addCallback("/ard/led98",efunc98); //função que vem do OSC "/ard/relePin"
server.addCallback("/ard/led99",efunc99); //função que vem do OSC "/ard/relePin"
server.addCallback("/ard/led100",efunc100); //função que vem do OSC "/ard/relePin"
}
18

```

Figura 5 – Plataforma de programação Arduino em Linguagem “C”.

Fonte: Autoria própria.

### 3.4 SOFTWARE TOUCHOSC INTERFACE EDITOR

Esta versão do *software* é específica para desenvolver a interface com usuário e é instalada no PC de desenvolvimento. O *software* TouchOSC Editor é o programa utilizado para a personalização da interface que foi utilizada no celular

para o usuário final. O *download* pode ser efetuado através do *site* do desenvolvedor TouchOSC (ANDRADE, 2013).

O aplicativo permite o controle remoto e o recebimento de informações de outros *softwares* ou *hardwares* que implementam protocolos OSC ou MIDI, como: Apple Logic Pro/Express, Data, Max/MSP/Jitter, OSCulator, Resolume Avenue 3, Modul8, Reaktor, Derivative TouchDesigner, Isadora e muitos outros. Sua interface gráfica pode ser visualizada na Figura 6.



Figura 6 – Interface TouchOSC.

Fonte: <http://hexler.net/>, 2015

A Figura 7 apresenta o ambiente de desenvolvimento do programa. Com esta plataforma de desenvolvimento o *layout* do projeto é desenvolvido em blocos que podem receber nomes de cômodos da casa, facilitando o desenvolvimento o projeto.

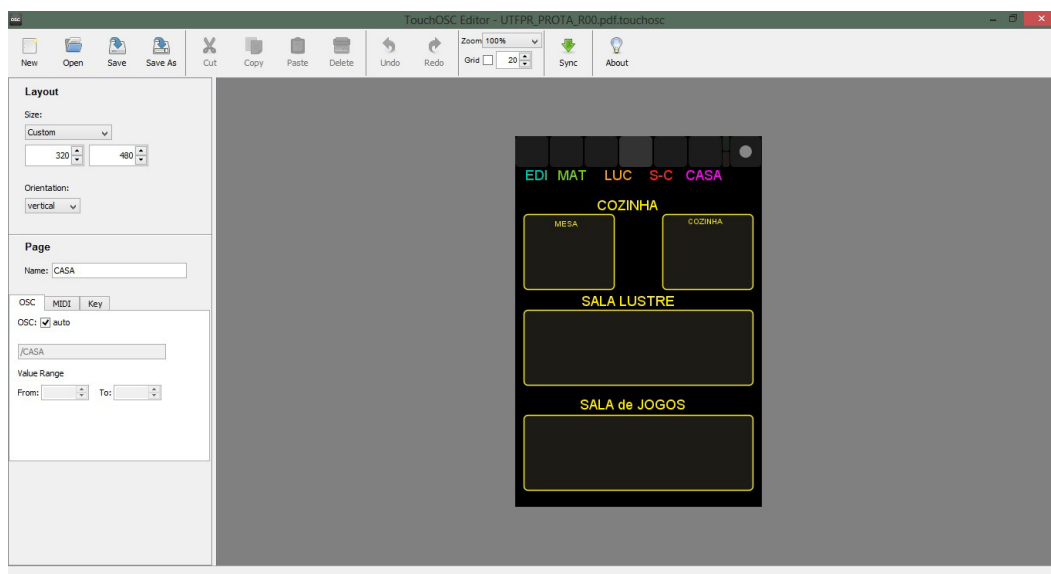


Figura 7 – Ambiente de personalização TouchOSC.

Fonte: Autoria própria.

A Figura 8 mostra a foto do aplicativo já instalado em um celular Android pronto para ser usado.



Figura 8 – Aplicativo em um celular Android.

Fonte: Autoria própria.

### 3.5 INSTALAÇÃO

Para montar o sistema micro controlado deste TCC, deve-se seguir alguns passos. Primeiramente, junta-se as placas Arduino Uno e o *shield* Arduino Ethernet (Figura 9). Conforme pode visualizado na Figura 10, uma placa se encaixa perfeitamente sobre a outra. Em seguida, conecta-se o cabo USB em uma das placas e no computador para realizar a programação e a alimentação temporária das placas.

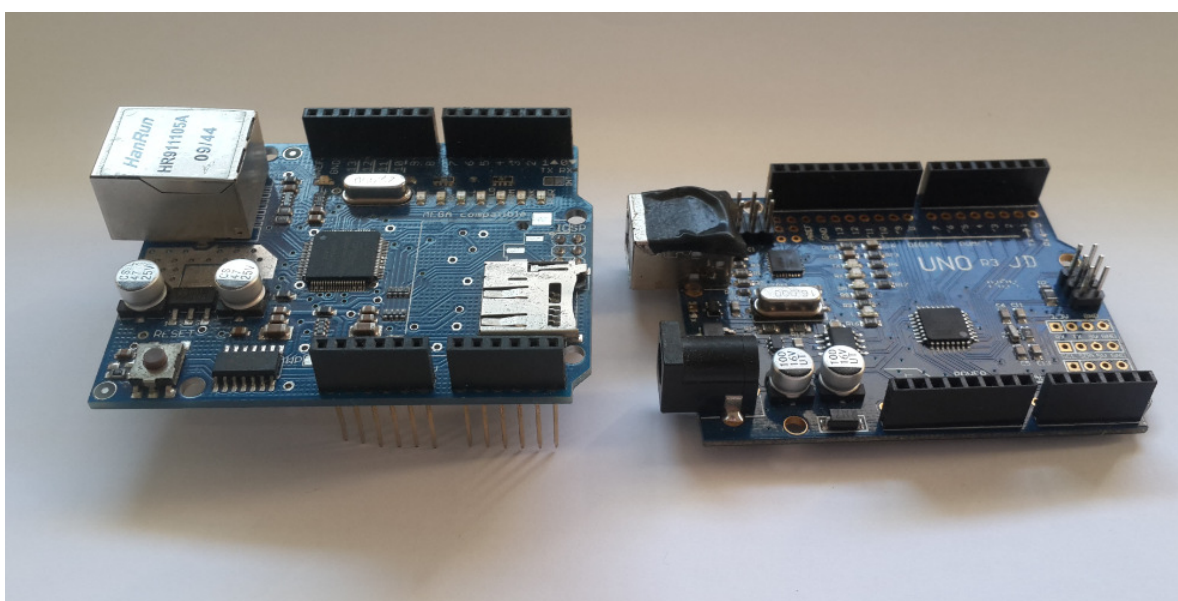


Figura 9 – Placas Arduino Uno com o *shield* Arduino Ethernet.

