

**UNIVERSIDADE FEDERAL DE SANTA MARIA
CAMPUS CACHOEIRA DO SUL
CURSO DE ENGENHARIA ELÉTRICA**

Leonardo Adam

**SISTEMA SUPERVISÓRIO PARA AUTOMAÇÃO
RESIDENCIAL**

**Cachoeira do Sul, RS, Brasil
2019**

Leonardo Adam

SISTEMA SUPERVISÓRIO PARA AUTOMAÇÃO RESIDENCIAL

Trabalho de Conclusão de Curso Apresentado ao Curso de Graduação em Engenharia Elétrica, da Universidade Federal de Santa Maria (UFSM) - Campus Cachoeira do Sul, como requisito parcial para obtenção do grau de **Bacharel em Engenharia Elétrica**.

Orientador: Prof. Dr. Paulo César Vargas Luz

**Cachoeira do Sul, RS, Brasil
2019**

Ficha catalográfica elaborada através do Programa de Geração Automática da Biblioteca Central da UFSM, com os dados fornecidos pelo(a) autor(a).

Adam, Leonardo

Sistema Supervisório para Automação Residencial /

Leonardo Adam - 2019

111 p.; 30 cm

Orientador: Paulo César Vargas Luz

TCC (graduado) – Universidade Federal de Santa Maria -
Campus Cachoeira do Sul, Curso de Engenharia Elétrica, RS,
2019

1.Engenharia Elétrica 2. Automação 3. Supervisório 4.
Residência Inteligente 5. Internet das Coisas I. Luz, Paulo Cesar.
II. Título.

© 2019

Todos os direitos autorais reservados a Leonardo Adam. A reprodução de partes ou do todo deste trabalho só poderá ser feita com autorização por escrito do autor.

Endereço: Rua João Trevisan, Nº 1218, Bairro Soares, Cachoeira do Sul, RS, Brasil, CEP:
96501-502;

Fone: (51) 999926274;

Endereço Eletrônico: leonardo.adam96@icloud.com.

Leonardo Adam

SISTEMA SUPERVISÓRIO PARA AUTOMAÇÃO RESIDENCIAL

Trabalho de Conclusão de Curso Apresentado ao Curso de Graduação em Engenharia Elétrica, da Universidade Federal de Santa Maria (UFSM) - Campus Cachoeira do Sul, como requisito parcial para obtenção do grau de **Bacharel em Engenharia Elétrica**.

Aprovado em 10 de Julho de 2019:

Paulo César Vargas Luz, Dr. (UFSM)
(Presidente/ Orientador)

Dion Lenon Prediger Feil, Me. (UFSM)

William Dotto Vizzotto, Eng. (UFSM)

**Cachoeira do Sul, RS, Brasil
2019**

RESUMO

SISTEMA SUPERVISÓRIO PARA AUTOMAÇÃO RESIDENCIAL

AUTOR: LEONARDO ADAM
ORIENTADOR: PAULO CÉSAR VARGAS LUZ

Este trabalho apresenta um sistema supervisório criado para gerenciar soluções em automação residencial. Foi utilizado o software Mango Automation, com o qual é possível realizar acionamentos, agendar tarefas, criar avisos, monitorar variáveis, entre outras funcionalidades. É apresentada uma revisão bibliográfica sobre os conceitos e componentes da automação residencial e de internet das coisas. Ainda são apresentados alguns sistemas de automação residencial atualmente encontrados no mercado. Também foram analisados e comparados *softwares* compatíveis com a aplicação desejada. A partir da análise de dispositivos e aparelhos habitualmente encontrados nas residências foram desenvolvidas soluções em automação residencial. Estão descritos os equipamentos utilizados para realizar a automação, os circuitos práticos e as lógicas de programação utilizadas. Por fim são apresentadas as configurações necessárias para a integração com o sistema supervisório, as praticidades que o uso desse tipo de sistema traz e os resultados obtidos.

Palavras-chave: Automação. Sistema Supervisório. Residência Inteligente. Internet das Coisas.

ABSTRACT

SUPERVISORY SYSTEM FOR HOME AUTOMATION

AUTHOR: LEONARDO ADAM
ADVISOR: PAULO CÉSAR VARGAS LUZ

This document presents a supervisory system created to manage home automation solutions. The software Mango Automation was used to design the supervisory system, with which is possible to drive devices, schedule events, create warnings, monitorate variables and another functionalities. A review is presented about the concepts and components of home automation and IoT. Some home automation systems currently found on the market are presented. Softwares able to the application were analysed and compared. Smart devices for home automation were developed. The equipments used to perform the automation, the practical circuits and the programming logic are described. Finally, the necessary configuration to link the home automation to the supervisory system, as well as the practical and final results are presented.

Keywords: Automation, Supervisory System, Smart Home, Programmable Logic Controller, Internet of Things.

LISTA DE FIGURAS

Figura 2.1 – Pirâmide da automação.	18
Figura 2.2 – Exemplo de diagrama de automação.	19
Figura 2.3 – Sistema supervisório residencial.	20
Figura 2.4 – Relação entre a população e dispositivos conectados à internet.	21
Figura 2.5 – Relé SONOFF básico.	22
Figura 2.6 – Aplicativo eWeLink.	23
Figura 2.7 – Tomadas inteligentes Broadlink.	24
Figura 2.8 – Interruptores inteligentes Broadlink.	25
Figura 2.9 – Controle remoto universal Broadlink.	25
Figura 2.10 – Aplicativo e-Control.	26
Figura 2.11 – Tomada inteligente Wall Plug.	27
Figura 2.12 – Single e Double Switch.	28
Figura 2.13 – Dimmer 2.	28
Figura 2.14 – Roller Shutter 3.	29
Figura 2.15 – Home Center 2.	30
Figura 2.16 – App Fibaro para computadores.	30
Figura 2.17 – App Fibaro para <i>smartphones</i> .	31
Figura 2.18 – Representação simplificada de sistema supervisório.	32
Figura 2.19 – Exemplo de tela sinótica.	33
Figura 2.20 – Tela gráfica no ScadaBR.	36
Figura 2.21 – Sistema supervisório no <i>Mango Automation</i> .	36
Figura 2.22 – Tela inicial <i>Mango Automation</i> .	37
Figura 2.23 – Configuração de <i>Data sources</i> .	38
Figura 2.24 – Interface gráfica Elipse.	40
Figura 2.25 – Elipse Mobile em diversas plataformas.	40
Figura 3.1 – Planta da residência.	43
Figura 3.2 – Arduino UNO.	45
Figura 3.3 – Módulo Ethernet W5100.	45
Figura 3.4 – Módulo ESP-01.	46
Figura 3.5 – Adaptador ESP-01 para USB.	47
Figura 3.6 – Adaptador USB para serial.	47
Figura 3.7 – Base para ESP-01.	48
Figura 3.8 – Módulo NodeMCU.	48
Figura 3.9 – Base adaptadora para NodeMCU.	49
Figura 3.10 – Módulo relé.	50
Figura 3.11 – Módulo relé para ESP-01.	50
Figura 3.12 – Módulo dimmer.	51
Figura 3.13 – Receptor Infravermelho IRM-3638T.	52
Figura 3.14 – LED infravermelho.	52
Figura 3.15 – LDR.	53
Figura 3.16 – Curva da resistência em relação à iluminação.	53
Figura 3.17 – Sensor de chuva YL-83.	54
Figura 3.18 – Sensor DS18B20.	54
Figura 3.19 – DS18B20 à prova d'água.	55
Figura 3.20 – Kit leitor RFID.	55
Figura 3.21 – Mini fechadura 12V.	56
Figura 3.22 – Esquema de ligação do Arduino.	57
Figura 3.23 – Circuito implementado do Arduino.	58
Figura 3.24 – Esquema de ligação de lâmpadas.	59

Figura 3.25 – Circuito implementado para acionamento on/off	59
Figura 3.26 – Sinal de entrada AC e <i>Zero Cross</i>	60
Figura 3.27 – Sinal de saída do dimmer.	61
Figura 3.28 – Esquema de ligação do dimmer.....	62
Figura 3.29 – Circuito implementado para acionamento com Dimmer.....	63
Figura 3.30 – Esquema de ligação para controle de ares condicionados.....	64
Figura 3.31 – Circuito implementado para acionamento de ar condicionado.	65
Figura 3.32 – Esquema de ligação para controle de fechaduras.	66
Figura 3.33 – Circuito implementado para a fechadura eletrônica.	67
Figura 3.34 – Exemplo de utilização protocolo Modbus.	68
Figura 3.35 – Estrutura de dados do protocolo.	68
Figura 3.36 – Configurações de <i>Data Sources</i>	70
Figura 3.37 – Configurações para protocolo Modbus IP.....	71
Figura 3.38 – Configurações de <i>Data Points</i>	72
Figura 3.39 – Tela de <i>Watch Lists</i>	73
Figura 3.40 – Criação de sistemas supervisório em modo <i>Legacy</i>	74
Figura 3.41 – Criação de sistemas supervisório com <i>Dashboard Designer</i>	75
Figura 3.42 – Configurações de <i>Point Links</i>	76
Figura 3.43 – Configurações de eventos agendados.	77
Figura 3.44 – Configurações do <i>Advanced Scheduler</i>	78
Figura 3.45 – Configurações do <i>Event Handler</i>	79
Figura 3.46 – Tela de configuração de usuários do <i>Mango Automation</i>	80
Figura 3.47 – <i>Data points</i> do Arduino.	81
Figura 3.48 – <i>Data point</i> do ESP-01 para acionamento on/off	82
Figura 3.49 – <i>Data points</i> do NodeMCU da fechadura.	82
Figura 3.50 – <i>Data points</i> do ESP-01 para controle do ar condicionado.	82
Figura 3.51 – Sistema supervisório desenvolvido.....	83
Figura 3.52 – Histórico de valores do <i>Data Point</i> referente a fechadura eletrônica.	84
Figura 3.53 – Gráfico da temperatura.	85

SUMÁRIO

1 INTRODUÇÃO	15
1.1 CONSIDERAÇÕES GERAIS	15
1.2 JUSTIFICATIVA	15
1.3 OBJETIVOS	16
1.3.1 Objetivos Gerais	16
1.3.2 Objetivos Específicos	16
2 REVISÃO BIBLIOGRÁFICA	17
2.1 AUTOMAÇÃO RESIDENCIAL	17
2.1.1 Conceito	17
2.1.2 Componentes	17
2.2 INTERNET OF THINGS	20
2.2.1 História	20
2.2.2 Conceito	21
2.3 SOLUÇÕES COMERCIAIS EM AUTOMAÇÃO RESIDENCIAL	22
2.3.1 SONOFF	22
2.3.2 Broadlink	24
2.3.2.1 Tomadas Inteligentes	24
2.3.2.2 Interruptores Inteligentes	24
2.3.2.3 Controles Universais	25
2.3.2.4 Aplicativo	26
2.3.3 Fibaro	26
2.3.3.1 Wall Plug	27
2.3.3.2 Single e Double Switch	27
2.3.3.3 Dimmer 2	28
2.3.3.4 Roller Shutter 3	29
2.3.3.5 Home Center 2	29
2.3.3.6 Aplicativos	30
2.4 SISTEMAS SUPERVISÓRIOS	31
2.4.1 História	31
2.4.2 Conceito	32
2.4.3 Componentes	33
2.4.4 Vantagens	34
2.4.5 Softwares	34
2.4.5.1 ScadaBR	35
2.4.5.2 Mango Automation	36

SUMÁRIO

2.4.5.3 <i>Eipse</i>	39
2.4.6 Comparativo softwares	41
2.5 CONCLUSÕES PARCIAIS	41
3 SISTEMA DESENVOLVIDO	43
3.1 PLANTA	43
3.1.1 Planta Inicial	43
3.1.2 Soluções implementadas	44
3.2 DISPOSITIVOS PARA AUTOMAÇÃO RESIDENCIAL	44
3.2.1 Arduino	44
3.2.2 Módulo Ethernet W5100	45
3.2.3 Módulos Wi-Fi ESP8266	46
3.2.3.1 <i>ESP-01</i>	46
3.2.3.2 <i>NodeMCU</i>	48
3.2.4 Módulo Relé	49
3.2.5 Módulo Dimmer	51
3.2.6 Receptor e Transmissor Infravermelho	51
3.2.7 Sensor de Luminosidade	52
3.2.8 Sensor de Chuva	53
3.2.9 Sensor de Temperatura	54
3.2.10 Leitor RFID	55
3.2.11 Mini Fechadura	56
3.3 APLICAÇÕES	56
3.3.1 Sensoriamento	56
3.3.2 Iluminação	58
3.3.3 Controle de Ar Condicionado	64
3.3.4 Controle de Fechadura	65
3.4 PROTOCOLO MODBUS	67
3.5 SISTEMA SUPERVISÓRIO	69
3.5.1 Configurações	69
3.5.1.1 <i>Data Sources</i>	69
3.5.1.2 <i>Data Points</i>	72
3.5.1.3 <i>Watch Lists</i>	73
3.5.2 Interface Sistema Supervisório	74
3.5.2.1 <i>Dashboard Designer</i>	74
3.5.3 Funcionalidades	75
3.5.3.1 <i>Point Links</i>	75
3.5.3.2 <i>Eventos Agendados</i>	76
3.5.3.3 <i>Advanced Scheduler</i>	77

SUMÁRIO

3.5.3.4 <i>Event Handlers</i>	78
3.5.3.5 <i>Usuários</i>	79
3.5.3.6 <i>Auto Login</i>	80
3.5.4 Atuação do Sistema Supervisório	81
3.6 RESULTADOS	83
4 CONCLUSÃO	87
4.1 TRABALHOS FUTUROS	88
REFERÊNCIAS	89
APÊNDICES	95

1 INTRODUÇÃO

1.1 CONSIDERAÇÕES GERAIS

Com o avanço e popularização da tecnologia, estão sendo criadas ferramentas para uso não somente produtivo, mas também para a melhoria na qualidade de vida e facilidade em tarefas cotidianas. Nesse sentido, sistemas utilizados em empresas estão sendo adaptados e utilizados em ambientes domésticos.

Atualmente, existem muitas tarefas que realizamos diariamente, mas que por estarmos tão habituados a realizá-las, não percebemos o tempo que elas nos consomem todos os dias. Com as novas tecnologias, podemos integrar essas tarefas e realizá-las ou agendá-las para que sejam realizadas automaticamente a partir da automação residencial. Assim, restando mais tempo para realização de outras tarefas ou até mesmo para descanso.

A automação residencial vem crescendo no Brasil nos últimos anos. Pesquisas feitas pela Aureside (Associação Brasileira de Automação Residencial e Predial) apontam um crescimento de cerca de 11,36% entre os anos de 2014 e 2020 (AURESIDE, 2015a). Ainda, é estimado que, no Brasil, cerca de 300 mil casas possuam sistema de automação residencial (AURESIDE, 2015b).

1.2 JUSTIFICATIVA

Com essa popularização, a automação residencial está deixando de ser um item exclusivo de residências de luxo, abrangendo uma gama cada vez maior de usuários.

O sistema de automação deve ser fácil e intuitivo, para que assim possa ser utilizado por qualquer pessoa, dispensando o conhecimento de programação e de sistemas lógicos.

O sistema supervisório traz a facilidade na visualização e controle dos dispositivos. Ao invés de alterar valores dentro de códigos no programa, o usuário pode simplesmente digitar um valor em uma tela ilustrativa e realizar a alteração.

Com a utilização de um sistema supervisório é possível também a comunicação entre o sistema e diferentes dispositivos, reduzindo a incompatibilidade entre diferentes dispositivos, uma vez que dispensa a interligação dos mesmos entre si.

1.3 OBJETIVOS

1.3.1 Objetivos Gerais

- Desenvolver um sistema supervisório de fácil uso e que execute, de forma satisfatória, a exibição do estado dos equipamentos de uma residência e o controle dos mesmos.
- Desenvolver um pequeno sistema de automação residencial, com dispositivos simples e que consigam controlar equipamentos presentes na maioria das residências atuais.

1.3.2 Objetivos Específicos

- Estudo de diferentes dispositivos para o uso na automação residencial.
- Estudo de *softwares* para uso no sistema supervisório.
- Implementação prática do sistema supervisório em controladores lógicos programáveis.

2 REVISÃO BIBLIOGRÁFICA

Neste capítulo é abordado o conceito de automação residencial e estão descritos seus principais componentes. Também é abordada a internet das coisas, a história do surgimento, o avanço e também seu conceito. Por fim, é apresentada uma visão geral sobre sistemas supervisórios, descrevendo a história, seus componentes, suas vantagens, funcionalidades e os principais *softwares* disponíveis para a criação e utilização.

2.1 AUTOMAÇÃO RESIDENCIAL

2.1.1 Conceito

A automação residencial também chamada de residência inteligente, ou domótica, tem como objetivos trazer comodidade, conforto, segurança e economia. Podendo ser definida como um conjunto de tecnologias que auxiliam na gestão e realização de tarefas domésticas (BOLZANI, 2004).

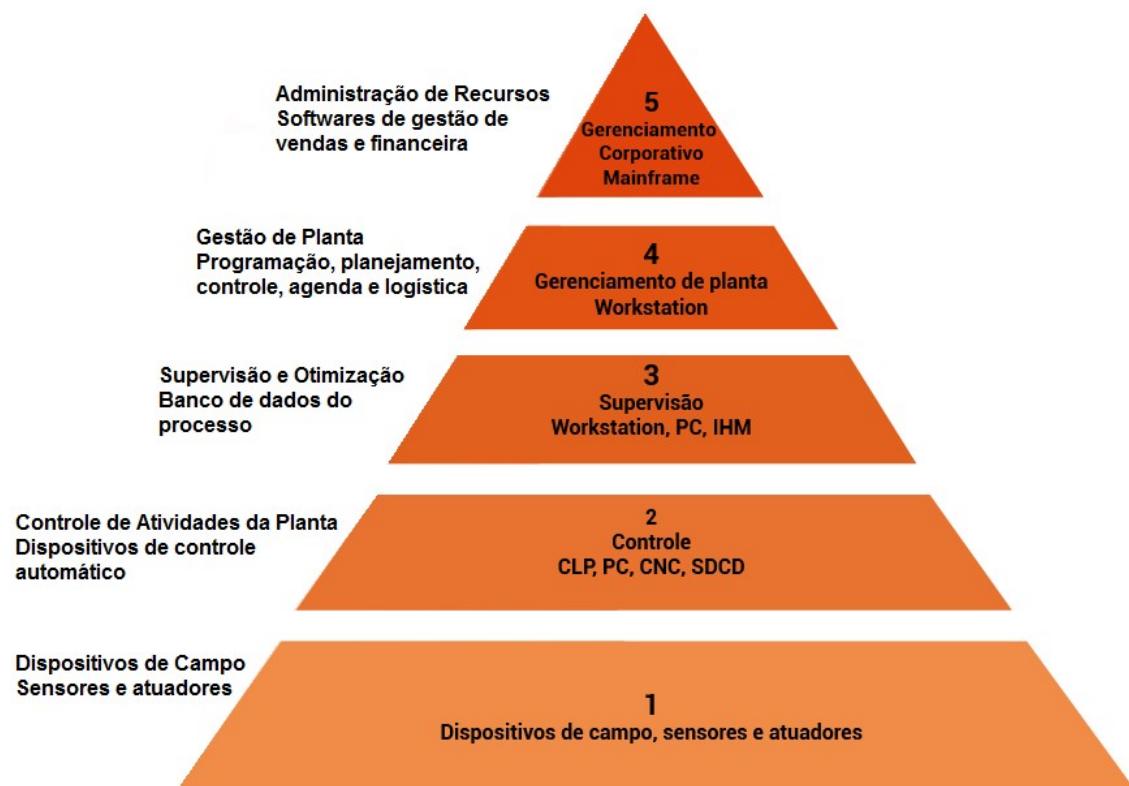
Isso é possível através da integração de vários dispositivos, controlando todo o sistema através de uma central única (ACCARDI; DODONOV, 2012). O princípio da automação residencial é utilizar os conceitos de automação industrial junto com *Internet of Things*(IoT) dentro de uma residência.

Para a implementação de um sistema de automação residencial é necessário adicionar novos dispositivos à casa. A seguir estão listados os principais componentes para a realização de tarefas e acompanhamento do sistema.

2.1.2 Componentes

De maneira geral, a automação residencial possui os mesmos tipos de componentes principais que a automação industrial, sendo eles: sensores, atuadores, barramentos ou sistemas de comunicação, controladores e supervisórios. O que difere são os tipos de equipamentos utilizados para desempenhar cada função. Os componentes utilizados estão presentes nos níveis 1, 2 e 3 da pirâmide da automação, apresentados na Figura 2.1.

Figura 2.1 – Pirâmide da automação.



Fonte: Adaptado de (ROURE, 2018).

Os elementos apresentados na Figura 2.1, podem ser descritos da seguinte maneira:

Sensores: tem a função de detectar alguma alteração no ambiente ou processo a ser controlado e converter em forma de sinal elétrico para informar ao controlador esta alteração. Um sensor acoplado a um controlador é chamado de transdutor (ROGGIA; FUENTES, 2016). Existem dois tipos de sensores: analógicos e digitais. Os sensores analógicos variam o sinal elétrico proporcionalmente ao valor da grandeza medida. Um bom exemplo são sensores de temperatura, como termistores (EPCOS, 2006) e (RAPID, 2007). Já os sensores digitais são capazes de gerar apenas dois sinais de saída: ligado ou desligado (on/off). Os sensores de presença, por exemplo, são caracterizados como sensores digitais (INJETEL, 2018) e (FORCELINE, 2018).

Atuadores: realizam alguma alteração no sistema, alterando o estado de alguma variável. Por exemplo, caso a temperatura monitorada por um sensor atinja determinado valor pode ser utilizado um ventilador ou ar condicionado como atuador (ROGGIA; FUENTES, 2016).

Sistemas de comunicação: protocolo utilizado para a comunicação de dados onde, conforme o tipo de comunicação, são encontradas diferentes taxas de transferências e tempos de resposta.

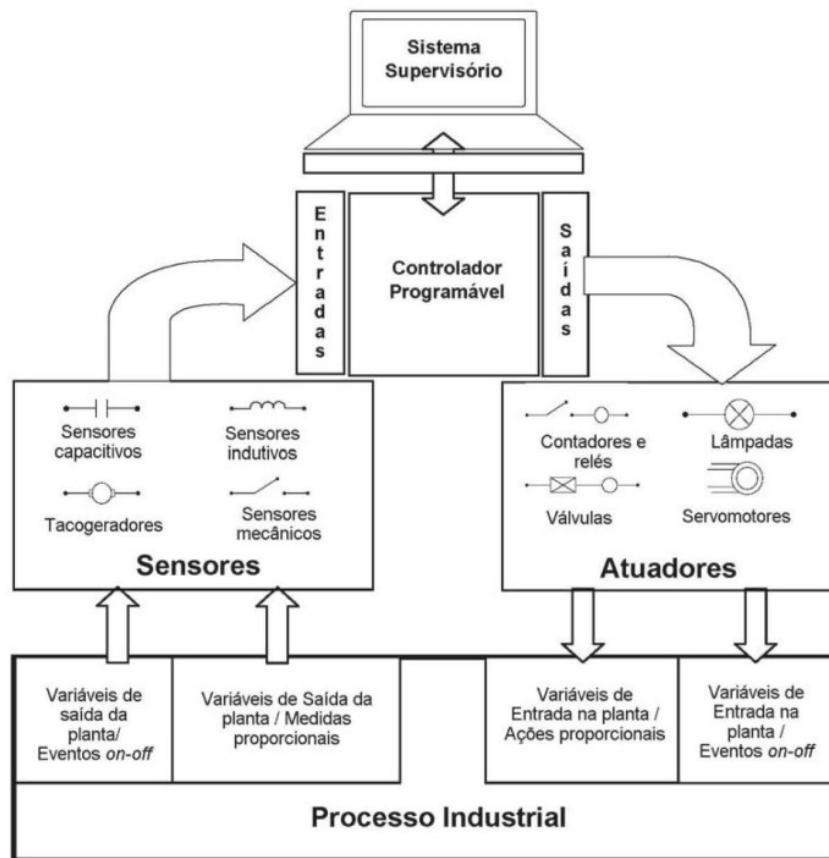
Controladores: dispositivos empregados para o recebimento de dados, processamento de dados e envio de comandos aos atuadores. O controlador é como o "cérebro"

de todo o sistema de automação. Em automação industrial são utilizados, geralmente, Controladores Lógicos Programáveis (CLPs) (MORAES; CASTRUCCI, 2010). Já em automação residencial são utilizados controladores de menores proporções, como os Arduinos.

Supervisório: também conhecido como SCADA (*Supervisory Control and Data Acquisition*), é a parte do sistema onde as informações são mostradas ao usuário. Através do sistema supervisório o usuário pode enviar comandos ao controlador e/ou monitorar o sistema. Podendo reunir informações de múltiplos controladores, sensores e atuadores em uma interface única, possibilitando o controle de dispositivos dispersos através de um só lugar (ROGGIA; FUENTES, 2016).

A Figura 2.2 mostra um exemplo de diagrama simplificado de automação contendo os níveis 1, 2 e 3 da pirâmide.

Figura 2.2 – Exemplo de diagrama de automação.

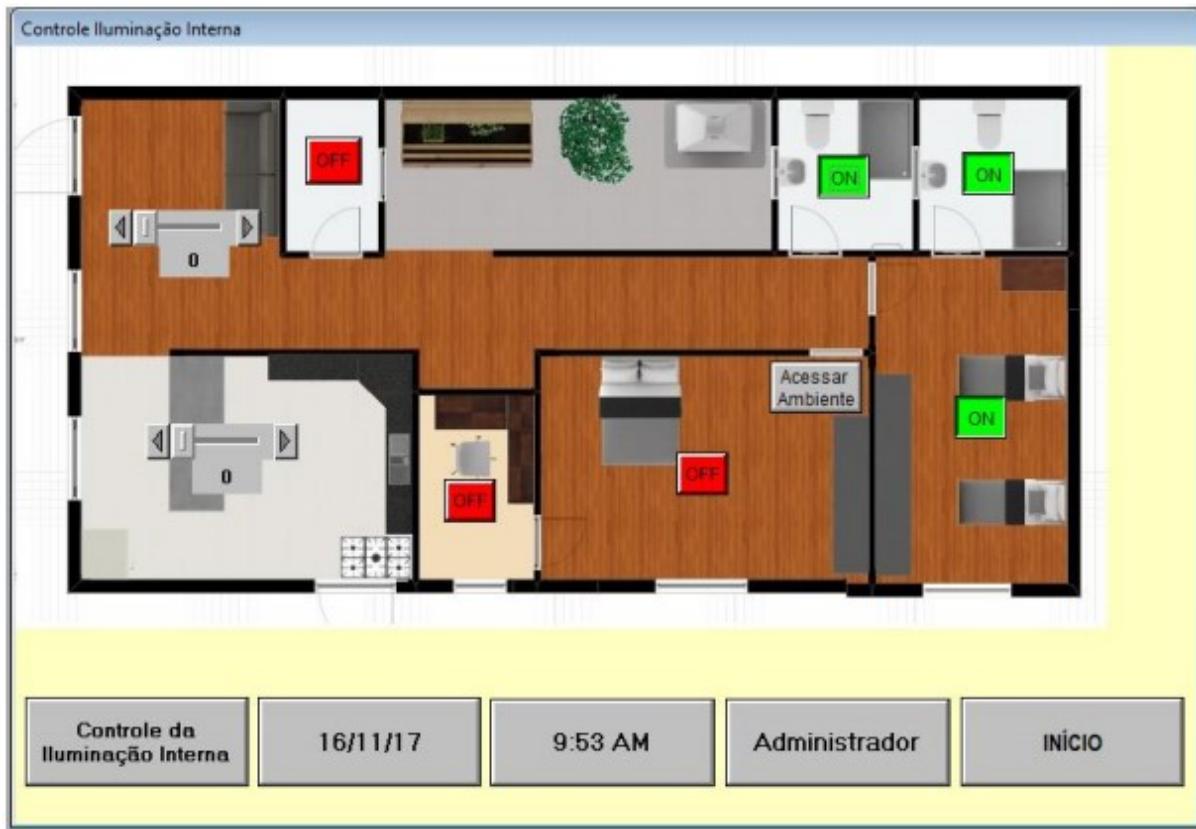


Fonte: (MORAES; CASTRUCCI, 2010).

O sistema supervisório também apresenta uma grande vantagem pela possibilidade de tornar a interface mais amigável e simples para uso, o que torna possível usuários comuns utilizarem, sem a necessidade de um estudo sobre o sistema. Por isso, ele é essencial em um projeto de automação residencial, pois será utilizado por usuários comuns,

diferentemente da indústria, onde existem técnicos especializados. A Figura 2.3 apresenta a interface de um supervisório de automação residencial para controle de luminosidade.

Figura 2.3 – Sistema supervisório residencial.



Fonte: (PELLENZ, 2017).

2.2 INTERNET OF THINGS

2.2.1 História

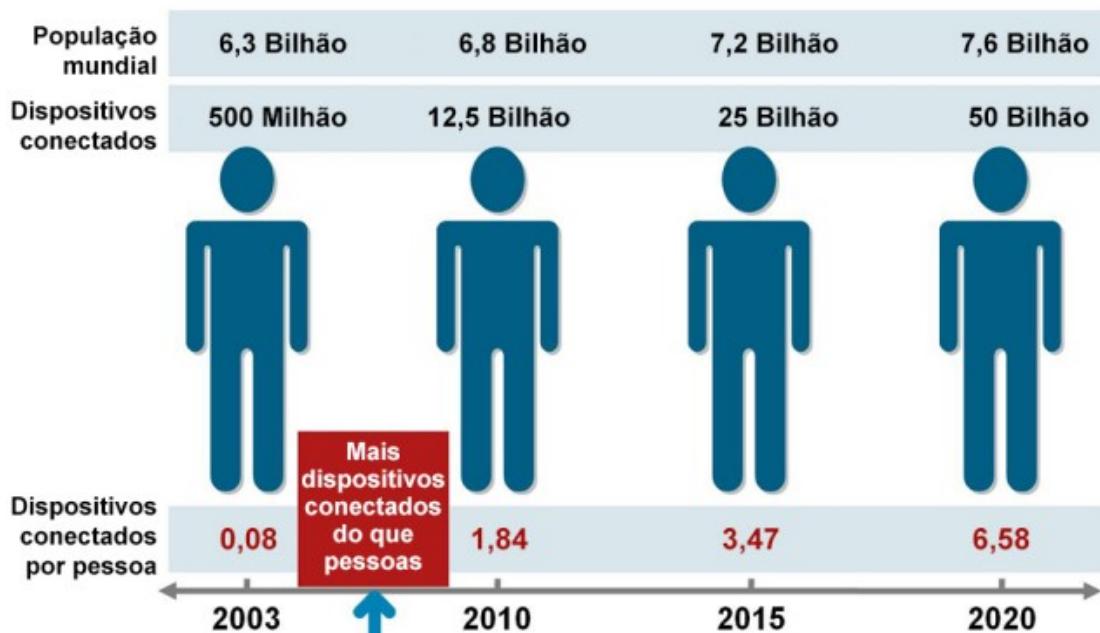
A internet das coisas surgiu após o avanço de áreas, como a microeletrônica, comunicação, sensoriamento e sistemas embarcados. Porém, não é um termo tão novo, visto que surgiu no ano de 1999, no *Massachusetts Institute of Technology* (MIT). Kevin Ashton apresentou pela primeira vez o conceito na literatura, afirmando que se todos os objetos do dia a dia possuíssem algum tipo de conexão de dados, eles poderiam se comunicar e ser gerenciados por computadores (MADAKAM; RAMASWAMY; TRIPATHI, 2015) e (SURESH et al., 2014).

Ashton apresentou uma visão muito inovadora para uma época que possuía tecnologias muito limitadas. Porém, atualmente, esse não é um conceito único, com o passar

do tempo e avanço de tecnologias, surgiram novos conceitos baseados na ideia de Ashton.

A popularização do IoT aconteceu por volta dos anos de 2008 e 2009, onde começou a aumentar de maneira muito significativa o número de dispositivos conectados à rede, ultrapassando o número da população mundial (EVANS, 2011). A Figura 2.4 ilustra esse processo.

Figura 2.4 – Relação entre a população e dispositivos conectados à internet.



Fonte: (EVANS, 2011).

Com os avanços da tecnologia e um início da popularização da IoT, já é possível encontrarmos dispositivos cotidianos com conexão à internet, como aparelhos de ar condicionado e geladeiras. A geladeira por exemplo, não pode ser controlada via computador, mas pode fornecer dados como a validade dos alimentos presentes nela e emitir notificações ao se aproximar da data, assim ampliando o conceito de IoT criado por Ashton (FREIRE, 2018).

2.2.2 Conceito

A internet das coisas pode ser explicado de maneira simples como a conexão entre pessoas, computadores e coisas. Sendo as coisas qualquer objeto, mesmo que não controlável, mas monitorável através de sensores para o conhecimento de seu estado, como ligado ou desligado. Também dispositivos simples que não executam processamento de dados, que simplesmente executam ações, como relés telecomandados utilizados no acionamento de lâmpadas (SURESH et al., 2014).

Nem todos os dispositivos conectados à internet são considerados parte da IoT. O termo é mais utilizado para dispositivos ”incomuns” conectados à rede. Computadores e celulares, por exemplo, não são considerados IoT (RANGER, S., 2018).

A partir dos conceitos apresentados é possível notar uma possível ligação entre IoT e a automação. A integração dos dois sistemas pode trazer um grande benefício aliando o telecomando e a aquisição de dados do IoT junto a automação, que pode armazenar, gerenciar e interpretar esses dados para realizar ações sem a necessidade de intervenção do usuário.

2.3 SOLUÇÕES COMERCIAIS EM AUTOMAÇÃO RESIDENCIAL

2.3.1 SONOFF

O SONOFF é um dispositivo de IoT fabricado pela ITEAD Intelligent Systems. A função do dispositivo é o acionamento e desligamento de cargas via *Wireless* (ITEAD INTELLIGENT SYSTEMS, 2016b). A Figura 2.5 apresenta o SONOFF básico.

Figura 2.5 – Relé SONOFF básico.



Fonte: (ITEAD INTELLIGENT SYSTEMS, 2016b).

O dispositivo apresenta as seguintes especificações (ITEAD INTELLIGENT SYSTEMS, 2016b):

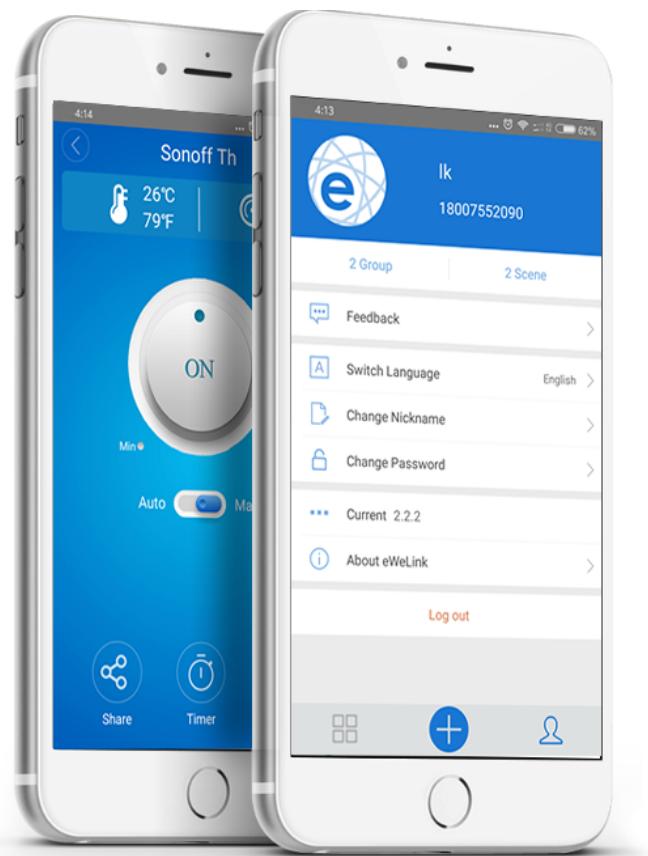
- Tensão de alimentação: 90V-250V AC.
- Corrente máxima de carga: 10A.
- Conexão Wireless: Wi-Fi 2.4GHz b/g/n.
- Mecanismos de segurança: WEP/WPA-PSK/WPA2-PSK.

- Temperatura ambiente de operação: 0°C-40°C.
- Umidade ambiente de operação: 5%-95%.

Há também modelos com algumas funcionalidades extras, como acionamento via controle remoto, monitoramento de umidade e temperatura e múltiplos canais. Em modelos com monitoramento de temperatura e umidade é possível definir ações automáticas conforme os valores de leitura dessas variáveis no ambiente (ITEAD INTELLIGENT SYSTEMS, 2016c).

O principal modo de controle do dispositivo é utilizando o aplicativo eWeLink, disponibilizado gratuitamente para os smartphones de sistema operacional iOS e Android (ITEAD INTELLIGENT SYSTEMS, 2016a). A Figura 2.6 apresenta a interface do aplicativo.

Figura 2.6 – Aplicativo eWeLink.



Fonte: (ITEAD INTELLIGENT SYSTEMS, 2016a).

Além do comando via aplicativo é possível realizar o comando através de um botão presente no dispositivo.

2.3.2 Broadlink

A Broadlink apresenta soluções em automação residencial nas áreas de conforto e segurança. Para a área de conforto os principais itens a se destacar são as tomadas inteligentes, interruptores inteligentes e os controles universais.

2.3.2.1 Tomadas Inteligentes

As tomadas inteligentes são dispositivos conectados às tomadas convencionais da residência e possuem uma tomada para a conexão do equipamento a ser ligado a partir da mesma. Com essas é possível realizar acionamentos remotos ou agendados (BROADLINK TECHNOLOGY, 2018b). A Figura 2.7 apresenta as tomadas inteligentes Broadlink.

Figura 2.7 – Tomadas inteligentes Broadlink.



Fonte: (BROADLINK TECHNOLOGY, 2018b).

As tomadas inteligentes são acionadas unicamente através do aplicativo para *smartphone*, pode também ser agendado o acionamento.

2.3.2.2 Interruptores Inteligentes

Os interruptores inteligentes são dispositivos que possuem múltiplas maneiras de serem acionados. Podem ser acionados através de um toque no dispositivo, utilizando rádio frequência através de um controle remoto universal ou através do aplicativo para *smartphone* (BROADLINK TECHNOLOGY, 2018c). A Figura 2.8 apresenta os inter-

ruptores inteligentes Broadlink.

Figura 2.8 – Interruptores inteligentes Broadlink.



Fonte: (BROADLINK TECHNOLOGY, 2018c).

2.3.2.3 Controles Universais

Os controles universais são dispositivos destinados a comandar diferentes equipamentos presentes no ambiente a partir de um único lugar. Há modelos com suporte a LED's infravermelhos (IR) e a rádio frequência (RF) (BROADLINK TECHNOLOGY, 2018d). A Figura 2.9 apresenta o módulo emissor do controle remoto universal Broadlink.

Figura 2.9 – Controle remoto universal Broadlink.

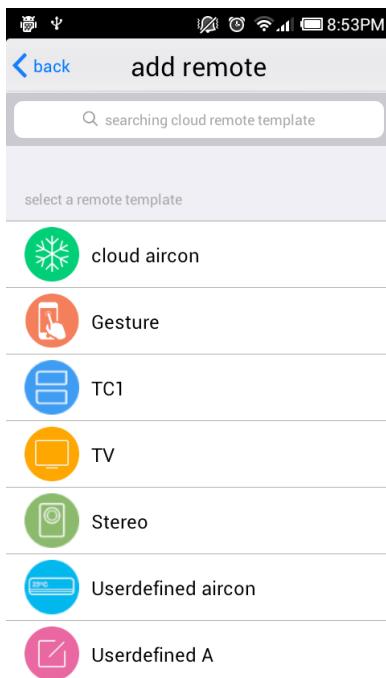


Fonte: (BROADLINK TECHNOLOGY, 2018d).

2.3.2.4 Aplicativo

A partir do aplicativo e-Control, fornecido pela BroadLink, para *smartphones* é possível realizar o comando dos dispositivos e o agendamento de acionamentos. Caso sejam utilizados sensores da BroadLink, estes podem ser integrados ao aplicativo e podem ser agendados acionamentos conforme eventos detectados pelos sensores (BROADLINK TECHNOLOGY, 2018a). A Figura 2.10 apresenta a interface do aplicativo.

Figura 2.10 – Aplicativo e-Control.



Fonte: (GOOGLE PLAY, 2019).

2.3.3 Fibaro

A Fibaro apresenta diversas soluções em automação residencial, com um destaque para a tecnologia *Wireless Z-Wave communication*. Essa tecnologia possibilita uma maior área para os dispositivos se conectarem sem fio, pois, ao invés de todos os dispositivos se conectarem a um roteador central, eles podem se conectar uns aos outros como se cada um atuasse como um repetidor (FIBAR GROUP, 2019b).

As principais soluções disponibilizadas pela Fibaro são voltadas aos acionamentos e monitoramento do consumo de energia.

2.3.3.1 Wall Plug

A tomada inteligente da Fibaro possui como principais funcionalidades: controle a distância, monitoramento de consumo e proteção contra sobrecarga (FIBAR GROUP, 2019h). A Figura 2.11 apresenta alguns modelos da tomada Wall Plug.

Figura 2.11 – Tomada inteligente Wall Plug.



Fonte: (FIBAR GROUP, 2019h).

São encontradas em modelos com diferentes níveis de potência de saída, de 70W a 1800W, conforme a cor do LED ao redor da tomada. Apresentam também uma porta USB para o carregamento de dispositivos.

As tomadas possuem conexão via *wireless*, podendo ser gerenciadas a partir de celulares, tablets, computadores ou notebooks. É possível acionar ou desligá-la, visualizar o consumo do dispositivo em tempo real e ver relatórios do consumo durante determinados períodos de tempo.

2.3.3.2 Single e Double Switch

Tratam-se de relés telecomandados, com funcionalidades similares as Wall Plugs, porém estas são instaladas juntamente as tomadas e/ou interruptores (FIBAR GROUP, 2019g). A Figura 2.12 apresenta o Single Switch e o Double Switch, a diferença entre eles é que um possui somente um canal e outro possui dois canais, respectivamente.

Figura 2.12 – Single e Double Switch.



Fonte: (FIBAR GROUP, 2019g).

Além das funcionalidades já apresentadas no Wall Plug, estes dispositivos apresentam também a possibilidade de serem integrados a sensores desenvolvidos pela Fibaro para, por exemplo, se ter as luzes acesas ao entrar na sala, se utilizados em conjunto a sensores de presença.

2.3.3.3 Dimmer 2

É um dispositivo com a finalidade de controle de nível de iluminação (FIBAR GROUP, 2019a). A Figura 2.13 apresenta o dispositivo Dimmer 2.

Figura 2.13 – Dimmer 2.



Fonte: (FIBAR GROUP, 2019a).

Além do controle de iluminação, o dispositivo também é capaz de monitorar o consumo energético.

2.3.3.4 Roller Shutter 3

É um dispositivo destinado ao controle de cortinas e portões (FIBAR GROUP, 2019e). A Figura 2.14 apresenta o dispositivo.

Figura 2.14 – Roller Shutter 3.



Fonte: (FIBAR GROUP, 2019e).

O motor vai conectado diretamente ao dispositivo, assim é possível o uso de um sistema de segurança, caso algo ocasione o travamento do portão/cortina. Permite o acionamento remoto ou via botões que podem ser conectados ao dispositivo (FIBAR GROUP, 2019e).

2.3.3.5 Home Center 2

É o dispositivo central de controle. Os atuadores e sensores são integrados a este dispositivo para que nele sejam programados os acionamentos e criados bancos de dados (FIBAR GROUP, 2019c). A Figura 2.15 apresenta o Home Center 2.

Figura 2.15 – Home Center 2.



Fonte: (FIBAR GROUP, 2019c).

2.3.3.6 Aplicativos

A Fibaro disponibiliza aplicativos para *smartphones*, Android e iOS, e para computadores (FIBAR GROUP, 2019f). A Figura 2.16 e a Figura 2.17 apresentam os aplicativos.

Figura 2.16 – App Fibaro para computadores.



Fonte: (FIBAR GROUP, 2019d).

Figura 2.17 – App Fibaro para *smartphones*.



Fonte: (APPLE, 2019).

2.4 SISTEMAS SUPERVISÓRIOS

2.4.1 História

Inicialmente, na automação industrial, eram encontradas as Interfaces Homem Máquina (IHMs) que tinham a função de monitorar e controlar os processos da planta industrial. São sistemas robustos, usados em chão de fábrica, e seu uso era bem específico e limitado, porém tinham um custo relativamente baixo (MORAES; CASTRUCCI, 2010).