Fonte: Autoria própria.

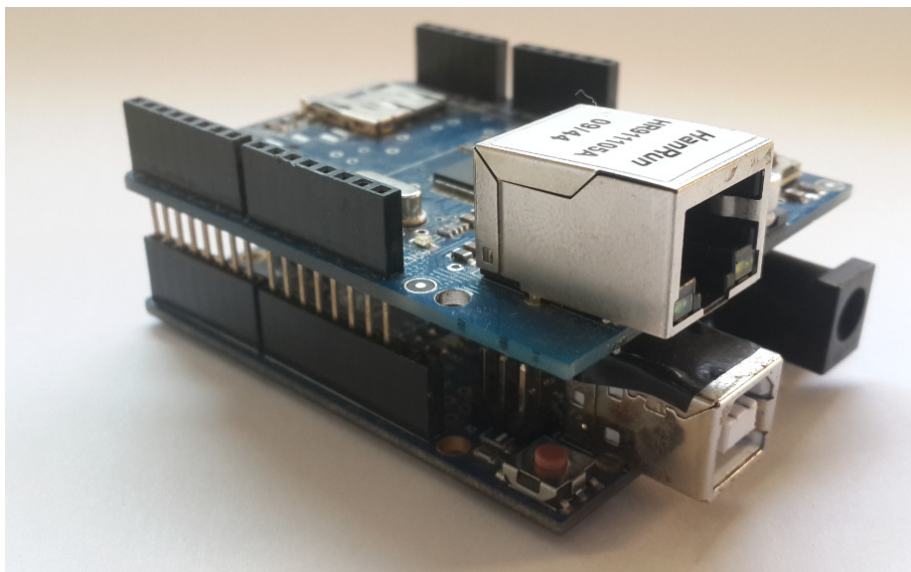


Figura 10 – Placas Arduino Uno e *shield* Arduino Ethernet conectadas.

Fonte: Autoria própria.

É necessário entrar no *site* do fabricante das placas e baixar o *software* de desenvolvimento. Esses dados encontrassem no *link* <<http://arduino.cc/en/Main/Software>>.

Para que o Arduino passe a aceitar comandos realizado via celular é necessário instalar uma biblioteca do sistema que pode ser facilmente encontrada no *site* TouchOSC (biblioteca "Open Sound Control"), através do *link* <<http://hexler.net/software/touchosc-android>>.

### 3.6 INSTALAÇÕES ELÉTRICAS E COMPONENTES

Para a instalação do comando de uma lâmpada foi utilizado um relé em paralelo com o interruptor. Caso, por algum motivo, o sistema não esteja respondendo, o usuário ficará livre para utilizar o sistema da forma convencional, pois o relé de contato funciona em paralelo com um dos contatos normalmente abertos e outro normalmente fechado do interruptor e, quando acionado, os mesmos se invertem. O diagrama da instalação pode ser visto na Figura 11.

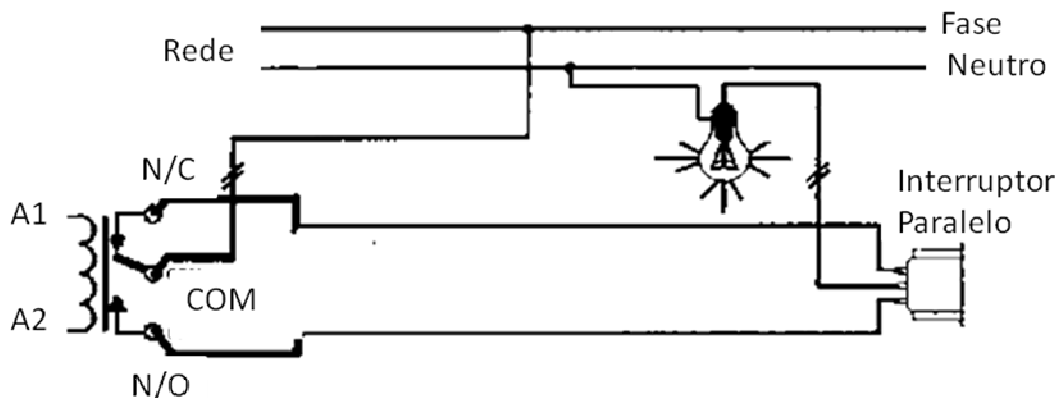
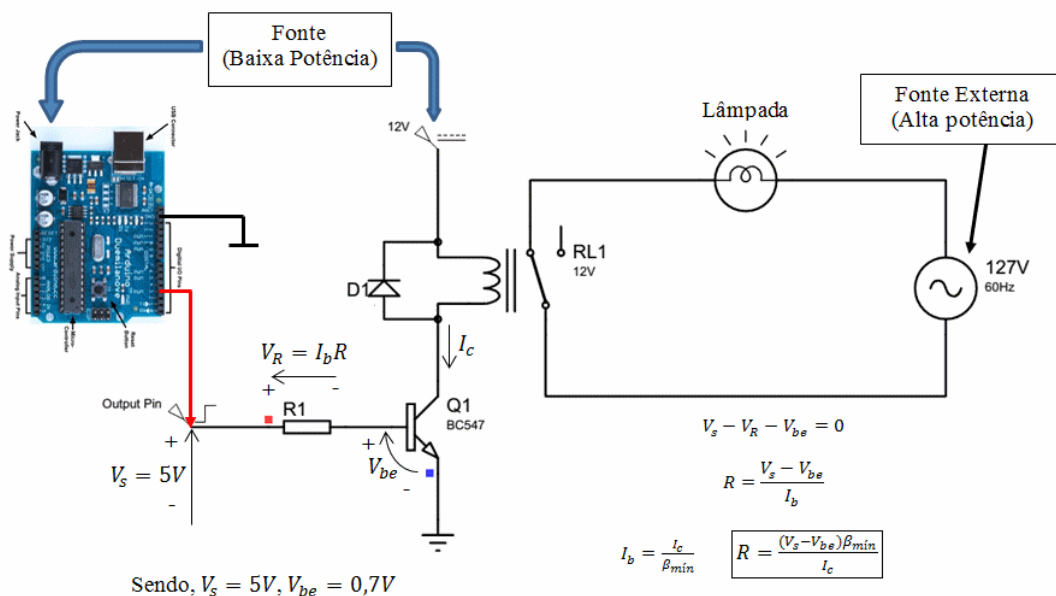


Figura 11 – Diagrama de funcionamento do relé.

Fonte: Autoria própria.

Para acionar o relé foi projetado um circuito elétrico de acordo com o modelo da Figura 12. A Equação utilizada para dimensionar o resistor foi para um transistor BC547 que tem sua base excitada com corrente 30 mA, conforme especificado pelo manual do fabricante.