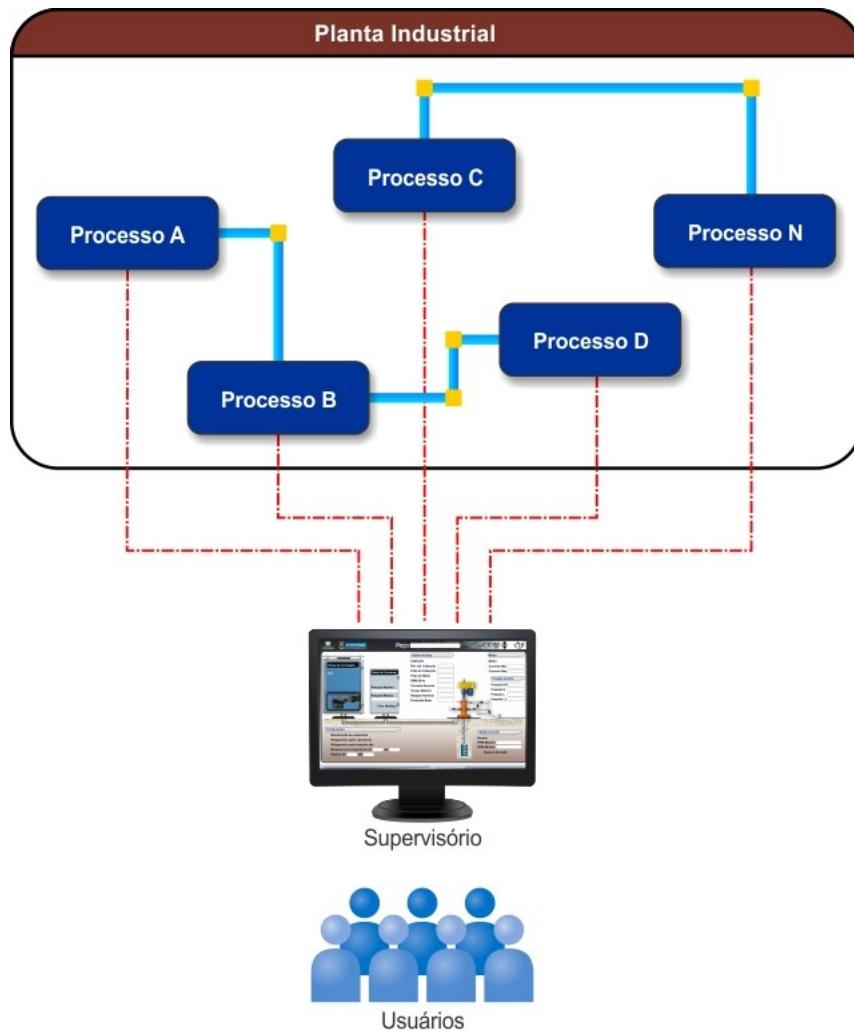
Com a evolução das tecnologias e a redução do custo de computadores, surgiram os sistemas supervisórios em computadores. Possuindo capacidade de processamento muito maior, tornou-se possível controlar uma quantidade muito maior de controladores, gerenciar mais variáveis na planta e gerar bancos de dados com informações da planta ao longo do tempo (JURIZATO; PEREIRA, 2003).

2.4.2 Conceito

Como o nome indica, sistemas supervisórios não fornecem um controle completo, mas focam na supervisão e envio de comandos. Sistemas supervisórios possuem uma gama muito grande de canais de entrada e saída (I/O), podendo variar entre 1000 e 10 mil canais.

O supervisório recebe as informações dos controladores e mostra ao usuário, e também recebe informações do usuário e as envia para os controladores (DANEELS; SALTER, 1999). A Figura 2.18 demonstra, de maneira simplificada, um sistema supervisório.

Figura 2.18 – Representação simplificada de sistema supervisório.



Fonte: (HI Tecnologia, 2018).

Por trás da interface de um sistema supervisório existem alguns elementos e recursos presentes na maioria dos sistemas.

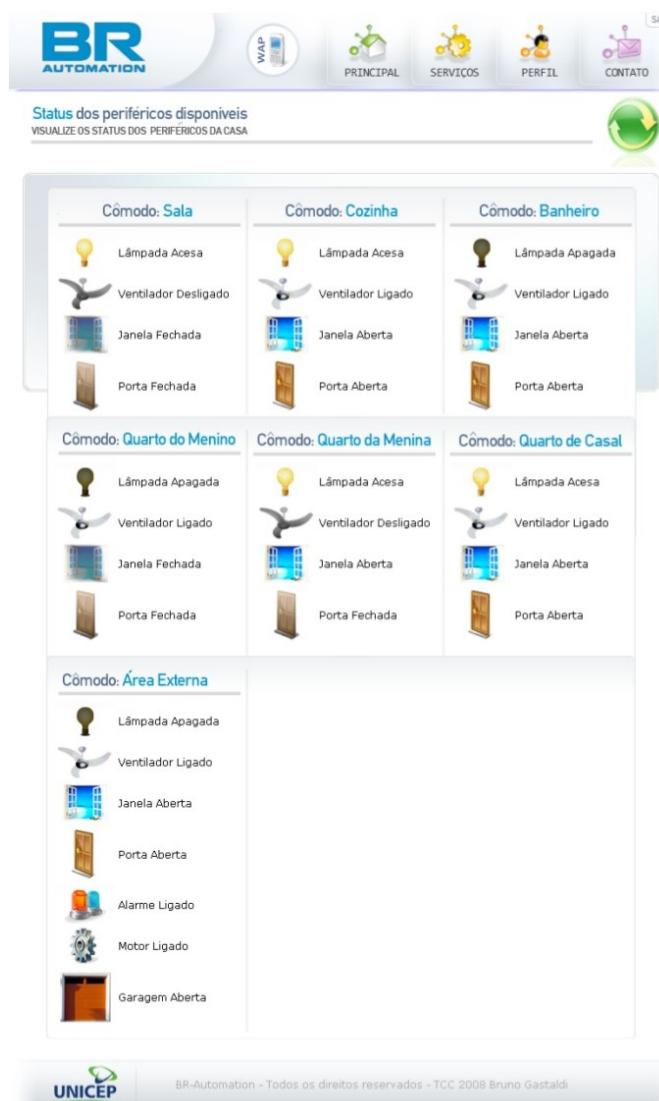
2.4.3 Componentes

Os principais elementos de um sistema supervisório são (AUGUSTO, 2015):

Tags: variáveis do sistema, internas (operações realizadas dentro do supervisório) ou externas (estado de sensores ou atuadores).

Telas sinóticas: telas que apresentam o sistema e valores das *tags* ao usuário. É um dos recursos mais importantes do supervisório. É totalmente personalizável, podendo demonstrar, de maneira mais destacada, as partes mais importantes da planta. Torna mais simples e fácil o controle da planta com a adição de figuras representando os componentes da planta, por exemplo, para que usuários, com um menor conhecimento sobre a automação, consigam controlar o processo. A Figura 2.19 demonstra um exemplo de sistema supervisório com figuras ilustrativas que representam os dispositivos gerenciados.

Figura 2.19 – Exemplo de tela sinótica.



Fonte: (GASTALDI, 2008).

Alarms e eventos: normalmente são ligados a alguma *tag* representando um aviso para alguma situação fora do comum ou até mesmo perigosa. Por exemplo, na indústria, caso um forno ultrapasse a temperatura desejada, ou uma caldeira apresente uma pressão muito elevada.

Históricos: armazenam valores de *tags* para que o usuário possa analisar posteriormente. Registra valores e horários.

Relatórios: ferramenta responsável por formatar a visualização das informações armazenadas nos bancos de dados no sistema. Podem formar gráficos ou tabelas para facilitar a visualização e interpretação dos dados armazenados.

2.4.4 Vantagens

A partir dos componentes citados, é possível destacar algumas vantagens (MOTT, 2012):

Análise de tendências: a partir dos relatórios gerados pelo supervisório é possível analisá-los e realizar ações proativas para melhorar o desempenho da planta.

Alarms: a partir dessa funcionalidade é possível perceber falhas ou problemas em tempo real, podendo ser feita através de sinais sonoros para uma notificação mais efetiva.

Operação remota: é possível monitorar e realizar ações a distância em uma sala de controle, sendo uma grande vantagem para uma planta onde há sensores e atuadores instalados em uma grande área.

Aumentar a eficiência da planta: como é possível ter total controle de uma grande planta em tempo real, é possível torná-la mais eficiente. O operador consegue supervisionar a planta toda a partir de um único lugar, conseguindo detectar alterações em qualquer parte da mesma e corrigi-las mais rapidamente.

2.4.5 Softwares

Para a aplicação de um sistema supervisório com controle a longa distância são necessários um servidor e um dispositivo para controle conectados à internet. Através do conhecimento em programação e desenvolvimento de páginas web, é possível utilizar computadores e celulares conectados à internet para desempenhar essas funções (GASTALDI, 2008). No entanto, hoje no mercado já existem *softwares* prontos para desempenhar essas funções específicas, tornando mais fácil o desenvolvimento desses sistemas (MOTT, 2012). Dentre esses *softwares* é possível destacar:

2.4.5.1 ScadaBR

Este é uma plataforma para desenvolvimento de sistemas supervisórios que roda diretamente em interface web a partir do navegador, o que traz uma facilidade para acesso a partir de outros dispositivos. Este *software* é *open source* e baseado em Java, ou seja, totalmente livre e editável, o que o torna interessante para aplicações de pequeno porte, devido ao alto custo de outros *softwares*. Suas principais funcionalidades são as seguintes (SENSORWEB, 2018):

- *Engine* de Alarmes e Eventos parametrizáveis;
- Alarmes Sonoros;
- Sistema de Permissões por Usuários;
- Relatórios de Variáveis, Alarmes e Eventos;
- *Engine* de *scripts* para controle, automação e *batches*, entre outros;
- Ambiente de Desenvolvimento para Aplicações SCADA;
- Acesso em *Browser* (*Desktop*, *mobile*);
- API com exemplos em Java, *Visual Basic*, PHP e *Python*, entre outros;
- Aquisição de dados em mais de 20 protocolos como: Modbus TCP/IP e Serial, OPC, DNP3, IEC, Serial ASCII, HTTP;
- Variáveis calculadas (com funções matemáticas, estatísticas, e lógica em geral)
- Visualização de dados em tempo real;
- Registro contínuo das variáveis em Banco de Dados (*Historian/ Datalogger*);
- Construção de telas gráficas (HMI ou Sinópticos).

É utilizado também em grandes aplicações, como sistemas de automação industrial, de geração, transmissão e distribuição de energia elétrica, entre outras aplicações. Devido a sua capacidade de comunicação em diversos protocolos, possibilita a integração com grande parte dos controladores disponíveis no mercado (SENSORWEB, 2018). A Figura 2.20 demonstra uma tela gráfica que apresenta dados obtidos com o ScadaBR.

Figura 2.20 – Tela gráfica no ScadaBR.



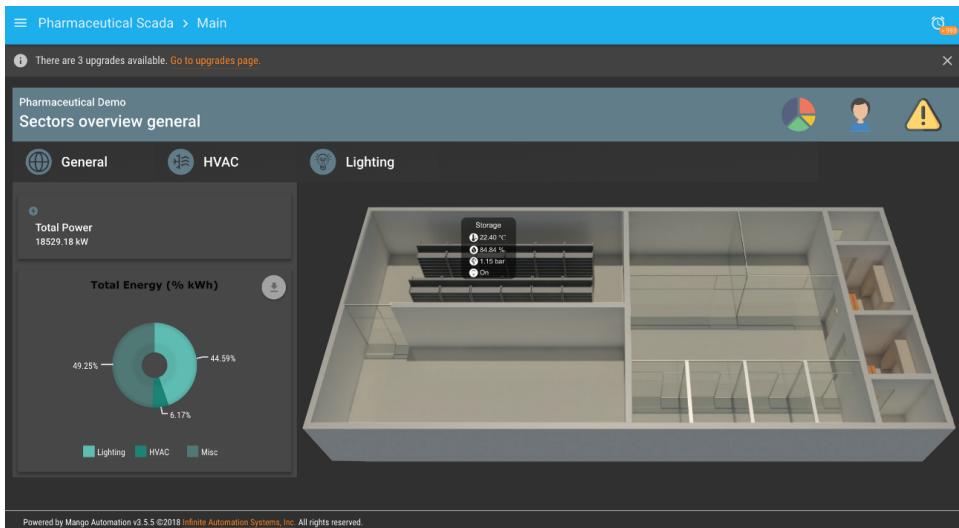
Fonte: (BASTOS, 2014).

2.4.5.2 Mango Automation

O software *Mango Automation* serviu como base para a criação do software ScadaBR (SENSORWEB, 2018), porém com o passar dos anos o ScadaBR parou de receber atualizações oficiais.

O *Mango Automation* engloba as funções presentes no ScadaBR e, como continua recebendo atualizações, apresenta novas funcionalidades e uma melhor interface gráfica. A Figura 2.21 apresenta um exemplo de sistema supervisório desenvolvido e gerenciado no *Mango Automation*.

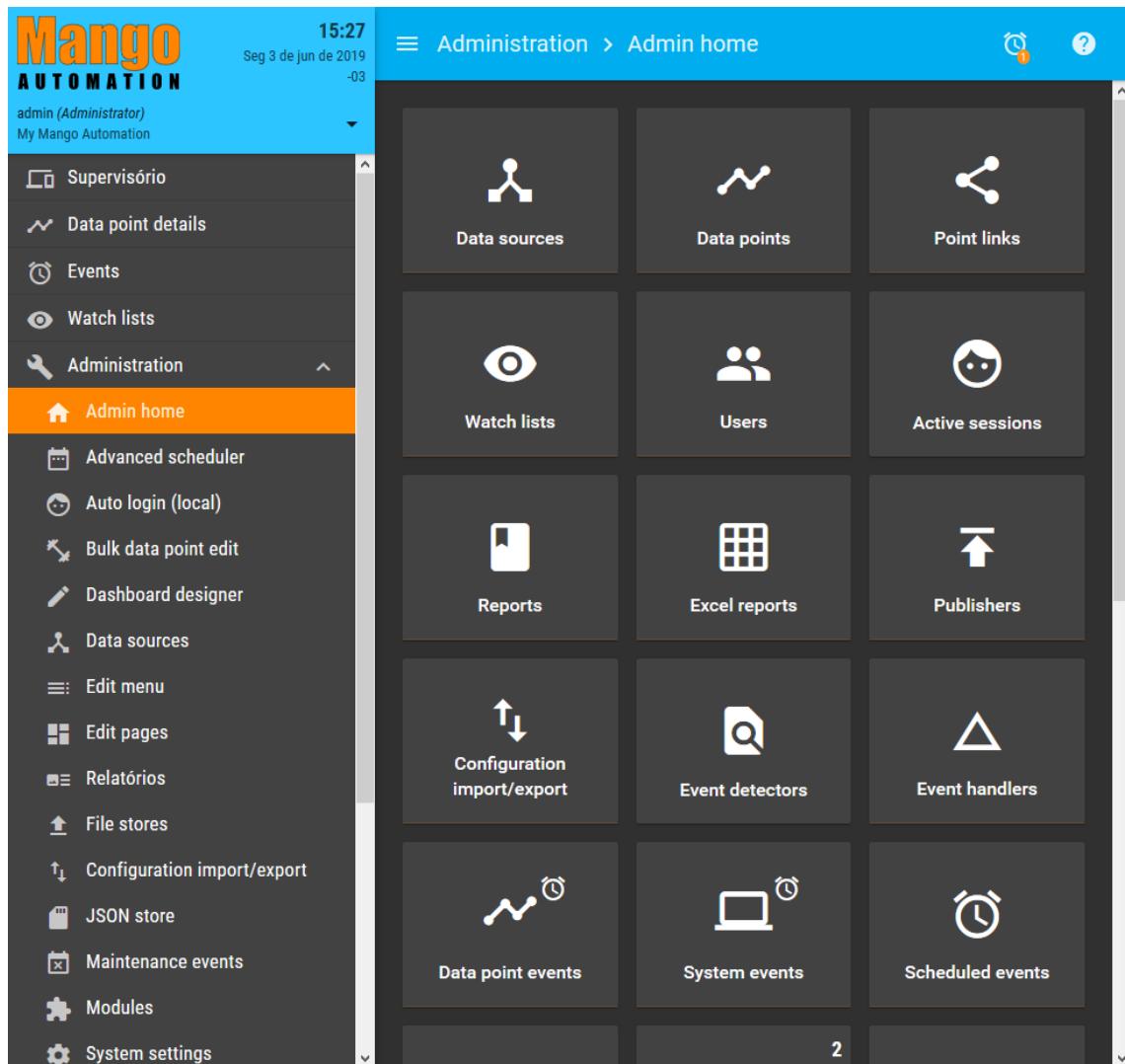
Figura 2.21 – Sistema supervisório no *Mango Automation*.



Fonte: (GUETTE, 2018).

Atualmente grande parte das configurações do *Mango Automation* encontram-se em um novo formato de interface gráfica chamada *User Interface* (UI) apresentado na Figura 2.22.

Figura 2.22 – Tela inicial *Mango Automation*.



Fonte: Autor.

No entanto, parte das configurações ainda encontram-se no formato apresentado pelo ScadaBR, chamado de modo *Legacy*. A Figura 2.23 apresenta uma interface de configuração do *Mango Automation* semelhante ao ScadaBR.

Figura 2.23 – Configuração de *Data sources*.

Name	Type	Connection
Arduino - Estação	Modbus I/P	192.168.0.108:502
Arduino_Dimmer	Modbus I/P	192.168.0.110:502
ESP01 - Ar Condicionado	Modbus I/P	192.168.0.100:502
ESP01 - Lampada LDR	Modbus I/P	192.168.0.102:502
ESP01 - Lampada ON/OFF	Modbus I/P	192.168.0.101:502
NodeMCU - RFID	Modbus I/P	192.168.0.104:502

This instance is running in free mode. Limited to 300 data points and non-commercial usage.
©2018 Infinite Automation Systems Inc. All rights reserved
www.InfiniteAutomation.com

Fonte: Autor.

Como principais características do *Mango Automation* é possível destacar:

- Totalmente utilizado via *browser*.
- Grande variedade de protocolos incorporados.
- Históricos em séries temporais de alta performance
- Telas de análises incorporadas.
- Alertas via e-mail.
- Alarmes
- Utilização de scripts para algoritmos de controle e automação.
- Possibilidade de instalação de módulos extras.

O *software* possui diferentes níveis de licença para utilização. A licença gratuita é limitada por: máximo de 300 data points, não suporta sincronização com Mango Cloud, uso comercial proibido e não é possível criar bancos de dados em séries temporais de alta performance (INFINITE AUTOMATION, 2019).

2.4.5.3 Elipse

A empresa Elipse possui múltiplos *softwares* voltados a sistemas supervisórios para automação. O Elipse E3 é uma plataforma completa para desenvolvimento e utilização de sistema supervisório. É um *software* que requer aquisição de licença, porém possui uma versão de uso livre, com algumas limitações. As principais características da versão completa são as seguintes (ELIPSE, 2015a):

- Multusuários e multiprojetos: Permite editar e executar diversos projetos simultaneamente;
- Redundância nativa com sincronismo de dados históricos e alarmes;
- Bibliotecas de objetos gráficos e estruturas de dados reutilizáveis;
- Editor de telas completo e poderoso;
- Conexão nativa transparente entre servidores remotos;
- Segurança e compactação na transmissão de dados;
- Fácil gerenciamento da aplicação;
- Grande flexibilidade na gestão de alarmes e eventos;
- Poderosa ferramenta de scripts;
- Acesso nativo a bancos de dados comerciais;
- Ferramenta de logs, consultas e relatórios integrada;
- Alta segurança e rastreabilidade de acordo com a norma FDA CFR 21 Part 11;
- OPC *Classic* e UA;
- Integração com o *Windows Active Directory*.

Com a versão de uso livre é possível criar um protótipo e testar sua funcionalidade. No entanto, com essa versão não é possível uma implementação real em uma residência. As principais limitações a serem destacadas, que impedem o uso para automação residencial, é o limite máximo de *tags*, que são 20, e que o uso é permitido por duas horas contínuas (ELIPSE, 2015a). Sua versão completa possui funcionalidades similares aos outros *softwares*, porém possui interfaces gráficas mais avançadas e mais completas, quando comparado ao ScadaBR, como é possível ver na Figura 2.24.

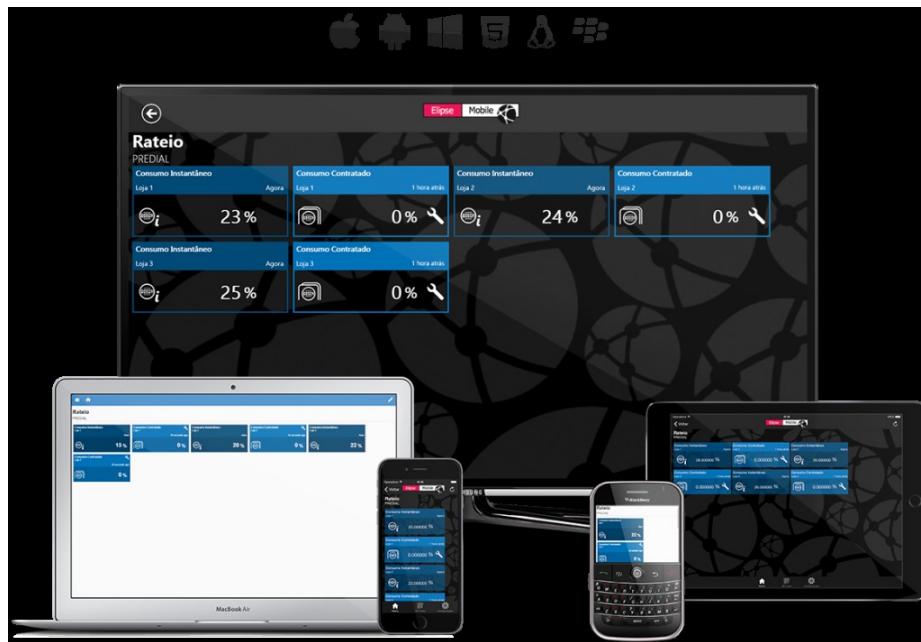
Figura 2.24 – Interface gráfica Elipse.



Fonte: (ELIPSE, 2015a).

A Elipse possui a plataforma Elipse Mobile, que consiste em um app gratuito desenvolvido para acesso ao sistema supervisório a partir de dispositivos móveis, como celulares e tablets. Isso traz mais uma facilidade para o acesso ao sistema em um caso de automação residencial, podendo monitorar o sistema e enviar comandos ao mesmo (ELIPSE, 2015b). A Figura 2.25 mostra como é possível utilizar o Elipse Mobile em variados dispositivos.

Figura 2.25 – Elipse Mobile em diversas plataformas.



Fonte: (ELIPSE, 2015b).

Além desses destacados, existem no mercado inúmeros outros *softwares* voltados a sistemas supervisórios, porém *softwares* que requerem a aquisição de licença e com funcionalidades muito parecidas aos já citados. Os principais são (MOTT, 2012):

- FactoryTalk View SE da Rockwell Automation;
- iFIX da General Electric;
- InduSoft Web Studio da InduSoft;
- ProcessView da SMAR;
- SIMATIC Wincc da Siemens;
- Vijeo Citect da Schneider Electric;
- Wonderware inTouch da Invensys.

2.4.6 Comparativo *softwares*

A plataforma Elipse apresenta uma solução completa para o desenvolvimento e utilização de sistemas supervisórios, tanto a nível industrial quanto a nível residencial. Para a automação residencial, um ponto importante a ser destacado é o aplicativo para celulares que o desenvolvedor disponibiliza, tornando mais simples a sua utilização. O *software* também disponibiliza mais opções gráficas. A sua versão de uso livre possibilita um número limitado no uso de *tags*, o que restringe o tamanho máximo do projeto. Outro fator limitante é o tempo de uso que é de somente duas horas, o que inviabiliza o seu uso remoto, pois o usuário não terá a disponibilidade de reiniciar o sistema a cada duas horas.

O *software* ScadaBR pode ser dito como um similar ao Elipse, um pouco mais simples, porém tão funcional quanto. Ele executa sua interface a partir do navegador. Também é possível o acesso remoto, porém esse envolve configurações de rede mais complexas. E por ser um *software* de uso totalmente livre, reduz o custo de instalação do sistema.