$$R = \frac{(5 - 0,7)110}{30,0mA} = 15,8k\Omega \rightarrow R < 15,8k\Omega$$

Figura 12 – Acionamento do Relé.

Fonte: Vianna Santos, 2013.



Para o comando do relé é utilizado o Arduino com uma placa *shield* para sistema de potência. Nesse caso, pode ser utilizado comandos ópticos ou arranjo de transistor NPN ou PNP. A Figura 13, apresenta uma placa que condiciona o sinal do Arduino de 5 V para 12 V, necessário para comandar os relés. Já a Figura 14 apresenta a placa adaptadora para entrada das tensões de alimentação.

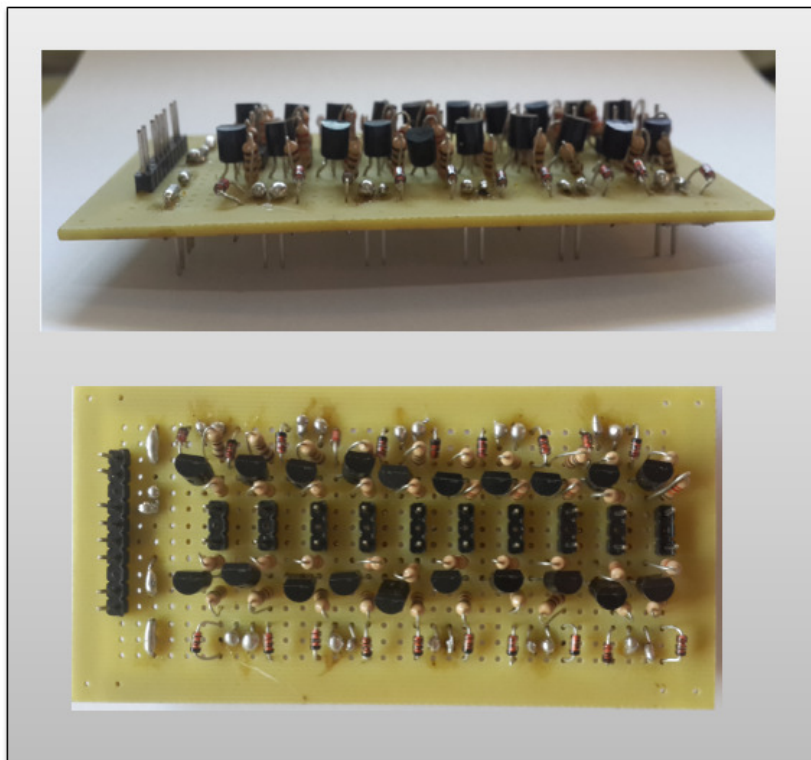


Figura 13 – Placa potência.

Fonte: Autoria própria.

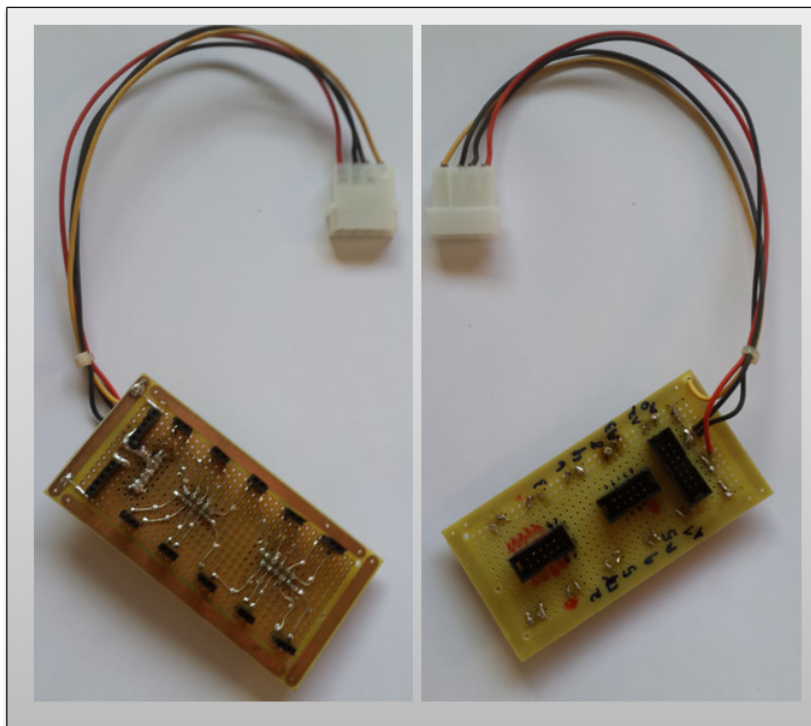


Figura 14 – Placa adaptadora de conexões.

Fonte: Autoria própria.

Outro modelo de placa adota é uma Interface Eletrônica Optoacoplada, que pode ser vista na Figura 15. A placa foi desenvolvida no NUPET (Núcleo de Pesquisas Tecnológicas da UTFPR – Campus Curitiba) para utilização nas disciplinas relacionadas em sistemas de automação baseada em computador. A placa pode ser alimentada com uma fonte 12 V contínua (CC) e a tensão alternada (CA) para alimentar as cargas nos relés de 127 V ou 220 V.

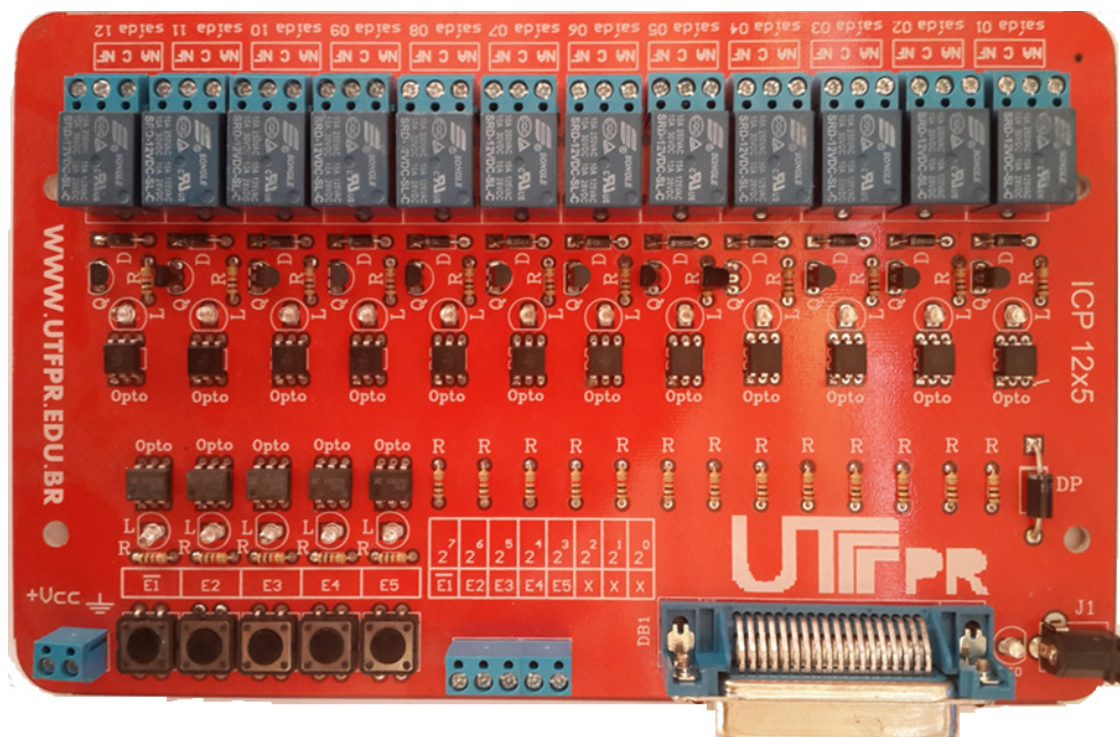


Figura 15 – Placa de Atuadores.

Fonte: Autoria própria.

Este modelo de placa se diferencia do modelo citado anteriormente em dois fatores: primeiro a placa vem com optoacoplador; segundo os atuadores são feitos para serem acionados diretamente na placa, não podem ser levados até a carga. A princípio a placa havia sido desenvolvida para ser conectada diretamente ao computador. Entretanto, neste projeto foi desenvolvido um adaptador que habilita os comandos da placa, que passam a ser atuados através do Arduino.

A conexão da placa de atuadores com Arduino só foi possível através da codificação de sua pinagem de entrada, modelo pelo qual segue um padrão de portas paralelas. Através deste padrão foi desenvolvida uma placa específica que interage com a pinagem do Arduino e a placa de atuadores. Na Figura 16 é apresentada a imagem do adaptador.

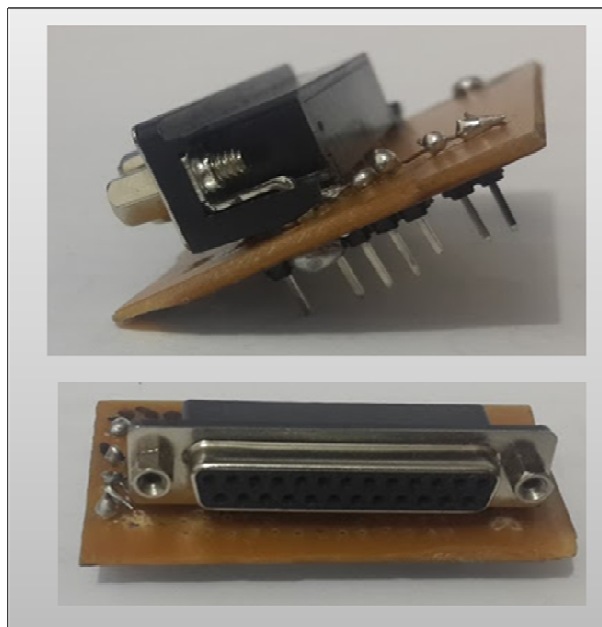


Figura 16 – Adaptador

Fonte: Autoria própria.

O adaptador foi desenvolvido para facilitar o acoplamento e é similar a uma porta paralela. Porém, difere-se de uma porta paralela padrão no quesito de comunicação, contando com a função apenas de envio de sinais, dos quais podem ser de níveis lógicos alto ou baixo. No caso, os valores utilizados estão entre 0 V e 5 VCC.

O modelo das placas seguiu o padrão da placa Arduino Uno, possibilitando que as placas fossem encaixadas uma sobre as outras. Na Figura 17 é apresentado o diagrama em blocos simplificado do sistema proposto neste TCC, formado pelos módulos Arduino, placa de adaptadora e de potência, placa de relé e dispositivo a ser controlado.

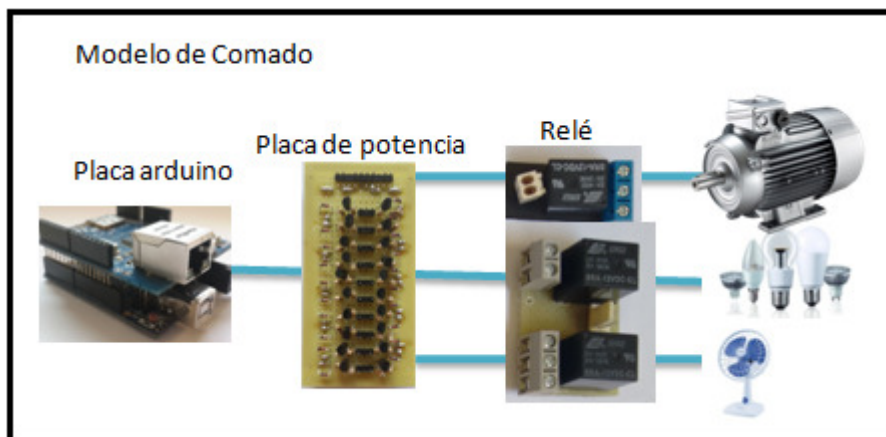


Figura 17 – Diagrama em blocos do modelo de comandos.

Fonte: Autoria própria.

Para a comunicação entre o usuário e a central foi utilizado um celular conectado via Wi-Fi. O Arduino foi conectado com o mesmo roteador Wi-Fi via cabo. Com um aplicativo instalado no celular e com a programação específica da residência rodando no aplicativo, o usuário escolhe um dos comandos que o aplicativo oferece, como lâmpada, tomada, ventilador, entre outros. Cabe ressaltar que somente o acionamento de lâmpada foi implementado neste TCC.

O comando é enviado através da conexão Wi-Fi do celular, que é recebido posteriormente pelo roteador e transmitido via cabo para a placa Arduino. O sinal é interpretado pelo Arduino que envia o comando para uma das saídas da placa. Na Figura 18 é apresentada uma representação da comunicação usuário e Arduino.