O *software* Mango Automation apresenta uma boa relação entre funcionalidades e interfaces gráficas. Possui diversas opções de customizações que tornam o seu uso mais simples e intuitivo, facilitando a utilização pelo usuário final.

2.5 CONCLUSÕES PARCIAIS

Conectados a um sistema supervisório, os dispositivos IoT possuem uma melhor usabilidade pelo usuário, podendo também serem utilizados como dispositivos de campo

em um sistema de automação. A partir do sistema supervisório, o usuário pode usufruir de todos os dispositivos a partir de um único local. Neste contexto, o *software Mango Automation* mostra-se como uma opção mais atrativa, por ser um *software* gratuito, para o uso com até 300 dispositivos/variáveis, sem limite de tempo de utilização e com grandes possibilidades de customizações na interface gráfica.

3 SISTEMA DESENVOLVIDO

Nesse capítulo, primeiramente, é apresentada uma planta residencial. A partir da mesma é desenvolvido um sistema de automação residencial e um sistema supervisório que atenda o sistema de automação proposto. É apresentado uma lista de dispositivos que são utilizados na implementação da automação residencial, e em seguida, o processo para a implementação do sistema supervisório, a partir do *software Mango Automation*. Ainda é apresentado algumas funcionalidades pertinentes para o uso em automação residencial.

3.1 PLANTA

3.1.1 Planta Inicial

A residência automatizada trata-se de um pequeno apartamento, ou uma kitnet. É composta por uma cozinha, uma sala de estar/trabalho, um quarto e um banheiro. A Figura 3.1 apresenta a planta utilizada para este trabalho.

Figura 3.1 – Planta da residência.



Fonte: (AMAZING ARCHTECTURE, 2017)

Considerando uma residência comum, os equipamentos geralmente encontrados e interessantes para a automação são lâmpadas e aparelhos de ar condicionado. Os eletrodomésticos e demais aparelhos da casa não são considerados nesta aplicação.

3.1.2 Soluções implementadas

Buscando maior comodidade, segurança e economia são implantados alguns equipamentos novos nessa residência.

Na porta de acesso é colocada um sistema de fechadura eletrônica. Desse modo é possível abri-la através de chaveiros ou cartões de identificação por radiofrequência (RFID). Assim é possível monitorar quando e quantas vezes a porta foi aberta, e ainda se algum cartão ou chaveiro inválido foi utilizado.

Nas lâmpadas pode ser feito o controle on/off ou o controle através de dimmer, que torna possível o controle do nível de luminosidade desejado.

O ar condicionado é controlado através de um controle on/off, onde seu acionamento é dado a partir da temperatura do ambiente onde o mesmo se encontra.

No entanto, para essas aplicações é necessário a adição de novos dispositivos na residência.

3.2 DISPOSITIVOS PARA AUTOMAÇÃO RESIDENCIAL

3.2.1 Arduino

As placas Arduino são controladores lógicos programáveis *open source*. Existe no mercado uma variedade de modelos com diferentes especificações de processador e tipos de entradas. Os Arduinos são programados em linguagem C++. Conforme o modelo, possuem entradas/saídas digitais, analógicas e com modulação por largura de pulso (PWM), além de contarem com diversas expansões e dispositivos periféricos acopláveis que expandem as possibilidades de uso (ARDUINO, 2019). A Figura 3.2 apresenta o Arduino UNO, um dos modelos mais simples e um dos mais utilizados.

Figura 3.2 – Arduino UNO.



Fonte: (ARDUINO, 2018)

3.2.2 Módulo Ethernet W5100

Este módulo é capaz de conectar o Arduino a internet através de cabo com conectores do tipo RJ-45. Possui também entrada para cartão de memória do tipo SD, que possibilita a criação de páginas web para controle do Arduino e exibição de dados (WIZNET, 2008). A Figura 3.3 apresenta o módulo *ethernet*.

Figura 3.3 – Módulo Ethernet W5100.



Fonte: (THOMSEN, 2014)

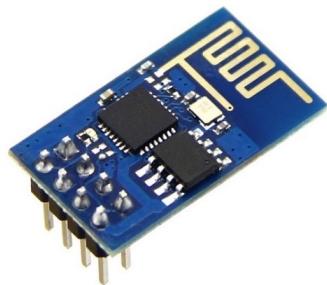
3.2.3 Módulos Wi-Fi ESP8266

Essa é uma família de módulos que são capazes de trabalharem como controladores lógicos independentes. Além de atuarem de forma autônoma, podem ser utilizados acoplados a outros controladores, como os Arduinos, para realizar a comunicação via Wi-Fi. A família é composta por vários dispositivos de mesma função, porém com características diferentes, como o número de entradas/saídas e possibilidade de inserção de antena externa para maior alcance. Estes módulos apresentam GPIOs (*General Purpose Inputs-Outputs*). Estas podem operar como entradas digitais, que podem ser utilizadas como detectores de interrupção externa, e como saídas digitais, que podem ser utilizadas também com modulação PWM (ESPRESSIF IOT TEAM, 2018).

3.2.3.1 ESP-01

O módulo mais simples da família é o ESP-01, que conta apenas com quatro GPIOs, podendo ser aplicado para acionamentos remotos e leituras de sensores digitais. Esse é apresentado na Figura 3.4.

Figura 3.4 – Módulo ESP-01.



Fonte: (FILIPEFLOP, 2018b)

Este módulo não apresenta uma porta específica para a comunicação com o computador, a mesma pode ser feita através das GPIOs 1 e 3 (TX e RX respectivamente). Para realizar a comunicação existem duas principais soluções: adaptador específico para ESP-01 ou conversor USB para serial. O adaptador específico é um dispositivo prático para a gravação da programação no ESP-01 por apresentar um engate próprio para este dispositivo. A Figura 3.5 apresenta o adaptador.

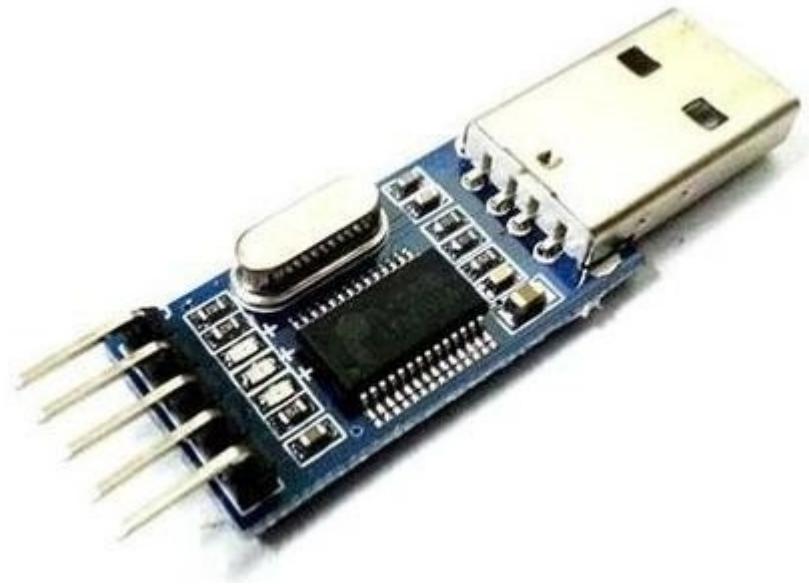
Figura 3.5 – Adaptador ESP-01 para USB.



Fonte: (FILIPEFLOP, 2019a)

O conversor USB para serial é um dispositivo compatível com diversos controladores. Ele apresenta saídas de alimentação, saída RX e saída TX. A vantagem sobre o adaptador específico para ESP-01 é que, com esse conversor, é possível comunicar com o ESP-01 enquanto ele está conectado ao circuito e operando. A Figura 3.6 mostra um conversor USB para serial.

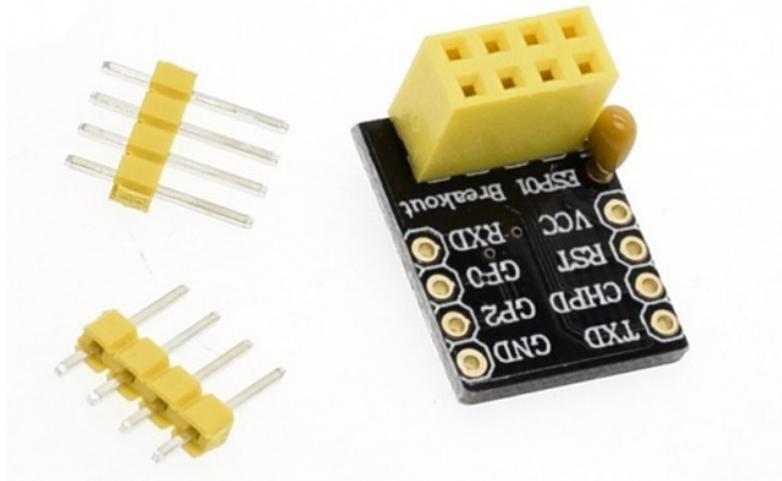
Figura 3.6 – Adaptador USB para serial.



Fonte: (AUTOCORE ROBÓTICA, 2019)

Devido ao espaçamento pequeno entre os pinos do dispositivo, o mesmo não pode ser conectado diretamente à *protoboard*s. Para facilitar sua utilização em *protoboard*s, existe uma base adaptadora que pode ser utilizada, que torna a conexão mais simples e eficiente. A Figura 3.7 apresenta a base.

Figura 3.7 – Base para ESP-01.

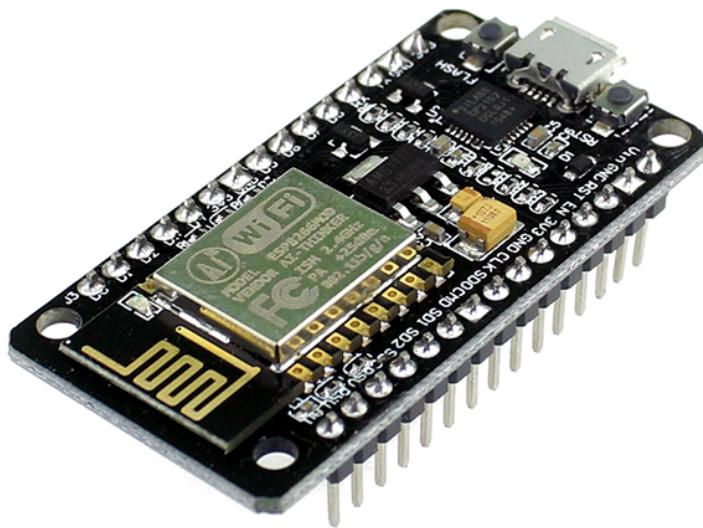


Fonte: (BAÚ DA ELETRÔNICA, 2019)

3.2.3.2 NodeMCU

O módulo ESP-12E, também chamado de NodeMCU, é outro módulo pertencente à mesma família, porém conta com maior número de entradas/saídas. Este possui com 15 GPIOs, além de uma entrada analógica. Este também apresenta conexão micro USB para comunicação com o computador para programação e monitoramento via porta serial. A Figura 3.8 apresenta o NodeMCU.

Figura 3.8 – Módulo NodeMCU.



Fonte: (MASTER WALKER SHOP, 2019b)

O dispositivo não é amigável a *protoboard*, devido a distância entre pinos. Para

facilitar a conexão com sensores e outros dispositivos, existe uma base adaptadora. Esta base apresenta as vantagens de possibilitar várias conexões ao mesmo pino e também suporta a alimentação externa entre 6 e 24V. Ainda possui conexões para alimentação de dispositivos na mesma tensão da fonte, e tensão regulada para 5V e 3,3V. A Figura 3.9 apresenta a base adaptadora.

Figura 3.9 – Base adaptadora para NodeMCU.



Fonte: (MASTER WALKER SHOP, 2019a)

3.2.4 Módulo Relé

Este é um módulo que aciona um relé a partir de uma saída digital do Arduino. Possui contatos NA (normalmente aberto), NF (normalmente fechado) e C (comum) e suporta corrente de até 10 ampères. Pode ser utilizado tanto para o acionamento simples de lâmpadas, para a fechadura eletrônica e para o acionamento da válvula da banheira. A Figura 3.10 apresenta um módulo relé.

Figura 3.10 – Módulo relé.



Fonte: (FILIPEFLOP, 2018a)

Há também uma versão de módulo relé projetada especialmente para o uso com o módulo ESP-01. Com ele é possível conectar a placa diretamente no módulo, tornando o conjunto bem compacto. O módulo alimenta o ESP-01, o mesmo possui um regulador de tensão AMS1117 que possibilita alimentação com tensão entre 5 e 12V (ADVANCED MONOLITHIC, 2014). A Figura 3.11 apresenta o módulo.

Figura 3.11 – Módulo relé para ESP-01.



Fonte: (USINAINFO, 2019)

3.2.5 Módulo Dimmer

Este módulo é utilizado em controle de luminosidade de lâmpadas e velocidade de ventiladores. É construído utilizando um Triac BT137, permitindo o uso com dispositivos com corrente de até 8 ampères (PHILIPS SEMICONDUCTORS, 1997). O módulo possui a alimentação em 5V, uma saída e uma entrada de sinal. O módulo gera um sinal de *Zero Cross* que o controlador usa para calcular o ângulo de disparo do Triac. A Figura 3.12 apresenta o módulo dimmer.

Figura 3.12 – Módulo dimmer.

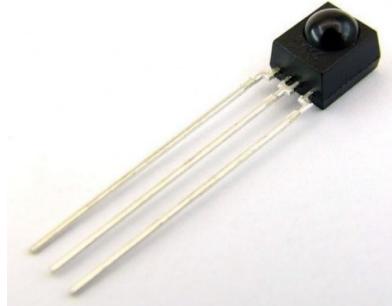


Fonte: (USINAINFO, 2018)

3.2.6 Receptor e Transmissor Infravermelho

Para a realização do controle de ar condicionado é necessário o uso de transmissores e receptores infravermelhos. O receptor tem a função de captar o sinal infravermelho para cadastramento dos comandos do controle remoto dos dispositivos. A Figura 3.13 apresenta um receptor infravermelho.

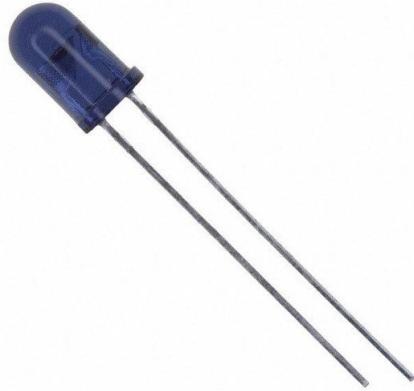
Figura 3.13 – Receptor Infravermelho IRM-3638T.



Fonte: (ELETRODEX, 2018)

O transmissor possui a função de enviar os comandos a partir do controlador. A Figura 3.14 apresenta um LED infravermelho (LED IR) que pode ser utilizado para esta função.

Figura 3.14 – LED infravermelho.



Fonte: (ELETROGATE, 2019)

3.2.7 Sensor de Luminosidade

Para captar o nível de luminosidade é utilizado um LDR (*Light Dependent Resistor*), que varia sua resistência proporcionalmente a luz, podendo determinar a intensidade da luz, e não só se está claro ou escuro (on/off) (SUNROM, 2008). A Figura 3.15 apresenta um LDR.

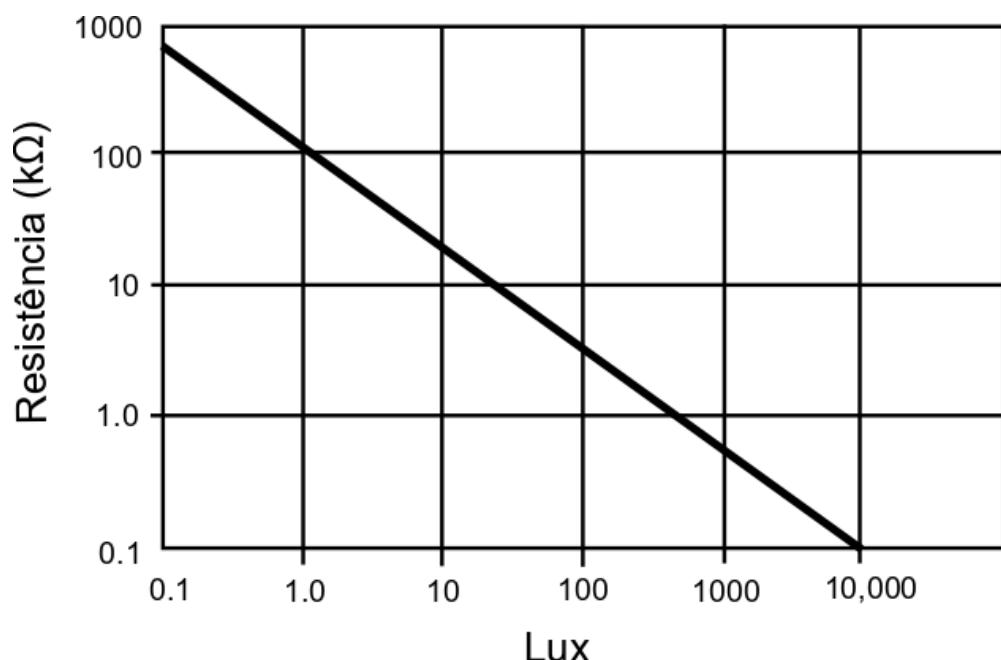
Figura 3.15 – LDR.



Fonte: (INVENT ELECTRONICS, 2018)

A Figura 3.16 apresenta a curva de resposta da resistência ôhmica do LDR em relação a iluminação que o mesmo está detectando no ambiente instalado.

Figura 3.16 – Curva da resistência em relação à iluminação.

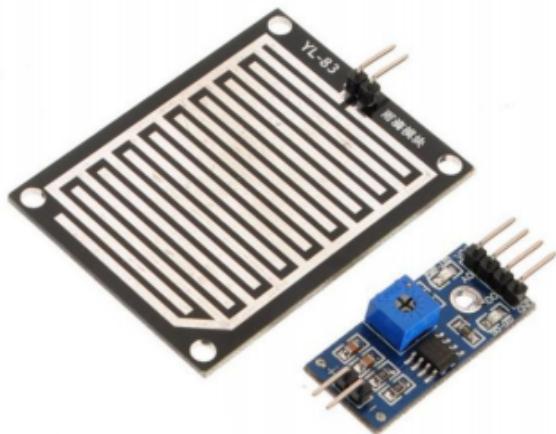


Fonte: Adaptado de (SUNROM, 2008)

3.2.8 Sensor de Chuva

O módulo Yl-83 é capaz de detectar chuva ou neve, e tem sua sensibilidade ajustada através de um potenciômetro. O mesmo possui uma saída analógica e uma digital (VAISALA, 2015). A Figura 3.17 mostra o módulo e o sensor para chuva.

Figura 3.17 – Sensor de chuva YL-83.

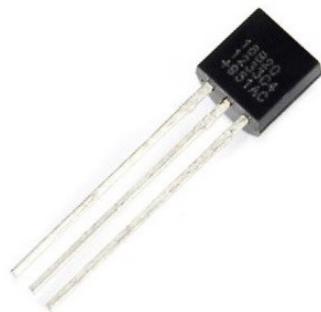


Fonte: Adaptado de (VAISALA, 2015)

3.2.9 Sensor de Temperatura

Para o monitoramento da temperatura ambiente é utilizado o sensor DS18B20. O sensor é capaz de detectar temperaturas entre -55°C e +125°C. Na faixa entre -10°C e 85°C apresenta precisão de $\pm 0,5^{\circ}\text{C}$. A saída do sensor é digital e utiliza o protocolo 1-Wire, que possibilita ligar vários sensores à mesma entrada digital de um controlador (MAXIM INTEGRATED, 2018). Este sensor pode ser encontrado em dois tipos de encapsulamentos. A Figura 3.18 apresenta o encapsulamento normal de circuito integrado.

Figura 3.18 – Sensor DS18B20.



Fonte: (FILIPEFLOP, 2019d)

Outro encapsulamento encontrado é com o circuito integrado dentro de uma ponteira de inox. Esta forma de encapsulamento permite seu uso em ambientes com alta umidade e até mesmo seu uso submerso. A Figura 3.19 apresenta o encapsulamento à prova d'água.

Figura 3.19 – DS18B20 à prova d’água.

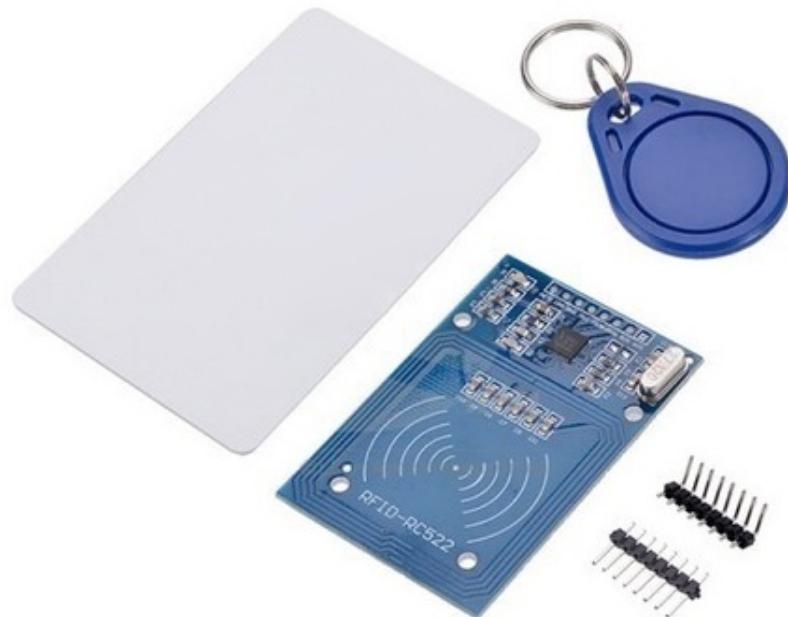


Fonte: (FILIPEFLOP, 2019e)

3.2.10 Leitor RFID

O módulo MFRC522 possibilita a leitura de cartões e chaveiros (*tags*) RFID. Esse é capaz de realizar a leitura do código interno sem contato a curtas distâncias (*contactless*). Opera com tensão de alimentação de 3,3V e na frequência de 13,56MHz (NXP SEMICONDUCTORS, 2016). A Figura 3.20 apresenta o módulo leitor de RFID, um cartão e um Chaveiro.

Figura 3.20 – Kit leitor RFID.



Fonte: (FILIPEFLOP, 2019b)

3.2.11 Mini Fechadura

Este é um dispositivo em aço inoxidável, alimentado em 12V que pode ser utilizado para controle de acesso em portas, gavetas, entre outros locais. A Figura 3.21 apresenta o dispositivo.

Figura 3.21 – Mini fechadura 12V.