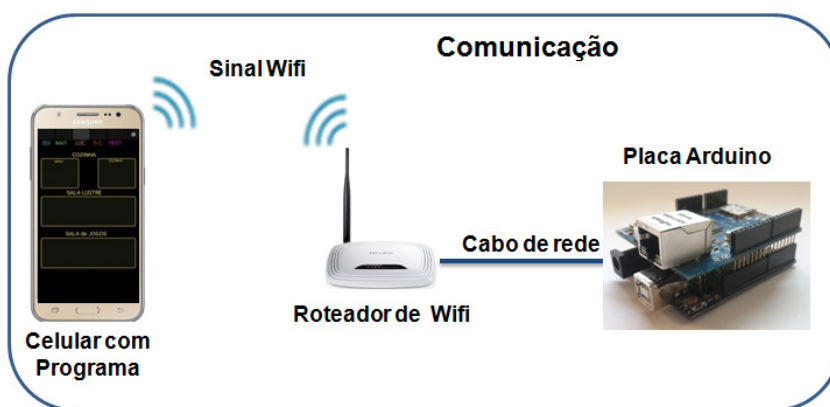


Figura 18 – Diagrama em blocos da Comunicação usuário e placa Arduino.

Fonte: Autoria própria.

## 4 ARQUITETURA DO SISTEMA

### 4.1 PLANEJAMENTO

A integração de tecnologias eletrônicas em residências ocorre através de um processo manual. Como Helal et al. (2005) observam, ao adicionar um novo componente requer uma análise de características de operação, predeterminar os métodos de configurar e integrar, e testar o equipamento o intuito de evitar futuros conflito e comportamentos inesperados.

Partindo do princípio acima, no ano de 2015, este projeto passou por uma implantação experimental para automatizar uma residência. Na época a casa estava em fase de planejamento. Logo foi feito um projeto elétrico, juntamente com o projeto de lógica, que conta com o sistema utilizado na automação, este projeto foi realizado a priori no software AutoCAD.

Para a implantação do sistema, o projeto teve que ter algumas modificações visando facilitar a passagem dos cabos e adaptação de caixas para os módulos.

Os ocupantes da residência foram orientados a monitorar possíveis falhas sem intervir. Para facilitar a manutenção do sistema, foi incluída uma chave no quadro de lógica da casa para desligar o sistema, funcionando em paralelo com os equipamentos. Logo, a casa continua a funcionar sem a automação, caso necessário.

### 4.2 CAMADA DE COMPONENTES FÍSICOS

Esta camada contém todos os componentes físicos que compõem o sistema. Dentre eles estão os (interruptores, relés, placa de comandos, fonte de

alimentação, e cabos) e atuadores (motores, solenoides, etc.). Esta etapa é necessária para que ocorra a conversão de sinais elétricos em ações.

#### 4.3 CAMADA DE TRANSDUTORES E COMUNICAÇÃO

A camada de transdutores é responsável pela conversão de sinais elétricos. BOLZANI, (2010) citam dois aspectos essenciais para ser priorizados com relação as camadas de comunicação que devem ser: possibilitar a comunicação entre os dispositivos; e abrir a conexão dos dispositivos para rede externa. Cook et al. (2003) também agregam uma necessidade que envolve a criação de uma conexão entre a casa e o usuário.

#### 4.4 DESCRIÇÃO DA PROGRAMAÇÃO DO ARDUINO

A programação do Arduino foi feita do seguinte método: primeiro o sistema cria um servidor (*server*) e define as portas de comunicação. Em seguida, libera as saídas para sinal lógico alto ou baixo e habilita uma variável para receber o comando do TouchOSC. Com isso, cada variável criada tem uma correspondente no Arduino. Na sequência, cria-se as funções de comando que analisam o estado da variável que foi enviada pelo TouchOSC, alterando o valor lógico das portas do Arduino.

#### 4.5 DESCRIÇÃO FUNCIONAL

Ao observar a Figura 19, que representa o diagrama simplificado do sistema, pode-se observar que os comandos ocorrem via rede Wi-Fi, após o comando do usuário, que é traduzido em dados. O mesmo pode ser realizado

diretamente nos interruptores, sendo que os comandos realizados pelo celular ocorrem via rede doméstica e são analisados pelo Arduino que, em seguida, passa para os atuadores que modificam o ambiente, seja ligando uma lâmpada, abrindo uma porta, janela ou cortina.

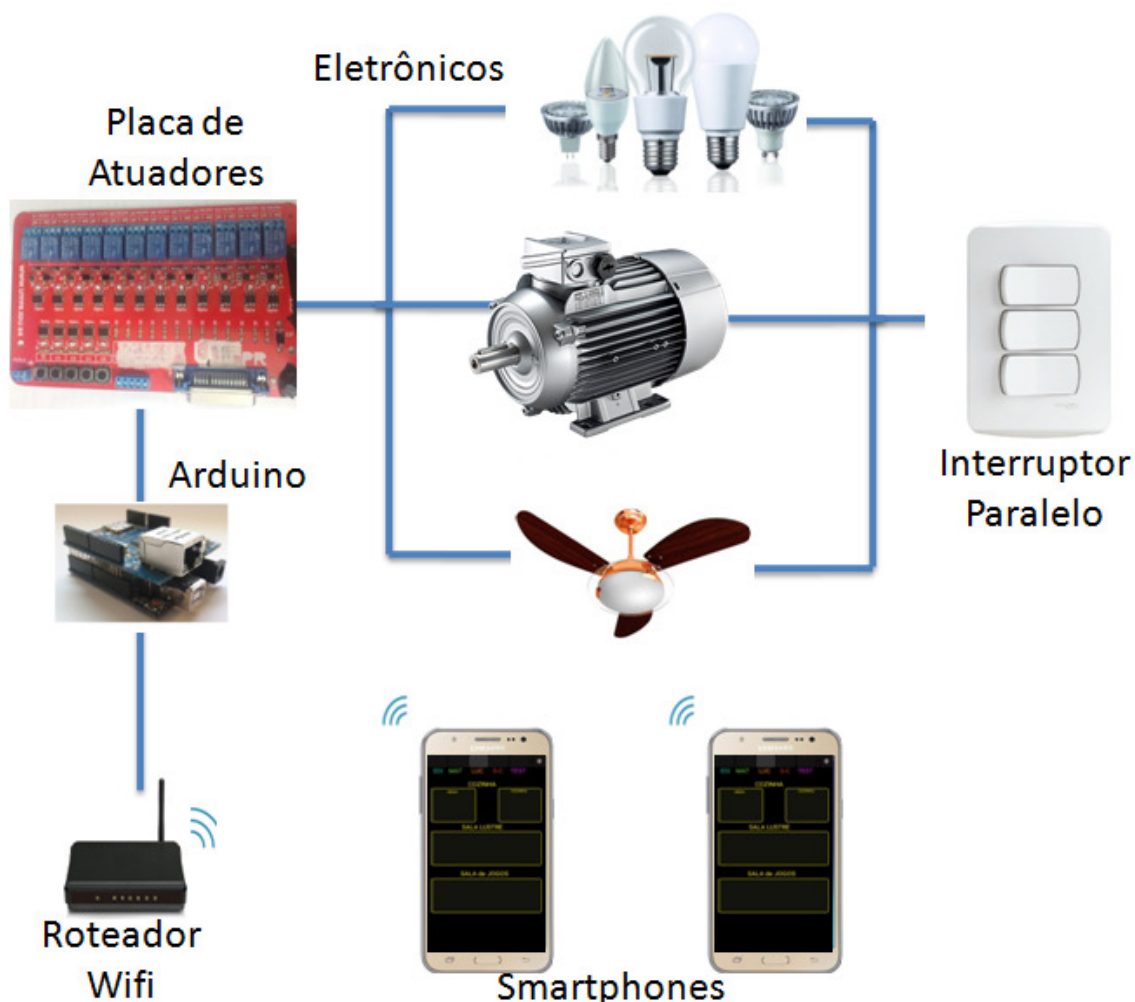


Figura 19 – Diagrama de funcionamento.

Fonte: Autoria própria.

Por se tratar de um sistema eletrônico dependente de celular (que pode vir a ficar sem bateria), a rede Wi-Fi pode ter alguma falha. Caso isso ocorra, o circuito da residência permanece funcionando de maneira tradicional através das teclas dos interruptores.



#### 4.6 COMPONENTES E CARACTERÍSTICAS DO PROJETO

A escolha dos componentes utilizado neste projeto estão relacionadas a alguns fatores dos quais envolvem, entre outros, o custo e a flexibilidade. Um dos componentes utilizado para programação foi o TouchOSC. O aplicativo é simples e de fácil de programação, pois trabalha com blocos de comandos. Entretanto, está não é sua única vantagem. A principal escolha do aplicativo é devido a sua flexibilidade quanto a sua padronização para ser implantado no Androide ou no iOS, pois o programa principal é disponível na loja do Google Play ou na Apple Store. Além disso, já vem predefinido os parâmetros de funcionamento de programação.

As telas de programação (Quadro A e B) podem ser vistas na Figura 20. No Quadro A é apresentada a tela da plataforma que é instalada no computador. No Quadro B é mostrada a plataforma que é baixada das lojas de aplicativos *online* de celular.

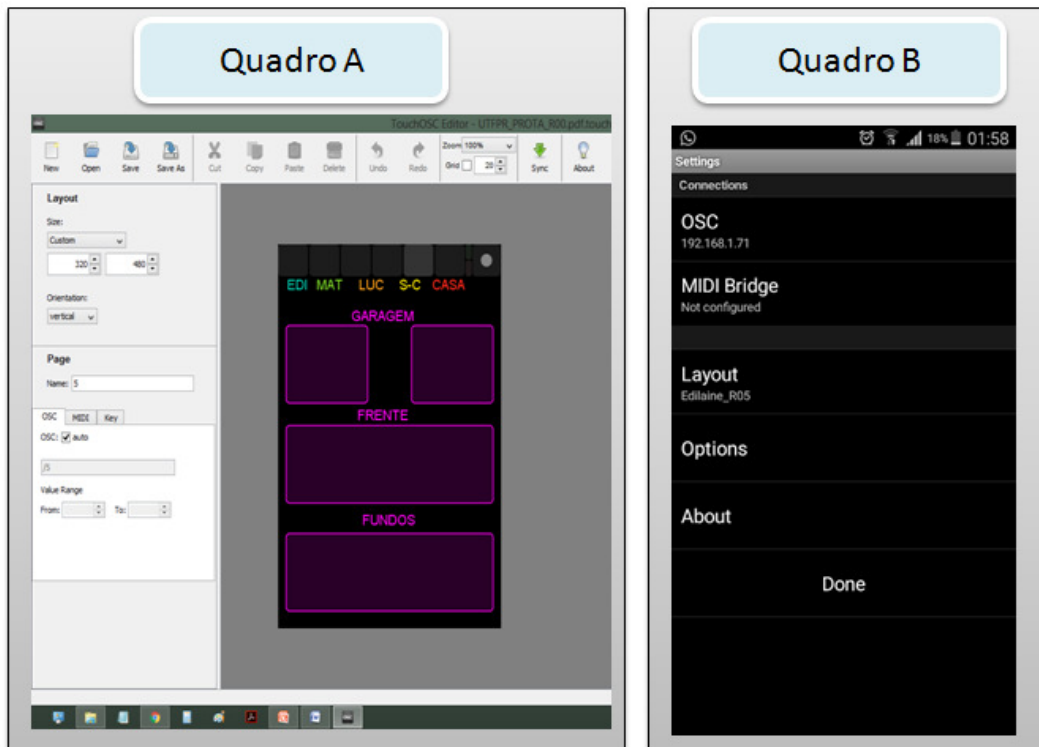


Figura 20 – Quadros do programa TouchOSC.

Fonte: Autoria própria.

Para a utilizar o programa que foi desenvolvido no computador para comandar os atuadores, basta habilitar a opção Sync do programa, que está instalado no PC conectado à rede. Em seguida, conectar o celular com o TouchOSC instalado na mesma rede e, com a opção (*Layout*), adicionar o IP do computador e pressionar em *download*. O aplicativo do celular baixa o *layout* criado no computador e fica pronto para se comunicar com o Arduino. O procedimento é o mesmo, tanto para Android quanto iOS. A grande vantagem é que não é necessário buscar uma nova autorização das lojas de aplicativos *online* para disponibilizar uma nova versão *layout*.

Como desvantagem, não se tem acesso direto ao código fonte do aplicativo Android, apenas às variáveis de comando e seus respectivos valores. Toda a lógica envolvida no acionamento fica a cargo da programação feita no Arduino.

A principal vantagem na escolha da plataforma Arduino foi devido ao seu custo de instalação, pois elimina a necessidade de um computador ou CLP ligado para coletar os dados e executar os comandos. Outro aspecto que justificou a escolha foi a flexibilidade para adicionar novos componentes, como, por exemplo, um componente que permite a conexão com um roteador Wi-Fi Arduino *shield*.

Conforme mencionado anteriormente, neste projeto foi adotado o paralelismo no comando das cargas. Assim, caso o sistema venha a falhar, ou não funcionar adequadamente, o usuário pode desativá-lo e continuar utilizando os equipamentos normalmente.

Na Figura 21 é possível observar o relé que acionara a lâmpada. Este relé receberá um sinal de 12V com corrente de 3 mA da placa de comandos, e esta do Arduino. Desta forma, obteve-se uma redução no custo de cabeamento e eletrodutos, uma vez que a fiação entre o Arduino e a placa de relé chegam no interruptor com um sinal de 12 V (CC). Caso fosse optado por levar o sinal de 127 V (CA) à Lâmpada, precisar-se-iam de mais condutores e eletrodutos.