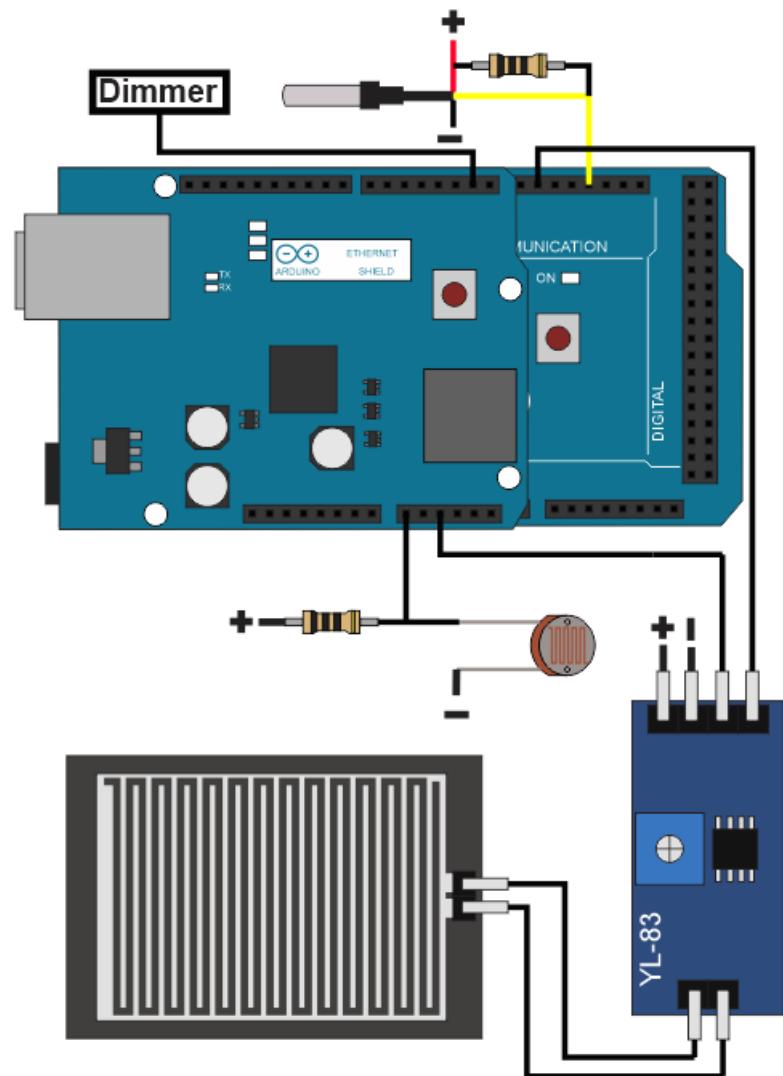
Fonte: (FILIPEFLOP, 2019c)

3.3 APLICAÇÕES

3.3.1 Sensoriamento

Para capturar as variáveis do ambiente, os sensores foram conectados a um conjunto composto por um *Ethernet Shield* acoplado a um Arduino. As variáveis medidas foram: temperatura, luminosidade e chuva. Para isso foram utilizados, respectivamente, os sensores: DS18B20, LDR e YL-83. O Arduino ainda é responsável por enviar o sinal de nível de luminosidade ao conjunto responsável pela função dimmer. O código implementado encontra-se no Apêndice A. A Figura 3.22 apresenta o esquema de ligação.

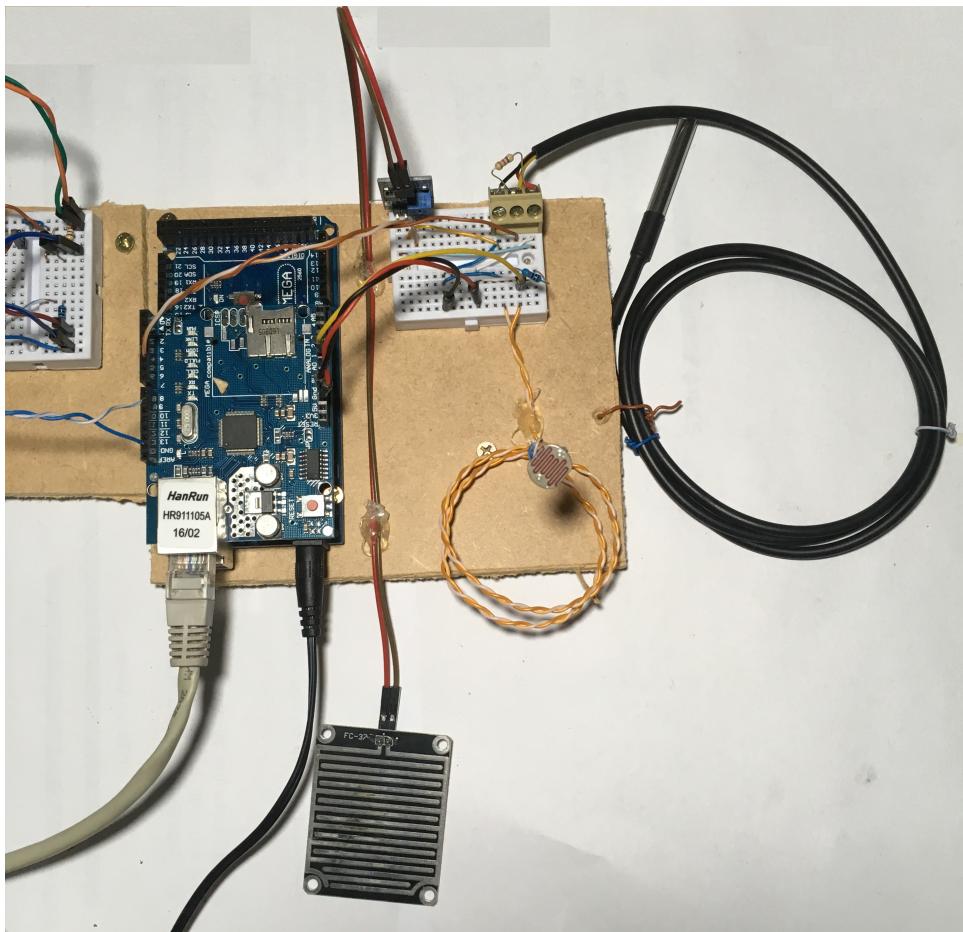
Figura 3.22 – Esquema de ligação do Arduino.



Fonte: Autor.

O circuito implementado é apresentado na Figura 3.23.

Figura 3.23 – Circuito implementado do Arduino.

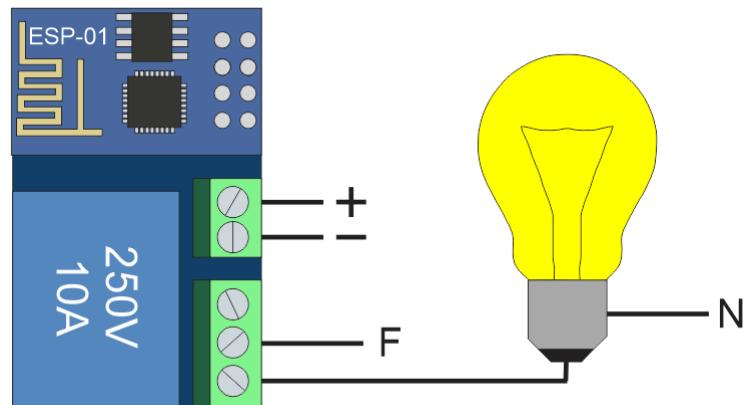


Fonte: Autor.

3.3.2 Iluminação

Para a iluminação é utilizado o módulo ESP-01 em conjunto com o módulo relé, devido a diferente tensão de alimentação das lâmpadas da residência e dos módulos. O ESP aciona o relé que por sua vez liga a lâmpada. O código implementado para esta solução encontra-se no Apêndice B. A Figura 3.24 apresenta um esquema simplificado de ligação das lâmpadas.

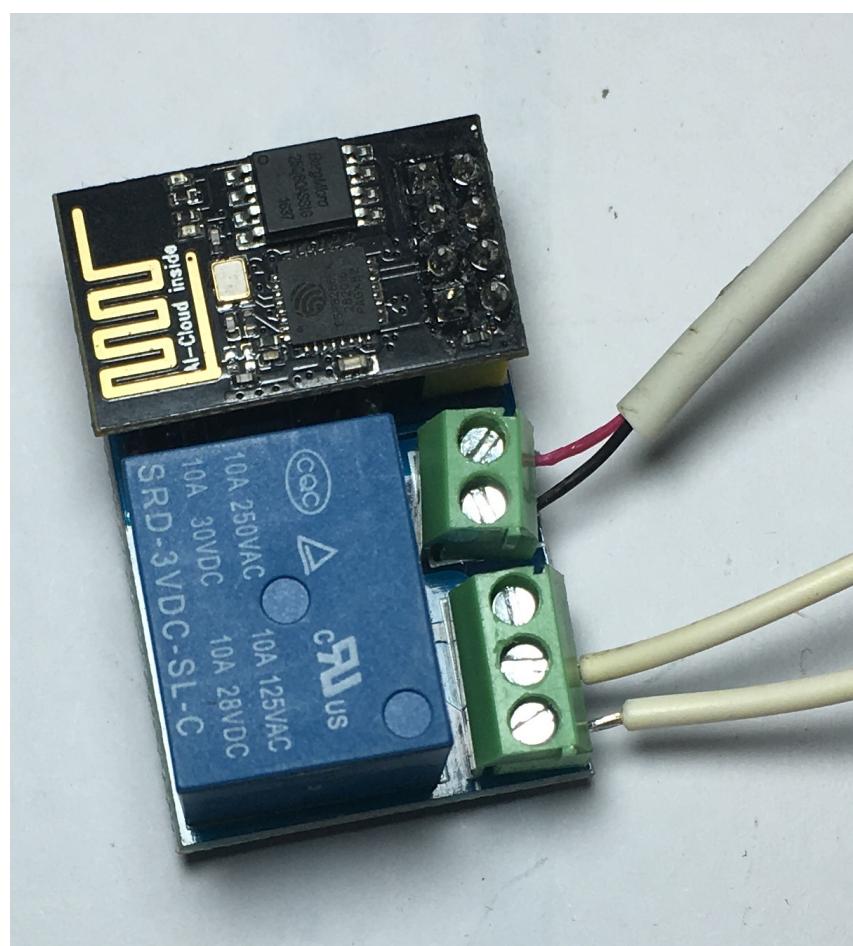
Figura 3.24 – Esquema de ligação de lâmpadas.



Fonte: Autor.

O circuito implementado para esta solução é apresentado na Figura 3.25.

Figura 3.25 – Circuito implementado para acionamento on/off.



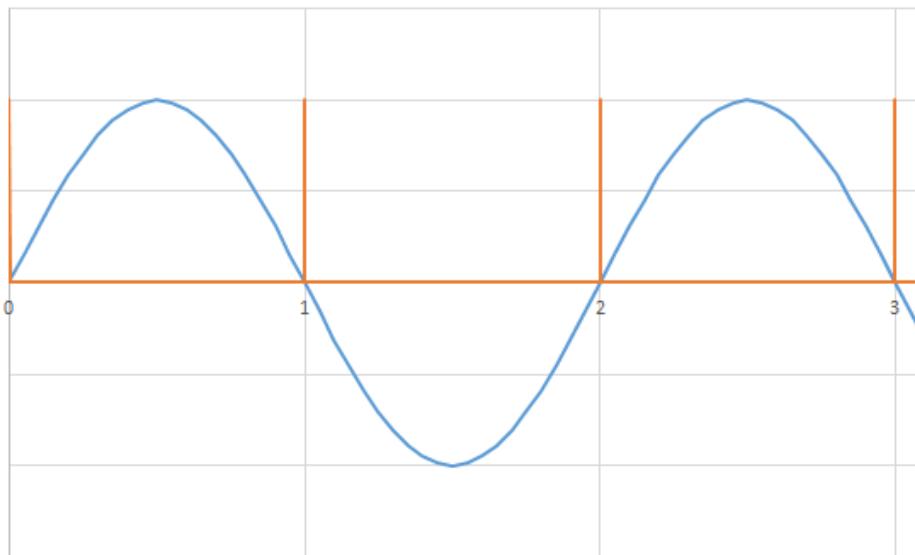
Fonte: Autor.

Nas lâmpadas com controle do nível de iluminação é necessária a utilização do

módulo dimmer, este módulo apresenta a entrada DIM, e a saída ZC (*Zero Cross*).

A saída ZC emite um pulso cada vez que o sinal senoidal da entrada AC do dimmer cruza o zero, conforme mostrado na Figura 3.26.

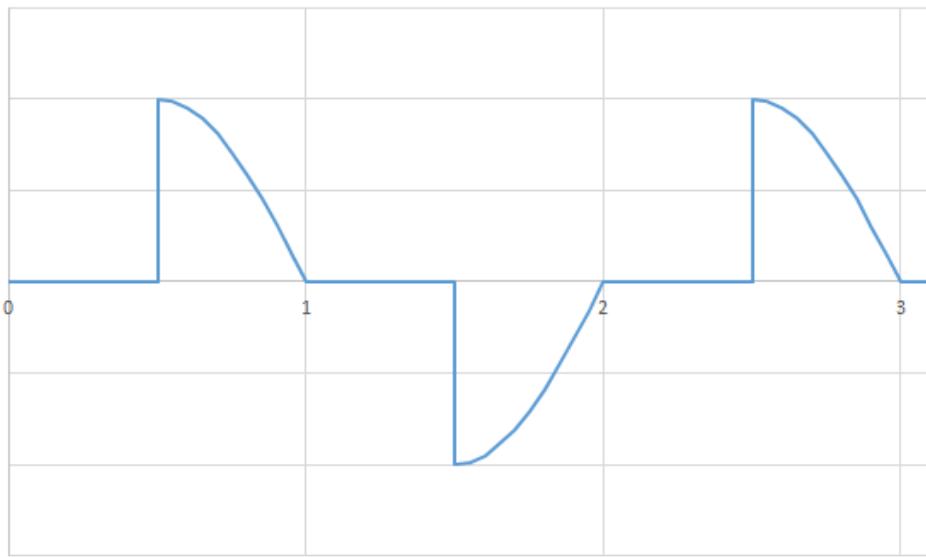
Figura 3.26 – Sinal de entrada AC e *Zero Cross*.



Fonte: Autor.

Uma vez definido o valor de luminosidade desejado, cada vez que o controlador recebe um sinal de *zero cross* ele interrompe sua rotina de execução e calcula a razão de tempo que o TRIAC do dimmer deve conduzir, enviando um sinal para a entrada DIM do dimmer. A Figura 3.27 apresenta um exemplo de sinal de saída do dimmer para uma luminosidade de 50%.

Figura 3.27 – Sinal de saída do dimmer.

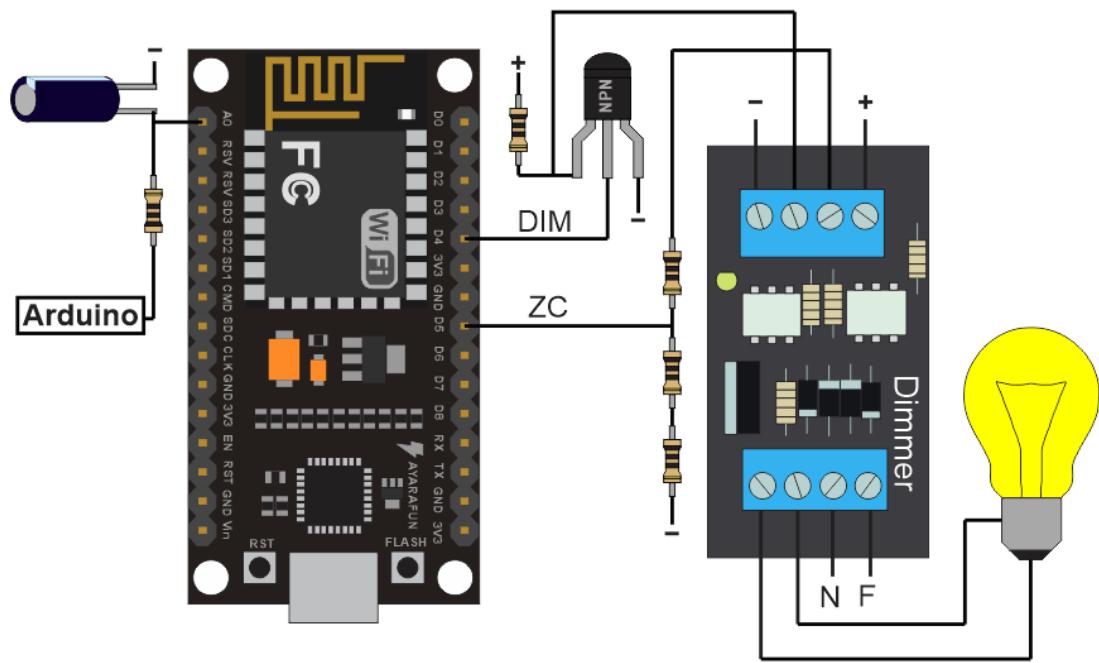


Fonte: Autor.

A frequência de interrupções causada pelo dimmer acabou impossibilitando a comunicação direta do dispositivo que controla o dimmer com o sistema supervisório. Para contornar o problema, foi utilizada uma porta analógica para realizar a leitura de um sinal vindo a partir do Arduino, para definir a luminosidade desejada. Uma saída PWM do Arduino fornece tensão a um capacitor para que o mesmo carregue, transformando a tensão pulsante em tensão estável, e o NodeMCU realize a leitura.

Como o módulo dimmer funciona em 5V e o NodeMCU em 3,3V algumas adaptações foram necessárias. Para reduzir a tensão do sinal *zero cross* foi utilizado um divisor resistivo com 3 resistores de mesmo valor de resistência. Para o sinal DIM foi utilizado um transistor chaveado a partir do NodeMCU. O código implementado para esta solução encontra-se no Apêndice C. A Figura 3.28 apresenta o esquema de ligação completo.

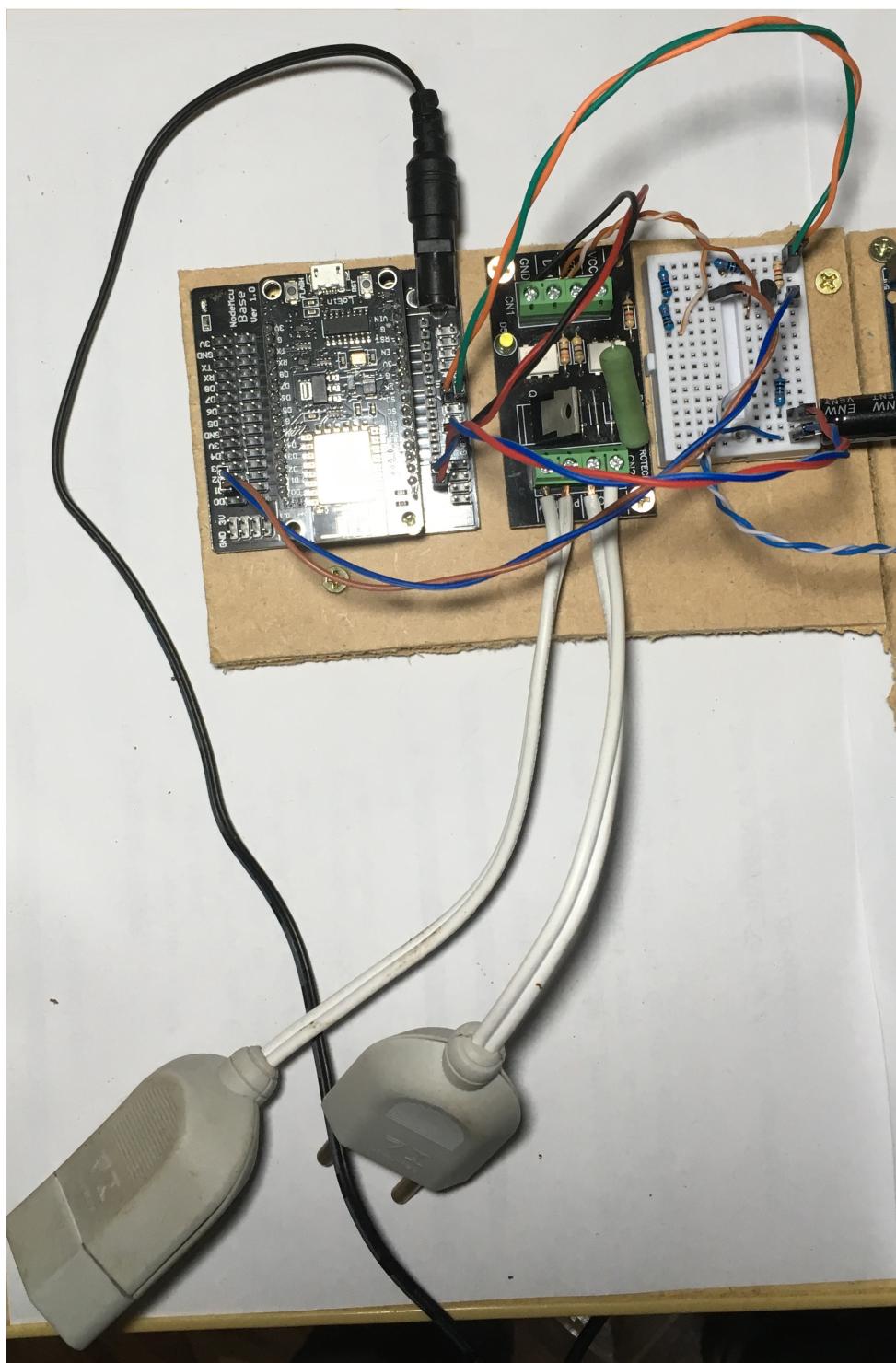
Figura 3.28 – Esquema de ligação do dimmer.



Fonte: Autor.

O circuito implementado para esta solução é apresentado na Figura 3.29.

Figura 3.29 – Circuito implementado para acionamento com Dimmer.



Fonte: Autor.

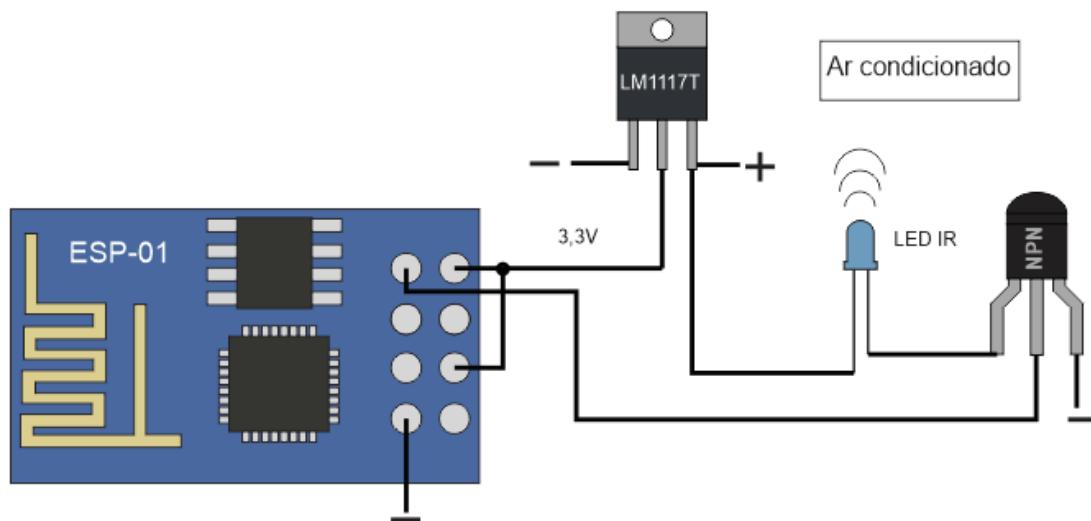
3.3.3 Controle de Ar Condicionado

O controle realizado no ar condicionado foi apenas on/off, a partir da cópia dos códigos infravermelho do controle. Para a cópia dos códigos foi utilizado o receptor IRM-3638T, ligado a uma entrada digital do Arduino.