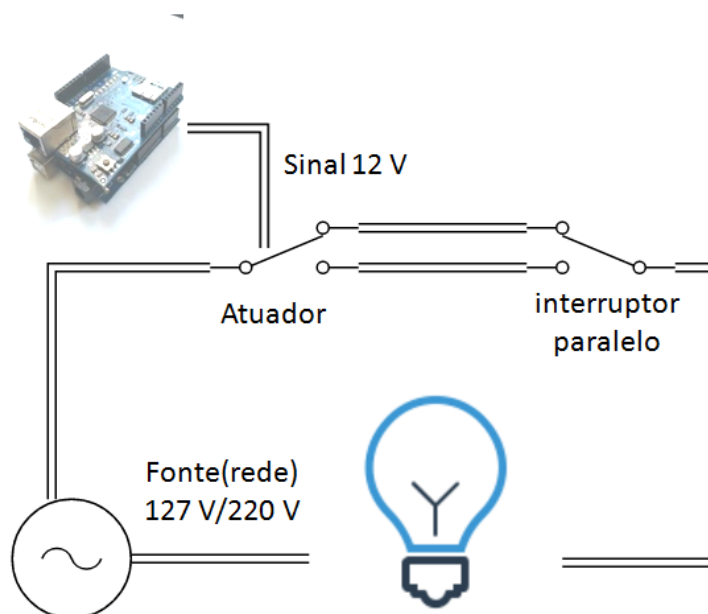


Figura 21 – Diagrama de acionamento de relé.

Fonte: Autoria própria.

#### 4.7 TESTE EM LABORATÓRIO

Nos testes do sistema em laboratório, a corrente elétrica para habilitar os sensores optoacoplador ficou entre 0,025 mA a 0,03 mA por sensor. Para ativar os comutadores, que utilizaram tensão nominal de 12 V, a corrente medida ficou entre 15 mA e 20 mA. Também foi verificado os níveis de aquecimento dos comutadores, não ultrapassando o valor de 60 °C.

Uma das falhas que ocorreu foi na placa de relés, que recebe o sinal do Arduino. A primeira placa foi montada sem o acoplamento óptico. Logo, todos os componentes (Arduino, placa de comando e atuadores) utilizados no sistema compartilhavam a mesma referência GND. Quando ocorreu a falta deste, todos os relés foram desligados, juntamente com a placa Arduino e o roteador Wi-Fi. Quando a energia é retomada, todos os comandos permanecem desligados e, em alguns casos, foi necessário reiniciar o Arduino manualmente para que a conexão com roteador Wi-Fi fosse restabelecida.

A maioria dos problemas encontrados ao utilizar o projeto foi relacionada às falhas mecânicas nas conexões mecânicas do equipamento, pois a caixa padrão do interruptor é muito pequena e força os contatos, que por sua vez tendem a se soltarem. Para solucionar este problema quando possível adotar caixas maiores e modificar o padrão utilizado para conexão dos cabos.

O sistema operou por mais de 6 meses sem a necessidade de intervenção pelo usuário. Um dos motivos no qual foi necessário a manutenção do sistema está relacionada a interrupção de energia pela concessionária. Quando este fator ocorre, é necessário pressionar o botão *reset* da placa para que o mesmo volte a se conectar ao roteador Wi-Fi.

Outra falha está relacionada ao programa instalado no celular do usuário. Caso o usuário venha a restaurar as configurações de fábrica do *smartphone*, o mesmo terá que entrar em contato com os desenvolvedores para que o *layout* da sua residência seja adicionado novamente em seu aparelho.

## 5 CONCLUSÕES

As possibilidades tecnológicas desenvolvidas durante os últimos anos, apresentadas e discutidas ao longo deste trabalho, avançaram para um desenvolvimento do setor de automação residencial. O cenário apresentado mostrou a necessidade e vantagem de sistemas de residências automatizadas, principalmente para facilitar a vida das pessoas, o que possibilitou entender que, além de acessibilidade, pode-se perceber a automação como uma ferramenta que melhora o bem-estar, conforto e economia para seus usuários, aumentando a interação entre os moradores e suas residências, sejam estes idosos, crianças.

Deste modo, o desenvolvimento de um sistema para a automação na iluminação de uma residência aliado a facilidade tecnológica na utilização de um *smartphone* para o controle no ambiente residencial, proposto neste estudo, foi alcançada através de uma estrutura econômica e adequada e a uma integração entre aplicativo e código desenvolvido.

A plataforma de programação IDE Arduino software, que permitiu a atuação do Arduino+shield, seguiu o estabelecido na estrutura proposta inicialmente. Como solução para redução de custos, uma das medidas adotada foi a instarção dos atuadores dentro das caixas de luz para interruptores e tomadas 2x4. Em muitos casos quando tinha mais do que dois atuadores na mesma caixa eram comuns o rompimento dos fios de comando ao fechar. Outro problema encontrado está relacionado falta de energia. Nesse caso, são desabilitados todos os atuadores e, em alguns casos, cria-se a necessidade de reiniciar o Arduino para possibilitar a conexão com o roteador.

O maior problema encontrado para reduzir os custos do projeto está relacionado ao valor da mão de obra. O valor somado de todos os componentes para acionar dez lâmpadas ficou próximo de R\$ 800,00, ao passo que o custo da mão de obra ficaria em R\$ 180,00 por dia, sendo gastos quinze dias, totalizando um valor com a mãe de obra de R\$ 2700,00. Logo podemos perceber que a mão de obra representa quase o triplo se compara ao valor dos componentes.

Apesar das dificuldades encontradas os objetivos propostos foram alcançados com algum sucesso. O protótipo apresentou limitações por conter um número limitado de pinos para o acionamento. No entanto através da linguagem de programação e do software, que estão preparados para a leitura específica de um caractere da *string* recebida, poderia melhorar a forma que é feita a leitura, desenvolvendo uma nova função de leitura para cada novo botão implementado.

## 5.1 TRABALHOS FUTUROS

A automação residencial é uma derivação da automação predial com foco em atender necessidades domésticas, com projetos de controle em equipamentos, como portas, janelas, televisores, sistemas de climatização e irrigação e outros. Grande parte dos comandos são realizados por meio de relés de contato que são ativados através de uma central que recebe o comando do usuário e repassa para seus periféricos atuarem. A central é programada para realizar ações pré-definidas ou mesmo através de comando externos realizados pelo usuário ou mesmo por sensores no meio.

No futuro, pode ser possível a modificação do projeto para se tornar um sistema com inteligência artificial (IA) aplicada na automação, utilizando, por exemplo, sensores de captura que visão para atender as necessidades do usuário e manipular relés e sensores para modificar o ambiente.

## REFERÊNCIAS

ABOWD, Gregory D.; STERBENZ, James PG. Final report on the inter-agency workshop on research issues for smart environments. **IEEE Personal Communications**, v. 7, n. 5, p. 36-40, 2000.

AGÊNCIA NACIONAL DE TELECOMUNICAÇÕES. 2013. **Brasil ultrapassa um celular por habitante**. Disponível em: <<http://www.anatel.gov.br/Portal/exibirPortalNoticias.do?acao=carregaNoticia&codigo=21613>>. Acessado em: 19 set. 2015.

ANDRADE, Anderson Vinicius Toledo. **Principais Desafios Na Automação Residencial: Prototipação Com Arduino**. 2013.

APPLE. **Apple Iphone Mobile Phone, Ipod and Internet Device**. 2008. Disponível em: <<http://www.apple.com/iphone>>. Acessado em: 15 mar. 2016.

BANZI, Massimo; **Getting Started with Arduino, 2nd Edition**. Maker Media, Inc, 2011.

BANZI, Massimo; SHILOH, Michael. **Primeiros Passos com o Arduino–2ª Edição**. Novatec Editora, 2015.

BOLZANI, Caio AM. **Residências Inteligentes: Domótica; Redes Domésticas; Automação Residencial**. São Paulo: Editora Livraria da Física, 2004.

BOLZANI, Caio Augustus Moraes. **Análise de Arquiteturas e Desenvolvimento de uma Plataforma para Residências Inteligentes**. 2010. Tese de Doutorado. Universidade de São Paulo.

BRUCHER, G. The electric home of the future. **Popular Mechanics**, v. 72, n.2, 1939.

COOK, D. et al. Mavhome: An agent-based smart home. **Pervasive Computing and Communications, 2003. (PerCom 2003). Proceedings of the First IEEE International Conference on**, 2003

- DEMOGRÁFICO, IBGE Censo. 2013. Disponível em:<  
<http://www.sidra.ibge.gov.br/pnad/pnadtic.asp>>. Acessado em: 19 set. 2015.
- FLETCHER, C. Epicurean high-tech. **Spectrum, IEEE**, v. 22, n. 5, p. 81–85, 1985.
- FRIEDEWALD, M.; COSTA, O. D. Science and technology roadmapping: Ambient intelligence in everyday life. **Unpublished working paper, Institute for Prospective Technological Studies (IPTS)**, 2003.
- GONZÁLEZ GARCÍA, Marta Isabel; LUJÁN LÓPEZ, José Luis; LÓPEZ CEREZO, José Antonio. **Ciência, tecnologia y sociedad: una introducción al estudio social de la ciencia y la tecnología**. Tecnos, 2000.
- HARPER, Richard. Inside the smart home: Ideas, possibilities and methods. In: **Inside the smart home**. Springer London, 2003. p. 1-13.
- HELAL, S. et al. The gator tech smart house: A programmable pervasive space. **Computer**, p. 50–60, 2005.
- KIDD, C.; ORR, R.; ABOWD, G. The aware home: A living laboratory for ubiquitous computing research. **Proceeding of 2nd International Workshop on Cooperative Buildings**, n. 191, p. 198, 1999.
- MACHADO, Silvana R. B.; MELUL, Maryse. **Computação gráfica em escritórios de projetos informatizados**. Rio de Janeiro: Brasport, 2005. 690 p.
- MATEAS, M. et al. Engineering ethnography in the home. CHI '96: **Conference companion on Human factors in computing systems**, p. 283–284, 1996.
- MELOSI, M. **Thomas A. Edison end the Modernization of America**. [S.l.]: Longman, 1990.
- MONK, S. 2013. **Programação com Arduino >>começando com sketches**. Porto Alegre, RS: BOOKMAN EDITORA LTDA.
- MORIMOTO, Carlos E. **Entendendo a arquitetura de hardware dos smartphones**. 2010. Disponível em: <



<http://www.hardware.com.br/dicas/arquitetura-hardware-smartphones.html>>.

Acessado em: 12/10/2016.

MURATORI, José Roberto; DAL BÓ, Paulo Henrique. **Capítulo I Automação residencial: histórico, definições e conceitos. O setor elétrico**, p. 70, 2011.

NORVIG, P.; RUSSEL, S. **Artificial Intelligence: A Modern Approach**. [S.l.]: Prentice Hall, 2002.

OLIVEIRA NETO, Reinaldo. **Automação de iluminação residencial utilizando microcontrolador arduino e tablet ipad via wi-fi**. 2011.

RAMOS, Amanda Lúcia Carstens; SANTOS, José Eduardo Lima dos. Sistema integrado de automação residencial com comunicação sem fio. 2015.

RIBEIRO DANIEL. **Como funciona um roteador e saiba quais os tipos existentes**. 2013. Disponível em: <<http://www.techtudo.com.br/artigos/noticia/2013/05/como-funciona-um-roteador-e-saiba-quais-os-tipos-existentis.html>>. Acessado em: 14/10/2016.

RICQUEBOURG, V. et al. The smart home concept: Our immediate future. **E-Learning in Industrial Electronics, 2006 1ST IEEE International Conference on**, p. 23–28, 2006.

SILVA, F. B.; FERREIRA, W. R. Parques urbanos de Uberlândia: estudo de caso no Parque Municipal Victorio Siquierolli. In: SIMPÓSIO REGIONAL DE GEOGRAFIA: PERSPECTIVAS PARA O CERRADO NO SÉCULO XXI, 2, 2003, Uberlândia. **Anais...** Uberlândia: UFU, 2003. p.01-11.

SOUZA, Adriano Fagali de; ULBRICH, Cristiane Brasil Lima. **Engenharia Integrada por Computador e Sistemas CAD/CAM/CNC – Princípios e Aplicações**. São Paulo, Artliber, 2009, 335p.

STIMSON, J. T. A house of magic. **Popular Mechanics**, 1954.

TANENBAUM, ANDREW S. **Redes de computadores**. Vrije Universiteit Amsterdam, Holanda. Tradução: Vandenberg D. de Souza; Analista de sistemas e tradutor Editora Campus, 4ª ed, 2003.

VIANNA SANTOS, GUILHERME. 2013. **Como acionar um relé com arduino ou pic projeto e funcionamento**. Disponível em <<http://eletronicaemcasa.blogspot.com.br/2013/12/como-acionar-um-rele-com-arduino-ou-pic.html>>. Acessado em: 12/11/2016

WEISER, M. The computer for the 21st century. **Scientific American**, v. 265, n. 3, p. 66–75, 1991.