Os códigos copiados foram armazenados em um ESP-01. O mesmo recebe o valor de temperatura do Arduino e compara com níveis estabelecidos pelo usuário para ligar e desligar o ar condicionado.

Como o ESP-01 fornece apenas 3,3V em suas GPIOs e uma baixa corrente, o LED infravermelho foi acionado a partir de um transistor que é chaveado pelo ESP-01. O código implementado para esta solução encontra-se no Apêndice D. A Figura 3.30 apresenta o circuito desenvolvido.

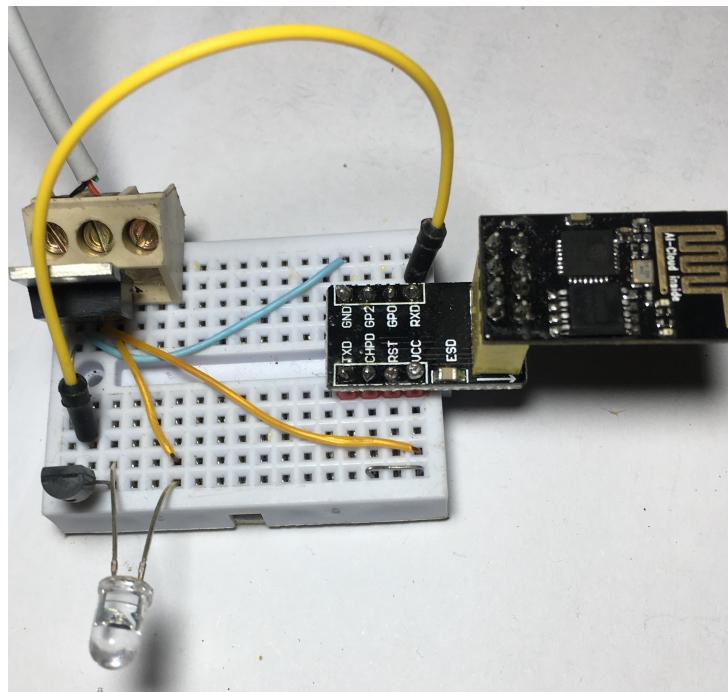
Figura 3.30 – Esquema de ligação para controle de ares condicionados.



Fonte: Autor.

O circuito implementado para esta solução é apresentado na Figura 3.31.

Figura 3.31 – Circuito implementado para acionamento de ar condicionado.

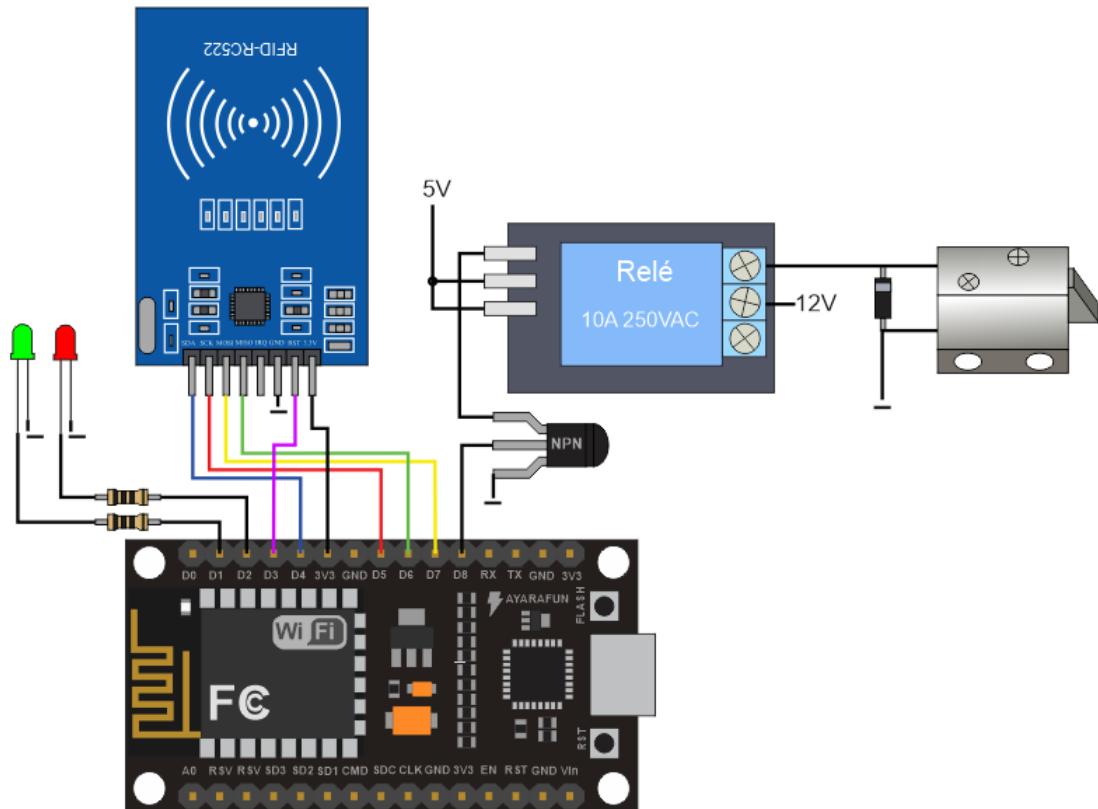


Fonte: Autor.

3.3.4 Controle de Fechadura

A fechadura é acionada por um módulo relé a partir de um NodeMCU ligado a um leitor de RFID. Caso o cartão/chaveiro lido esteja cadastrado, o LED verde acenderá e a fechadura irá abrir durante 5 segundos, caso contrário, o LED vermelho acenderá por 5 segundos e o sistema será habilitado para ler novamente. O código utilizado para esta solução encontra-se no Apêndice E. A Figura 3.32 mostra o esquema de ligação.

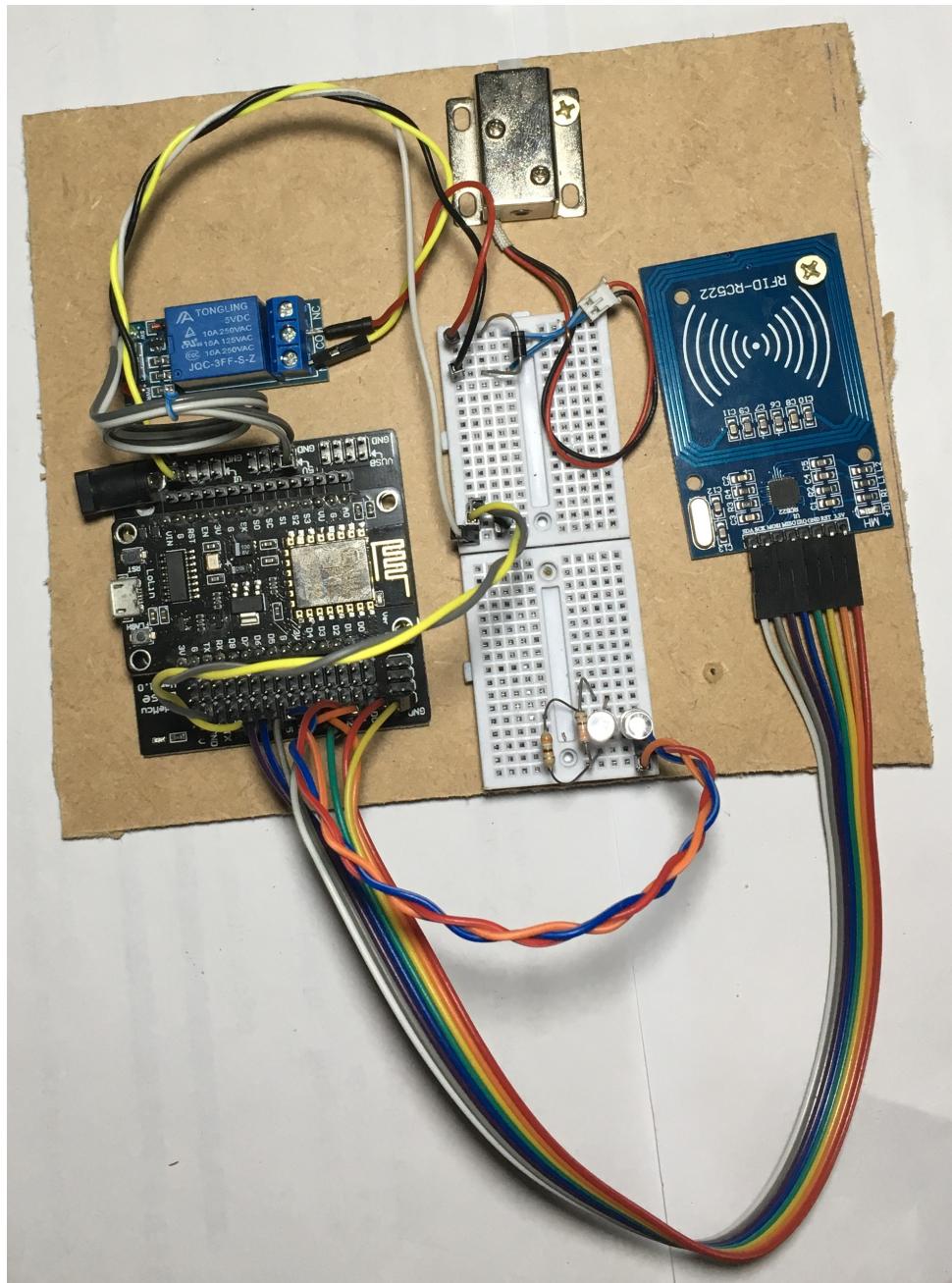
Figura 3.32 – Esquema de ligação para controle de fechaduras.



Fonte: Autor.

O circuito implementado para esta solução é apresentado na Figura 3.33.

Figura 3.33 – Circuito implementado para a fechadura eletrônica.



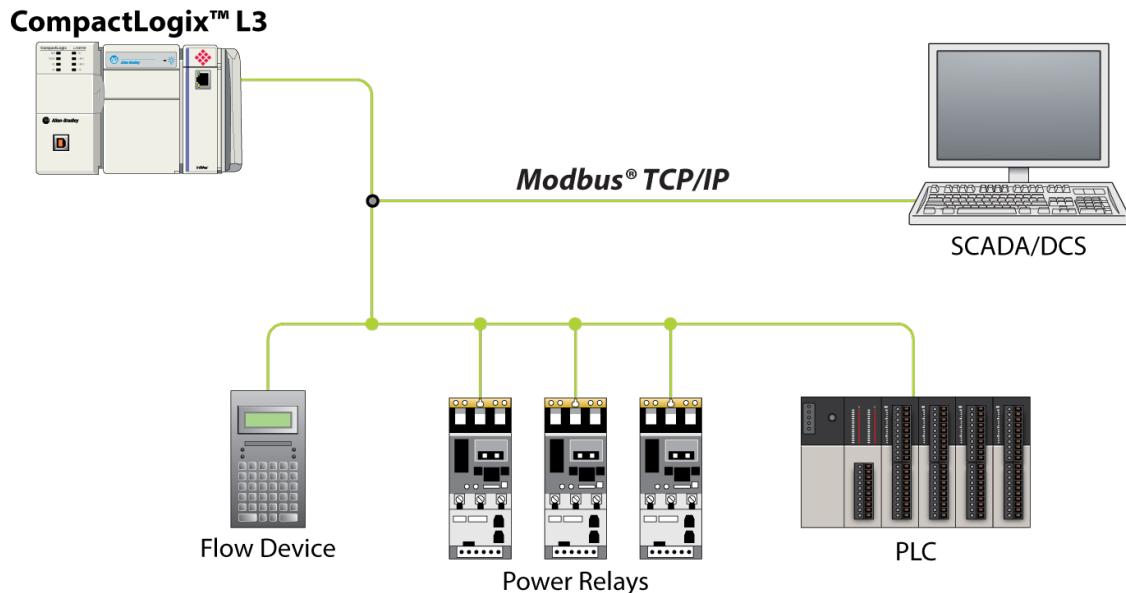
Fonte: Autor.

3.4 PROTOCOLO MODBUS

O protocolo Modbus foi desenvolvido pela empresa Modicon em 1979. Ele é utilizado para estabelecer comunicação entre dispositivos mestre/escravo ou servidor/cliente, como controladores programáveis, sensores e computadores (MODBUS ORGANIZATION, 2019). O protocolo pode ser implementado a partir de conexões do tipo RS-232, RS-485 e *Ethernet* (Modbus TPC/IP). A Figura 3.34 apresenta um exemplo de utilização

do protocolo Modbus via *Ethernet*.

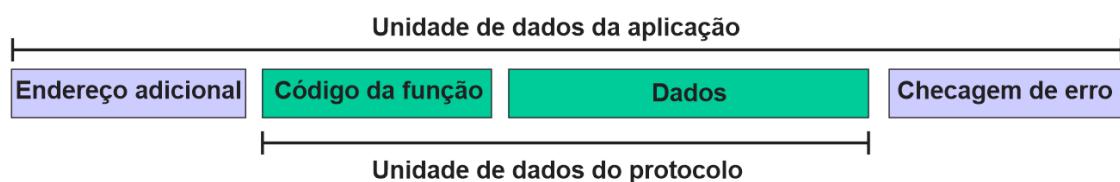
Figura 3.34 – Exemplo de utilização protocolo Modbus.



Fonte: (PROSOFT TECHNOLOGY, 2019)

O protocolo define uma simples unidade de dados independente das outras camadas de comunicação. A utilização do protocolo Modbus, em barramentos específicos de comunicação ou em redes, pode adicionar campos extras na unidade de dados da aplicação, como visto na Figura 3.35.

Figura 3.35 – Estrutura de dados do protocolo.



Fonte: Adaptado de (MODBUS ORGANIZATION, 2019)

O protocolo Modbus possui os códigos de função apresentados na Tabela 3.1.

Tabela 3.1 – Códigos de funções do protocolo Modbus.

Código da Função	Significado
0x02	Lê Input Registers
0x01	Lê Coils
0x05	Escreve Coils Únicas
0x0F	Escreve Múltiplas Coils
0x04	Lê Input Registers
0x03	Lê Holding Registers
0x06	Escreve Registers Únicos
0x10	Escreve Múltiplos Registers
0x17	Lê/Escreve Registers
0x16	Mask Write Registers
0x18	Lê Primeira Entrada e Primeira Saída da Fila

Fonte: Adaptado de (JALOUDI, 2019)

O protocolo possui quatro tipos de registradores de dados (ZURAWSKI, 2015):

- *Coil Status* - Também chamado de *Discret Output*, é uma variável binária que pode ser escrita pelo usuário ou operador do sistema.
- *Input Status* - Variável binária que só pode ser lida, sua finalidade é ser apenas monitorada.
- *Holding Register* - Variável de 16 bits que pode ser escrita.
- *Input Register* - Variável de 16 bits que só pode ser lida.

O Modbus tornou-se o padrão na automação industrial, não por uma definição, mas por ter seu uso muito difundido. Atualmente é o protocolo com maior suporte entre os dispositivos de automação industrial (ZURAWSKI, 2015).

3.5 SISTEMA SUPERVISÓRIO

3.5.1 Configurações

3.5.1.1 Data Sources

O primeiro passo para a utilização do *Mango Automation* é adicionar os dispositivos que serão controlados e/ou monitorados, estes são chamados *Data Sources*. Para isso, é

necessário determinar o protocolo de comunicação utilizado, como destacado na Figura 3.36.

Figura 3.36 – Configurações de *Data Sources*.

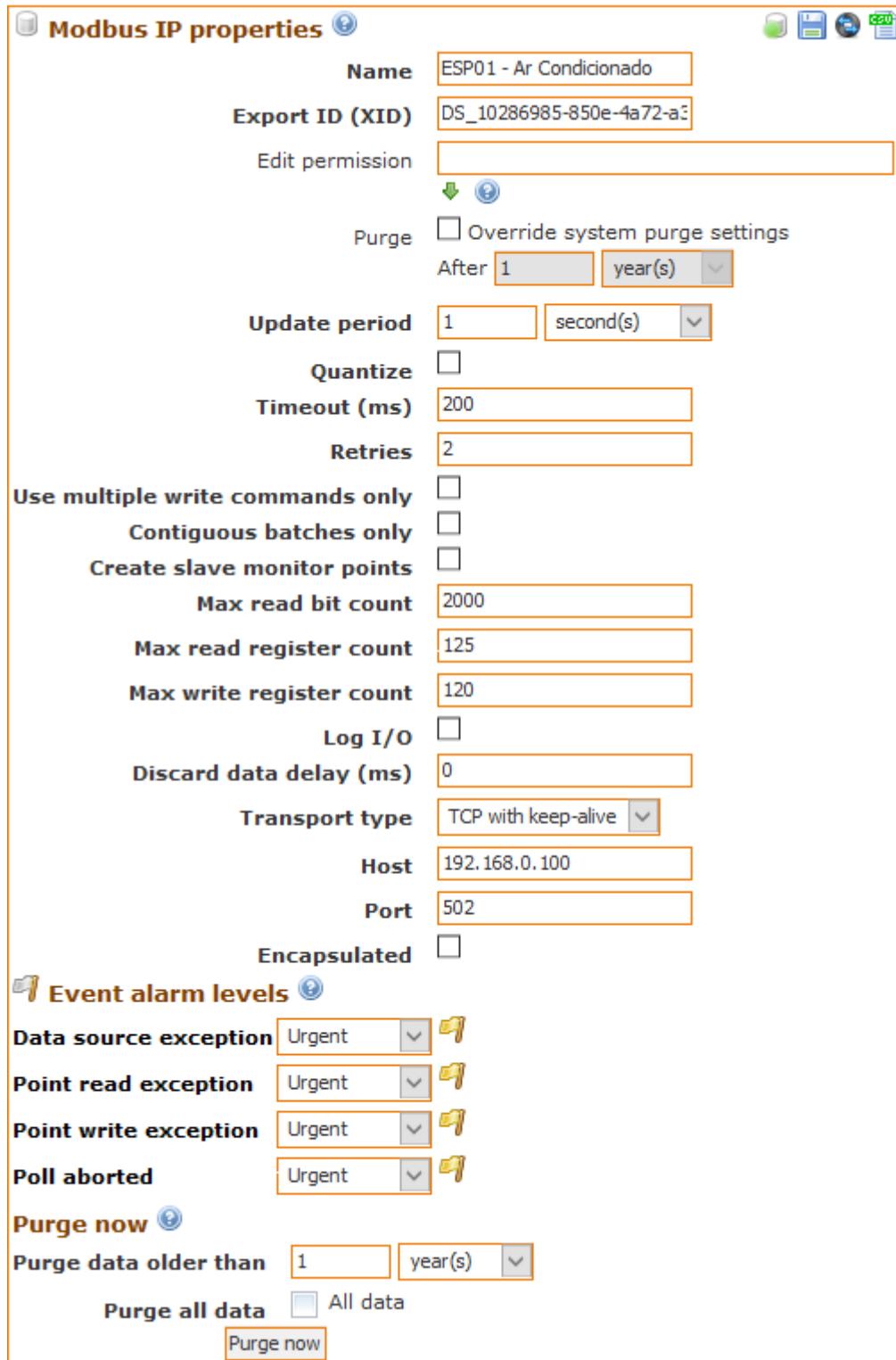
The screenshot shows the Mango Automation software interface. At the top, there is a navigation bar with various icons and the text "My Mango Automation" and "User: admin". Below the navigation bar, there are two tabs: "Data sources" (which is selected) and "Data points". In the main area, there is a search bar with the placeholder text "Modbus I/P" and a dropdown arrow icon. A large black arrow points from the text "Protocolo de comunicação" in the previous slide to this search bar. Below the search bar is a table with the following data:

Name	Type	Connection	Actions
Arduino - Estação	Modbus I/P	192.168.0.108:502	[edit, delete, add, copy]
Arduino_Dimmer	Modbus I/P	192.168.0.110:502	[edit, delete, add, copy]
ESP01 - Ar Condicionado	Modbus I/P	192.168.0.100:502	[edit, delete, add, copy]
ESP01 - Lampada LDR	Modbus I/P	192.168.0.102:502	[edit, delete, add, copy]
ESP01 - Lampada ON/OFF	Modbus I/P	192.168.0.101:502	[edit, delete, add, copy]
Mango Internal	Internal Data Source	10 second(s)	[edit, delete, add, copy]
NodeMCU - RFID	Modbus I/P	192.168.0.104:502	[edit, delete, add, copy]

Fonte: Autor.

Após selecionado o protocolo de comunicação utilizado, deve-se realizar as demais configurações do dispositivo conforme mostrado na Figura 3.37.

Figura 3.37 – Configurações para protocolo Modbus IP.



Fonte: Autor.

As principais configurações são:

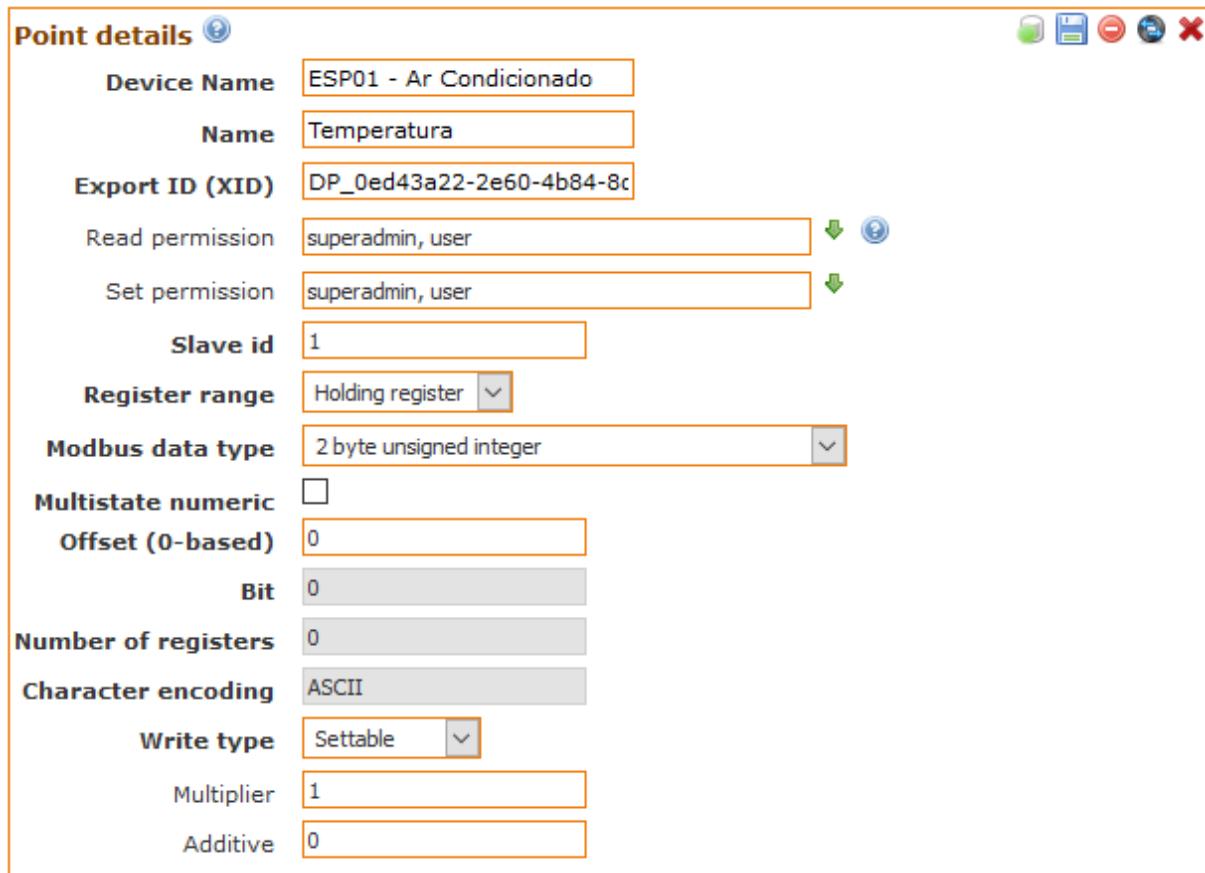
- *Name*: Nome dado ao dispositivo dentro do *Mango Automation*.

- *Update Period*: Define o período entre cada vez que o software atualiza seu valor.
- *Transport Type*: Utilizando a opção ”*TCP with keep-alive*” mantém a conexão ativa mesmo quando não há novos comandos, garantindo melhor tempo de resposta.
- *Host*: Endereço IP do dispositivo na rede.
- *Port*: a porta 502 é a porta padrão para utilização do protocolo Modbus.

3.5.1.2 Data Points

Após o dispositivo ser adicionado, é necessário adicionar as variáveis a serem controladas/monitoradas que estão presentes naquele dispositivo. Essas são chamadas de *Data Points*. As configurações estão apresentadas na Figura 3.38.

Figura 3.38 – Configurações de *Data Points*.



Fonte: Autor.

As principais configurações são:

- *Name*: Nome dado ao *Data Point* dentro do software, independente do nome da variável dentro do dispositivo que o software está acessando.
- *Read Permission*: Níveis de usuário que podem visualizar o valor do *Data Point*.
- *Set Permission*: Níveis de usuário que podem alterar o valor do *Data Point*.
- *Register Range*: No caso do Modbus, tipo de variável do protocolo de comunicação.
- *Modbus Data Type*: Formato do valor do *Data Point*.
- *Offset (0-based)*: Posição da memória do dispositivo onde se encontra o *Data Point*, a partir desse parâmetro que a variável a ser controlada/monitorada é escolhida dentro do dispositivo. Pode haver duas variáveis com o mesmo valor de *Offset*, porém de tipos diferentes.

3.5.1.3 Watch Lists

Após adicionar os dispositivos e pontos a serem monitorados/controlados é possível criar listas para visualizá-los, estas são chamadas *Watch Lists*. A Figura 3.39 apresenta a tela onde são exibidos os pontos monitorados.

Figura 3.39 – Tela de *Watch Lists*.



The screenshot shows a software interface for managing data points. On the left, there is a tree view labeled "Points" with various nodes expanded, such as "Arduino Estação - Chovendo", "ESP-01 Ar Condicionado - Ligado", and "NodeMCU Fechadura - Certo". To the right of the tree view is a table titled "Watch list" with the header "(unnamed)". The table lists 14 data points with their current values and timestamps. Each row includes icons for edit, delete, and other actions. The columns are: Point Name, Current Value, Last Update, and a set of checkboxes and icons for configuration.

Point	Value	Last Update	Actions
ESP-01 Ar Condicionado - Temperatura	17,00 °C	12:03:47	
ESP-01 Ar Condicionado - Ligado	OFF	12:03:47	
Arduino Estação - Temperatura	17,00 °C	12:03:45	
Arduino Estação - Luminosidade Dimmer	0,00 %	12:03:45	
Arduino Estação - LDR Escuro	Claro	12:03:45	
Arduino Estação - Intensidade Chuva	0,00	12:03:45	
Arduino Estação - Chovendo	Não	12:03:45	
ESP-01 Lampada ON/OFF - Liga	OFF	12:03:48	
Lampada LDR - Liga LDR	OFF	12:03:48	
NodeMCU Fechadura - Certo	OFF	12:03:47	
NodeMCU Fechadura - Errado	OFF	12:03:47	

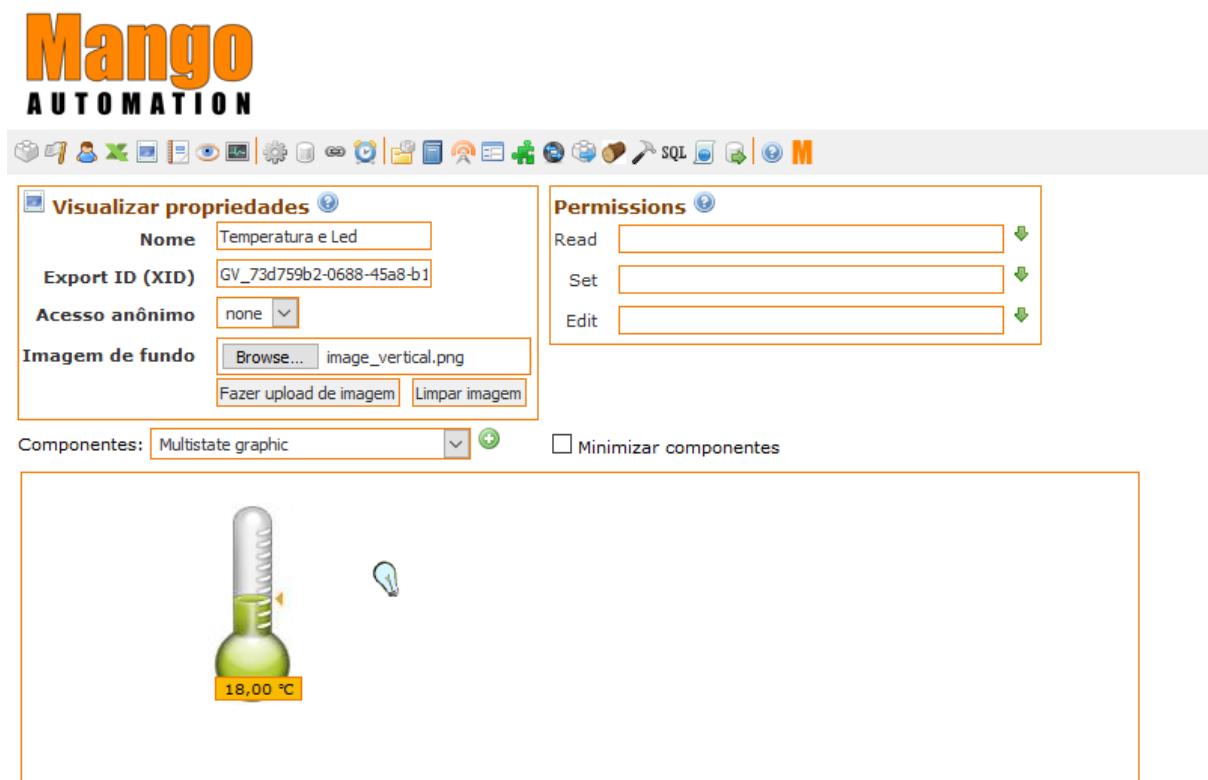
Fonte: Autor.

No lado esquerdo da Figura 3.39 estão listados os pontos para adicionar à *Watch List*, no lado direito estão listados os pontos. Ainda no lado direito é possível alterar os valores dos pontos.

3.5.2 Interface Sistema Supervisório

O software *Mango Automation* apresenta uma ferramenta de criação de telas gráficas, ainda no modo *legacy*, apresentada na Figura 3.40.

Figura 3.40 – Criação de sistemas supervisório em modo *Legacy*.



Fonte: Autor.

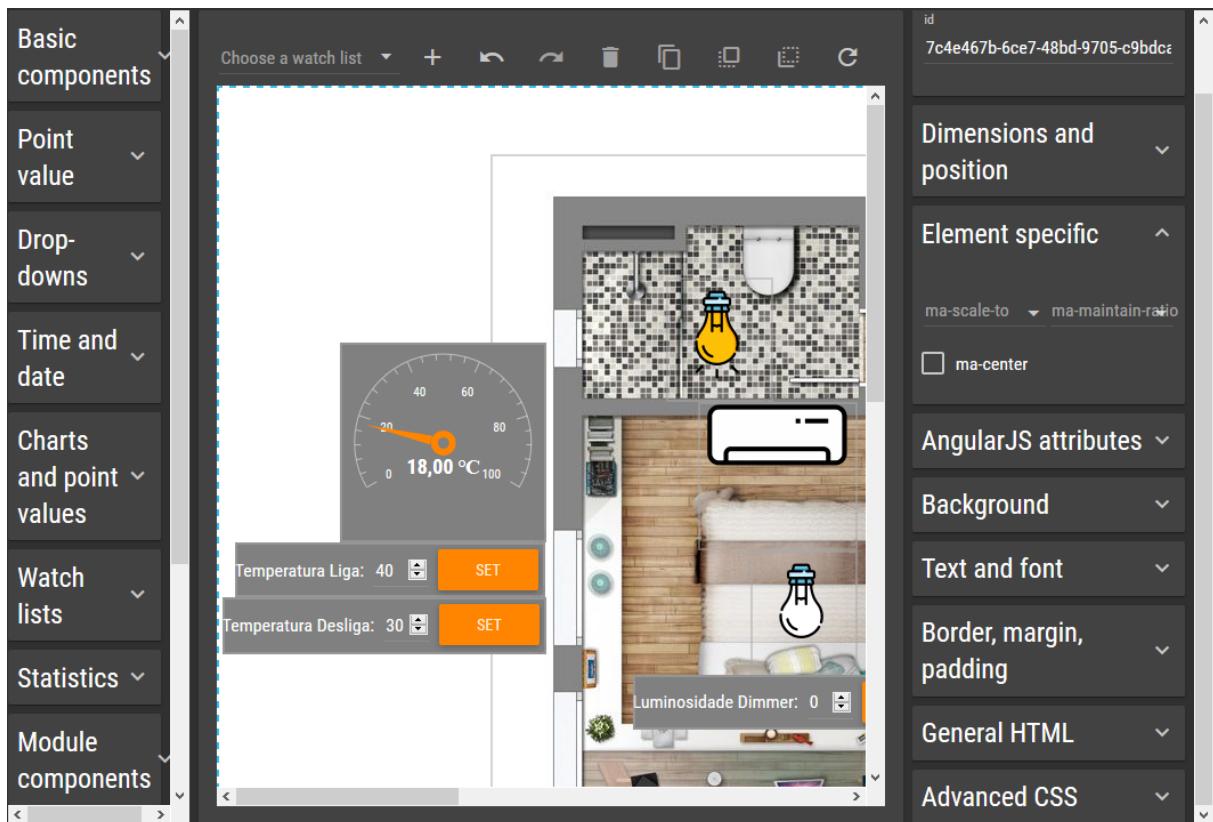
Este modo apresenta recursos muito limitados, poucas opções de edição. Como visto na Figura 3.40, há uma representação de termômetro e uma de lâmpada desproporcionais e sem opções de alteração de tamanho. Não há também opções de importações de novas imagens para os elementos, somente para imagem de fundo. Porém, no modo UI existe a ferramenta *Dashboard Designer*, que proporciona maiores possibilidades de criação de sistemas supervisórios.

3.5.2.1 Dashboard Designer

Este modo de criação de interfaces supervisórias apresenta uma grande capacidade de personalização. Permite o *upload* de imagens, alterações nas escalas, personalizações em fontes, utilização de códigos HTML, entre outras funcionalidades. A Figura 3.41

apresenta a tela do *Dashboard Designer*.

Figura 3.41 – Criação de sistemas supervisório com *Dashboard Designer*.

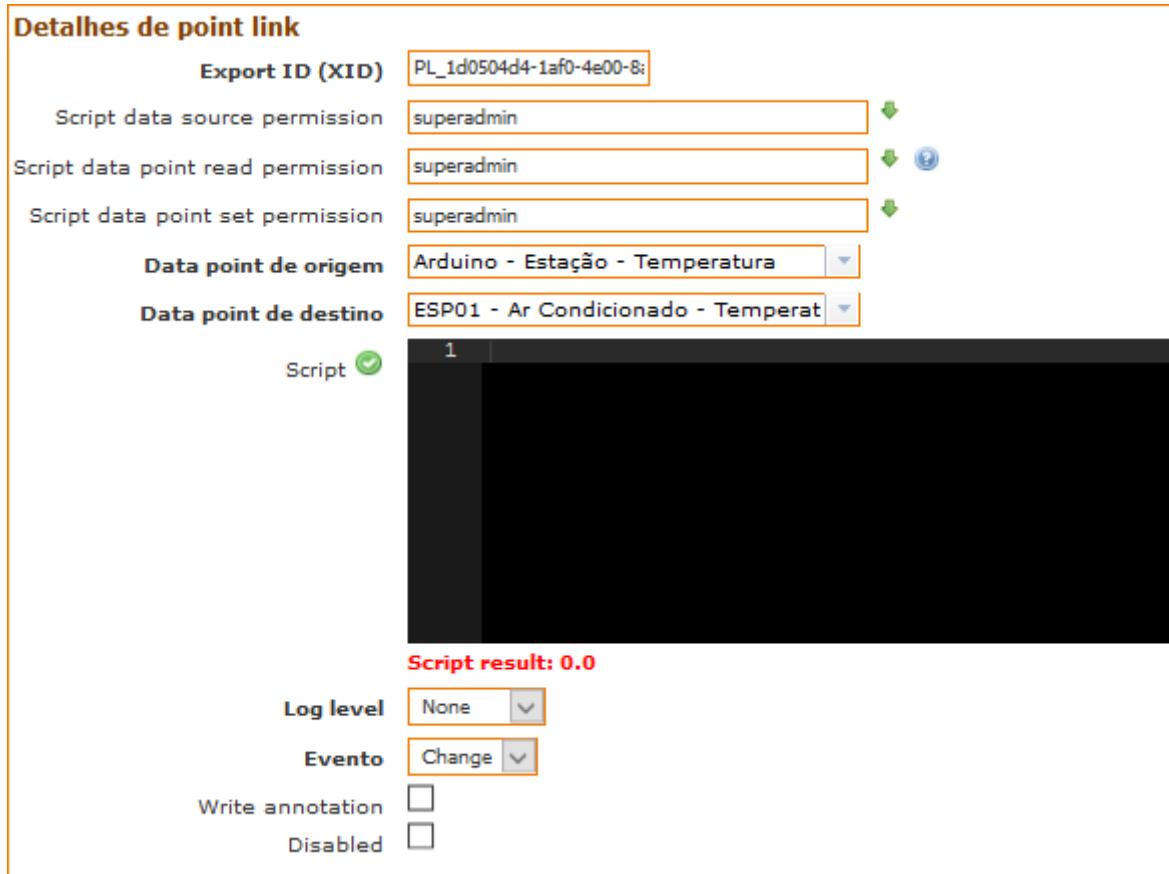


Fonte: Autor.

3.5.3 Funcionalidades

3.5.3.1 Point Links

Com esta função é possível associar valores de diferentes variáveis em diferentes controladores umas com as outras. É possível, por exemplo, associar a leitura de temperatura de um sensor presente em um dispositivo com outro dispositivo, para que esse realize alguma ação de controle. Ainda, pode ser criado um código em JavaScript para realizar múltiplas funções. A Figura 3.42 apresenta a tela de configurações de *Point Links*.

Figura 3.42 – Configurações de *Point Links*.

Fonte: Autor.

3.5.3.2 Eventos Agendados

Esta é uma funcionalidade, apresentada em modo *Legacy*, que permite a criação de eventos, em determinados dias e horários. Com ela é possível, por exemplo, agendar horários para acionamentos de dispositivos e também horários para desligamentos, para evitar que dispositivos sejam esquecidos ligados. A Figura 3.43 apresenta a tela de configurações para criação de eventos.

Figura 3.43 – Configurações de eventos agendados.



Fonte: Autor.

3.5.3.3 Advanced Scheduler

O Advanced Scheduler é um módulo extra, encontrado no modo UI, capaz de criar rotinas de eventos. Este módulo opera na UI, apresentando um modo mais completo e simples para a criação dos eventos. É possível definir diferentes intervalos de tempo para eventos durante a semana, finais de semana ou até intervalos distintos para cada dia, assim como criar exceções. A Figura 3.44 apresenta um exemplo com diferentes intervalos em finais de semana e dias de semana.

Figura 3.44 – Configurações do *Advanced Scheduler*.



Fonte: Autor.

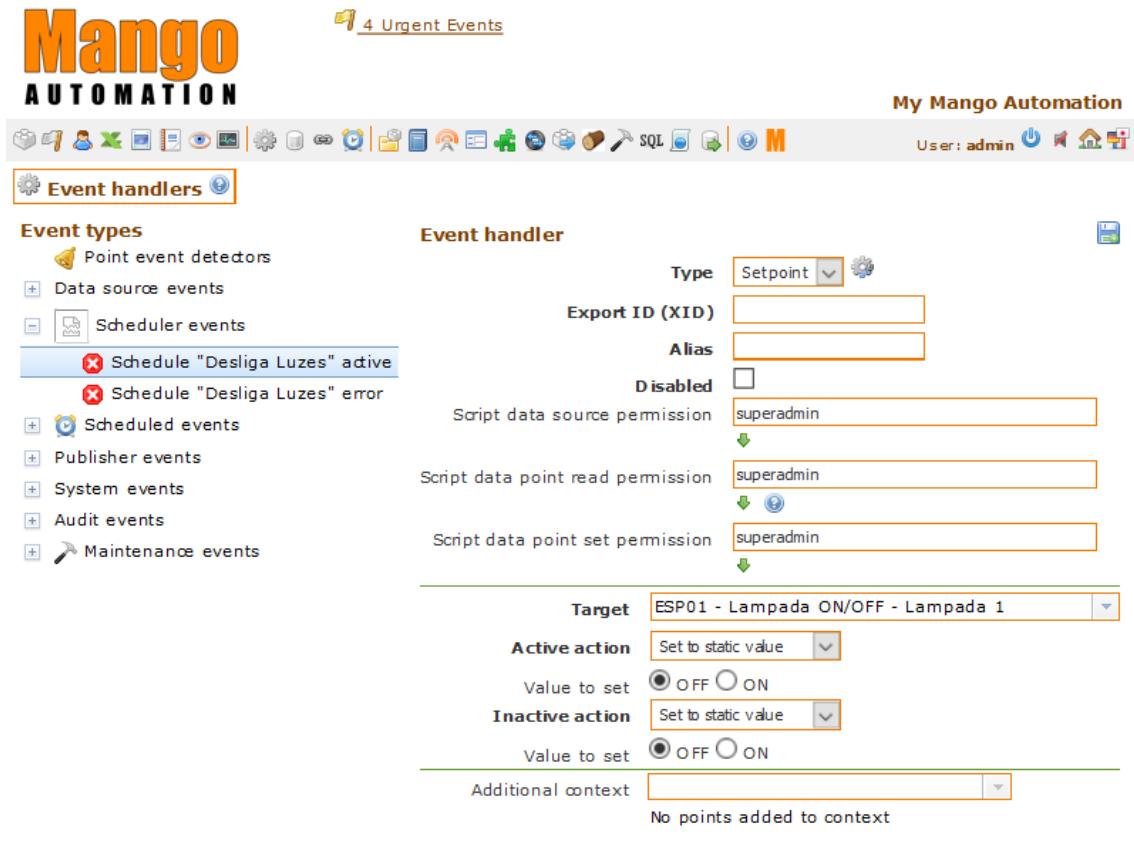
3.5.3.4 Event Handlers

Esta é uma funcionalidade capaz de realizar determinadas ações a partir dos eventos criados ou de eventos do sistema. É possível enviar e-mails de alerta ou alterar o estado dos dispositivos supervisionados. Na segunda opção ainda é possível definir va-

lores determinados ou associá-lo ao valor de outra variável do sistema, similar a função *Point Link*.

Por exemplo, pode se criar um evento para algum horário onde a casa costuma estar vazia. Utilizando o *Event Handler*, é possível, a partir desse evento, definir quais dispositivos serão desligados. A Figura 3.45 apresenta a tela de configuração do *Event Handler*.

Figura 3.45 – Configurações do *Event Handler*.

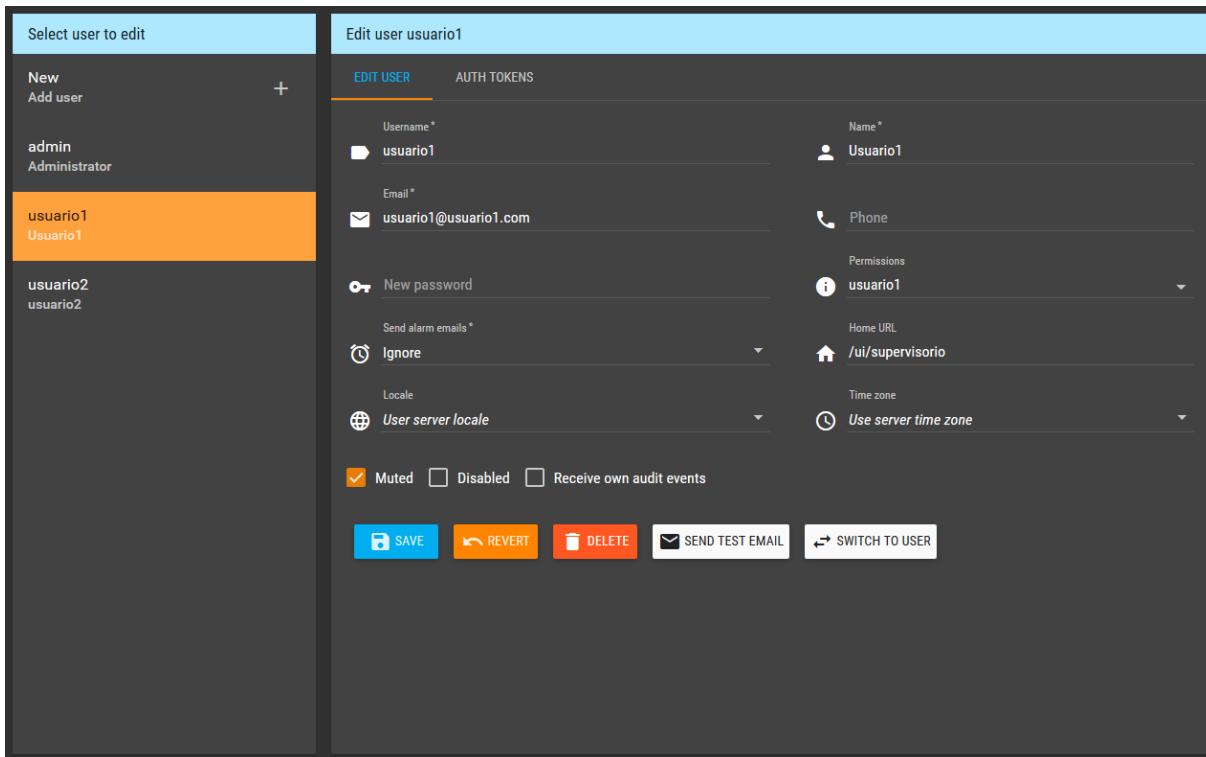


Fonte: Autor.

3.5.3.5 Usuários

Para o acesso ao sistema supervisório é necessário realizar o login no *software* a partir do navegador. O *Mango Automation* oferece a possibilidade de criar usuários com diferentes níveis de acesso. A Figura 3.46 apresenta a tela de configuração de usuários.

Figura 3.46 – Tela de configuração de usuários do *Mango Automation*.



Fonte: Autor.

Uma vez logado como administrador, tem-se acesso a todas as configurações disponíveis no *software*. O administrador também pode definir permissões globais para todos os usuários de mesmo nível. Por exemplo, em uma casa de uma família de dois filhos, pode-se criar a classe filhos e definir menores permissões de acesso para a mesma.

3.5.3.6 Auto Login

É possível habilitar o login automático de um usuário em um determinado dispositivo com a autorização de um usuário de nível administrador. Por exemplo, em uma casa com três moradores, cada um pode ter seu usuário salvo para login automático no sistema em seu *smartphone*, sem a necessidade de inserir o nome de usuário e senha cada vez que desejar utilizar o sistema. O *software* não permite o uso dessa funcionalidade para usuários com permissões de administrador do sistema.

3.5.4 Atuação do Sistema Supervisório

O sistema supervisório comunica-se diretamente com todas as soluções apresentadas, com exceção do dimmer. O supervisório trabalha com a troca de informações das variáveis apresentadas na Figura 3.47 com o Arduino responsável pelo sensoriamento.

Figura 3.47 – *Data points* do Arduino.

Data points									
Device Name	Name	Data type	Set permissions	Read permissions	Slave	Range	Offset (0-based)		
filter text	filter text	filter text	filter text	filter text					
sort	sort	sort	sort	sort					
Arduino Estação	Chovendo	Binary	superadmin	superadmin, usuario2, visita	1	Input register	4/0		
Arduino Estação	Intensidade Chuva	Numeric	superadmin	superadmin, usuario2, visita	1	Input register	2		
Arduino Estação	Lampada Dimmer	Binary	superadmin	superadmin, v...	1	Input register	5/0		
Arduino Estação	LDR Escuro	Binary	superadmin	superadmin, usuario2, visita	1	Input register	1/0		
Arduino Estação	Luminosidade Dimmer	Numeric	superadmin	superadmin, usuario2, visita	1	Holding register	0		
Arduino Estação	Temperatura	Numeric	superadmin	superadmin, usuario2, visita	1	Input register	0		

Fonte: Autor.

A variável "Chovendo" é obtida a partir do sensor de chuva. No sistema desenvolvido tem apenas a função de monitoramento, no entanto, a partir da função *Point Link* poderia ser enviada a outro dispositivo que realizasse alguma ação a partir da mesma. O mesmo vale para a variável "Intensidade Chuva".

A variável "Luminosidade Dimmer" é a variável que o usuário define o valor da luminosidade desejada, para que o Arduino envie um sinal para o NodeMCU responsável pelo controle do módulo dimmer. A variável "Lampada Dimmer" tem a função somente de indicar que a lâmpada acionada pelo dimmer está ligada.

A variável "LDR Escuro" é uma variável de monitoramento. A mesma fica em nível lógico alto quando o ambiente está escuro, a partir de um nível pré determinado. Esta variável é utilizada com a função *Point Link* para acionar uma lâmpada.

A variável "Temperatura" é a variável que informa a temperatura ambiente. A mesma é utilizada com a função *Point Link* para o dispositivo responsável pelo controle do ar condicionado. Esta variável é utilizada também para mostrar a temperatura dentro do sistema supervisório.

Os módulos responsáveis pelos acionamentos on/off trocam informações de apenas uma variável com o supervisório. A Figura 3.48 apresenta o *Data Point* comunicado.

Figura 3.48 – *Data point* do ESP-01 para acionamento on/off.

Device Name	Name	Data type	Set permissions	Read permissions	Slave	Range	Offset (0-based)	
filter text sort	filter text sort	filter text sort	filter text sort	filter text sort				
ESP-01 Lampada ON/OFF	Liga	Binary	superadmin	superadmin, usuario2, visita	1	Coil status	0	

Fonte: Autor.

Nesta aplicação, o sistema implementado para a fechadura não aceita comandos a partir do sistema supervisório. Há somente a comunicação de quando foi utilizado um chaveiro/cartão válido ou inválido, realizado a partir das variáveis "Certo" e "Errado", respectivamente. A Figura 3.49 apresenta as variáveis utilizadas.

Figura 3.49 – *Data points* do NodeMCU da fechadura.

Device Name	Name	Data type	Set permissions	Read permissions	Slave	Range	Offset (0-based)	
filter text sort	filter text sort	filter text sort	filter text sort	filter text sort				
NodeMCU Fechadura	Certo	Binary	superadmin	superadmin, usuario2, visita	1	Input status	0	
NodeMCU Fechadura	Errado	Binary	superadmin	superadmin, usuario2, visita	1	Input status	1	

Fonte: Autor.

Com o módulo responsável pelo controle do ar condicionado há a comunicação de 4 variáveis, apresentadas na Figura 3.50.

Figura 3.50 – *Data points* do ESP-01 para controle do ar condicionado.

Device Name	Name	Data type	Set permissions	Read permissions	Slave	Range	Offset (0-based)	
filter text sort	filter text sort	filter text sort	filter text sort	filter text sort				
ESP-01 Ar Condicionado	Ligado	Binary	superadmin	superadmin, usuario2, visita	1	Input status	0	
ESP-01 Ar Condicionado	Temperatura	Numeric	superadmin	superadmin, usuario2, visita	1	Holding register	0	
ESP-01 Ar Condicionado	Temperatura Desliga	Numeric			1	Holding register	2	
ESP-01 Ar Condicionado	Temperatura Liga	Numeric			1	Holding register	1	

Fonte: Autor.

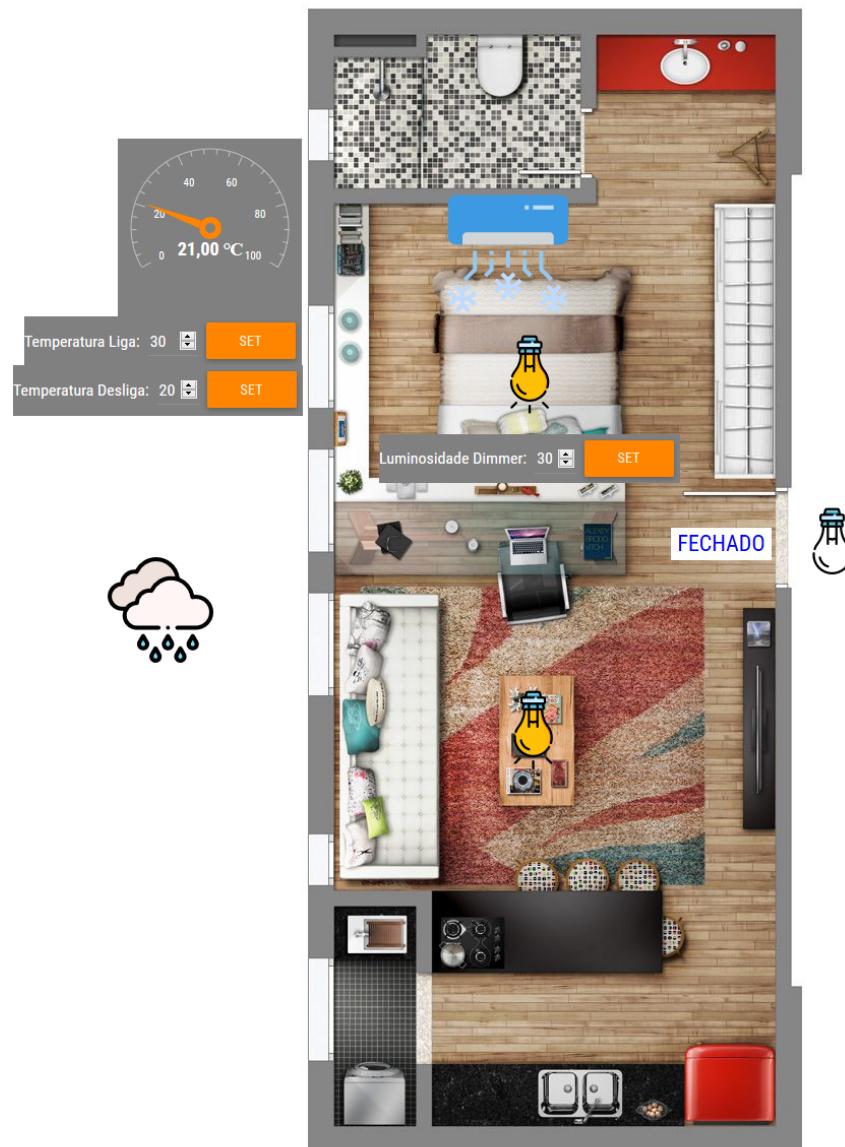
A variável "Ligado" é responsável por informar quando o ar condicionado encontra-se ligado ou desligado. A variável "Temperatura" é recebida com o comando *Point Link*,

a partir do sensor instalado juntamente ao Arduino. As variáveis "Temperatura Liga" e "Temperatura Desliga" são inseridas pelo usuário com a finalidade de definir quando o ar condicionado deve ligar e desligar.

3.6 RESULTADOS

Com o intuito de que o sistema tenha um uso simples e intuitivo, a interface do sistema supervisório desenvolvida ficou com a aparência apresentada na Figura 3.51.

Figura 3.51 – Sistema supervisório desenvolvido



Fonte: Autor.

A partir da interface, para o controle das lâmpadas on/off, um clique no ícone da lâmpada alterna seu estado. A lâmpada externa, mesmo nessa aplicação que é acionada com um sinal do sensor de luminosidade, pode também ser acionada ou desligada através do sistema supervisório.

A lâmpada acionada por dimmer pode ter seu nível luminoso definido a partir da interface. Assim como as temperaturas nas quais o ar condicionado deve ligar e desligar.

O ícone das nuvens e da chuva aparece quando é detectado chuva pelo sensor de chuva, e quando não há chuva o ícone desaparece. Na porta há uma exibição escrita de quando encontra-se com a fechadura fechada ou aberta.

Há ainda um indicador de temperatura, para melhor visualização, e um ícone de ar condicionado que muda a aparência conforme o estado do aparelho.

Existem menus para a visualização dos valores dos *Data Points* nos dispositivos, como visto na Figura 3.52.

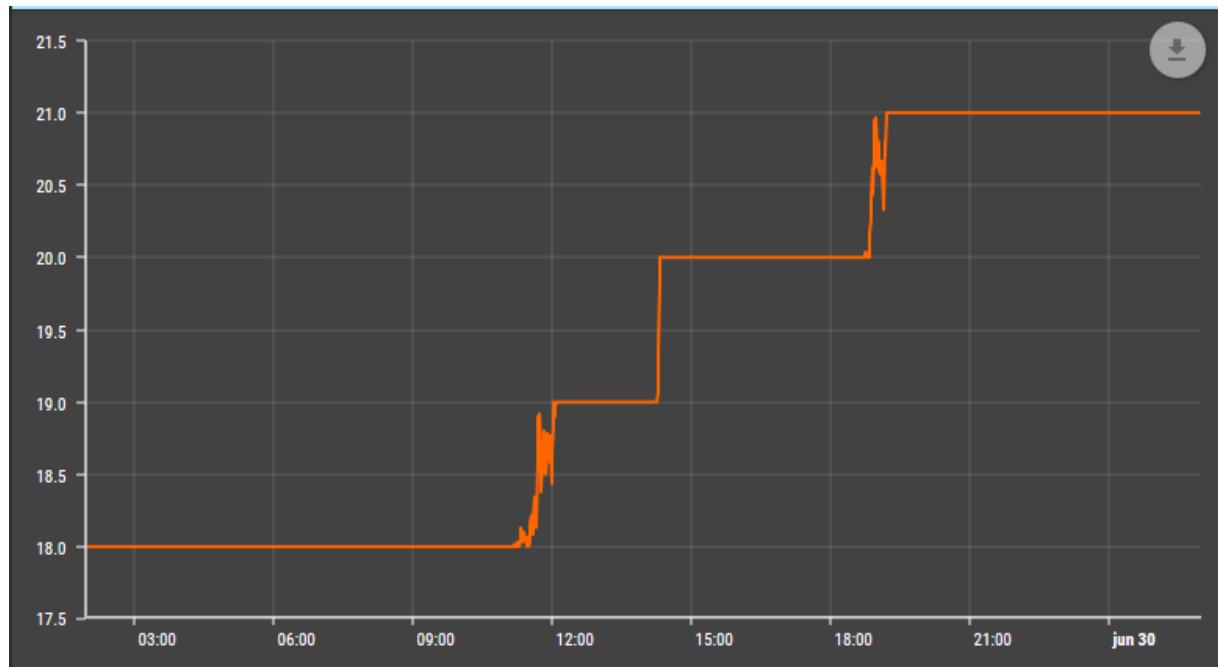
Figura 3.52 – Histórico de valores do *Data Point* referente a fechadura eletrônica.

Point value history C		Configure
Time ↓	Point value	
29/6/2019 19:08:55	FECHADO	
29/6/2019 19:08:53	ABERTO	
29/6/2019 13:26:19	FECHADO	
29/6/2019 13:26:15	ABERTO	
29/6/2019 06:57:15	FECHADO	
29/6/2019 06:57:11	ABERTO	
27/6/2019 23:10:24	FECHADO	
27/6/2019 23:10:22	ABERTO	
27/6/2019 23:10:20	FECHADO	
27/6/2019 23:10:18	ABERTO	

Fonte: Autor.

É possível também, conforme o tipo de dado, a visualização através de gráfico, como na Figura 3.53.

Figura 3.53 – Gráfico da temperatura.



Fonte: Autor.

4 CONCLUSÃO

O uso da automação residencial está crescendo nos últimos anos com a popularização da IoT e a redução nos preços dos dispositivos dedicados a esses sistemas. Nos dias de hoje, a automação residencial não é vista somente em residências de luxo, em lares de classe média já são encontrados esse tipo de sistema. Os lares brasileiros estão se adequando às novas tecnologias que trazem mais conforto e segurança. Porém, há a necessidade de novas instalações para a implementação do sistema.

Nesse contexto, foi apresentado um sistema supervisório destinado ao uso com dispositivos de automação residencial. Esse sistema visa gerenciar diversas soluções comuns em automação residencial. Ainda, esse sistema supervisório visa facilitar o controle de diversos dispositivos, a partir de um só lugar. Além de facilitar o controle, isso facilita a instalação. Como o sistema supervisório pode integrar diversos dispositivos, isso dispensa o uso de um controlador único ligado a vários sensores e atuadores, diminuindo o uso de cabos.

As soluções para automação residencial, geridas pelo sistema supervisório proposto, foram desenvolvidas utilizando as plataformas Arduino e ESP8266. As soluções desenvolvidas tratam-se de soluções simples, porém facilmente adaptáveis e escaláveis. Como o sistema para acionamento de lâmpadas, a mesma solução pode ser aplicado em qualquer dispositivo que seja do tipo on/off. O sistema desenvolvido para o acionamento de ar condicionado pode ser aplicado para qualquer outro acionamento através de infravermelho, somente há a necessidade da clonagem do código enviado pelo controle original do dispositivo.

Para a aplicação foi utilizado o *software Mango Automation* por se tratar de um *software* gratuito para fins não comerciais, que cumpre as funções necessárias. Apresenta uma interface simples e de fácil utilização. Como ponto forte, é importante destacar a grande possibilidade de personalização. É possível não só adicionar novos elementos, mas também editar os já presentes, alterar tamanhos de fonte, cores, estilos. Há a possibilidade de inserir novos menus, criar listas de exibição de menus diferentes para os usuários.

Visto que as soluções implementadas são soluções gerais, elas podem ser adaptadas e aplicadas a diferentes equipamentos. Para adicionar mais dispositivos ao sistema basta replicar os procedimentos realizados com os dispositivos presentes. Ainda com o uso de suas funcionalidades do *Mango Automation*, como os *Point Links* e o uso do *Advanced Scheduler*, é possível transformar a casa em uma residência inteligente, trazendo a possibilidade do próprio sistema realizar ações de forma autônoma.

Desta maneira, tanto os objetivos gerais como os objetivos específicos foram alcançados. O sistema desenvolvido é funcional, robusto e escalável. Os resultados obtidos consolidam os conhecimentos adquiridos ao longo do curso, principalmente nas áreas de dispositivos lógicos e programáveis, automação e algoritmos.

4.1 TRABALHOS FUTUROS

As soluções em automação e o sistema desenvolvido atingiram resultado satisfatório para a aplicação desejada. No entanto, algumas melhorias podem ser feitas.

Uma das possíveis melhorias pode ser feita na programação dos controladores. Nos módulos da família ESP8266, o endereço de IP não é fixo. Conforme as configurações do roteador, os módulos conectados podem, em algum momento, trocar o endereço IP, logo será necessário alterar nas configurações de *Data Sources* no *Mango Automation*.

Outro ponto que pode ser mais explorado, é as possibilidades de programação em JavaScript no *Mango Automation*. Em projetos maiores, com a presença de mais sensores e atuadores, é possível criar combinações de condições para os acionamentos, como por exemplo, determinar um horário e outra condição para determinado acionamento.

Ainda é possível realizar um estudo mais aprofundado de funcionalidades do *Mango Automation*. Por tratar-se de um *software* de grande potencial e constantemente atualizado, já possui ferramentas para criação de relatórios e grandes bancos de dados (*Big Data*). Assim como seu uso para um acesso remoto e não somente na mesma rede da residência em uso.

REFERÊNCIAS

- ACCARDI, A.; DODONOV, E. Automação Residencial: Elementos Básicos, Arquiteturas, Setores, Aplicações e Protocolos. **Revista T.I.S.**, v. 2, n. 2, p. 2316–2872, 2012. ISSN 1949-243X.
- ADVANCED MONOLITHIC. **AMS 1117**: Folha de especificações. [S.l.]: Advanced Monolithic Systems, Inc, 2014.
- AMAZING ARCHTECTURE. **Studio style | Neorama - Floor Plan**. [S.l.], 2017. Disponível em: <<https://www.instagram.com/p/BPFNKHol5si/>>.
- APPLE. **FIBARO for iPhone na App Store**. [S.l.], 2019. Disponível em: <<https://apps.apple.com/br/app/fibaro-for-iphone/id511085659>>.
- ARDUINO. **ARDUINO UNO REV3**. [S.l.], 2018. Disponível em: <<https://store.arduino.cc/usa/arduino-uno-rev3>>.
- _____. **Arduino - Introduction**. [S.l.], 2019. Disponível em: <<https://www.arduino.cc/en/Guide/Introduction>>.
- AUGUSTO, H. **Sistema Supervisório: "Entenda o que é**. [S.l.], 2015. Disponível em: <<https://www.hitecnologia.com.br/blog/sistema-supervisorio-entenda-o-que-e/>>.
- AURESIDE. **A Automação Residencial alavanca a demanda por eficiência**. [S.l.], 2015. Disponível em: <<http://www.aureside.org.br/noticias/a-automacao-residencial-alavanca-a-demanda-por-eficiencia>>.
- _____. **Automação Residencial: demanda na Construção Civil**. [S.l.], 2015. Disponível em: <<http://www.aureside.org.br/noticias/automacao-residencial--demanda-na-construcao-civil>>.
- AUTOCORE ROBÓTICA. **Conversor Usb-Serial Ttl PL2303**. [S.l.], 2019. Disponível em: <<https://www.autocorerobotica.com.br/produto/adaptador-usb-serial-ttl-pl2303.html>>.
- BASTOS, S. A. **ScadaBR - Arduinos - Influência entre as portas analógicas**. [S.l.], 2014. Disponível em: <<http://labdegaragem.com/forum/topics/scadabr-arduinoss-influencia-entre-as-portas-anal-gicas>>.
- BAÚ DA ELETROÔNICA. **Adaptador ESP8266 ESP-01 Para DIP**. [S.l.], 2019. Disponível em: <<http://www.baudaelectronica.com.br/adaptador-esp8266-esp-01-para-dip.html>>.
- BOLZANI, C. A. M. **Residências Inteligentes**. [S.l.]: Livraria da Física, 2004.
- BROADLINK TECHNOLOGY. **App**. [S.l.], 2018. Disponível em: <<https://www.ibroadlink.com/app>>.

_____. **Smart Plug.** [S.l.], 2018. Disponível em: <<https://www.ibroadlink.com/products/smart-plug>>.

_____. **Smart Wall Switch.** [S.l.], 2018. Disponível em: <<https://www.ibroadlink.com/products/smart-wall-switch>>.

_____. **Universal Remote.** [S.l.], 2018. Disponível em: <<https://www.ibroadlink.com/products/ir+rf>>.

DANEELS, A.; SALTER, W. What is SCADA? **International Conference on Accelerator and Large Experimental Physics Control Systems**, 1999.

ELETRODEX. **Módulo Receptor Infravermelho 38Khz IRM-3638T.** [S.l.], 2018. Disponível em: <<https://www.eletrodex.com.br/modulo-receptor-infravermelho-38khz-irm-3638t.html>>.

ELETROGATE. **Led Infravermelho TSAL6200 5mm.** [S.l.], 2019. Disponível em: <<https://www.eletrogate.com/led-infravermelho-tsal6200-5mm>>.

ELIPSE. **Elipse E3.** [S.l.], 2015. Disponível em: <<https://www.elipse.com.br/produto/elipse-e3>>.

_____. **Elipse Mobile.** [S.l.], 2015. Disponível em: <<https://www.elipse.com.br/produto/elipse-mobile>>.

EPCOS. **NTC thermistors for temperature measurement Series B572**v2:** Folha de especificações. [S.l.]: EPCOS, a TDK company, 2006.

ESPRESSIF IOT TEAM. **ESP8266EX:** Folha de especificações. [S.l.]: ESPRESSIF System, 2018.

EVANS, D. **A Internet das Coisas Como a próxima evolução da Internet está mudando tudo.** [S.l.], 2011. Disponível em: <https://www.cisco.com/c/dam/global/pt_br/assets/executives/pdf/internet_of_things_iot_ibsg_0411final.pdf>.

FIBAR GROUP. **Dimmer 2.** [S.l.], 2019. Disponível em: <<https://www.fibaro.com/pt/products/dimmer-2/>>.

_____. **FAQ FIBARO Z-Wave Home Automation System.** [S.l.], 2019. Disponível em: <<https://manuals.fibaro.com/faq-zw/>>.

_____. **Home Center 2.** [S.l.], 2019. Disponível em: <<https://www.fibaro.com/pt/products/home-center-2/>>.

_____. **Home Center 2 Fibaro Manuals.** [S.l.], 2019. Disponível em: <<https://manuals.fibaro.com/home-center-2/>>.

_____. **Roller Shutter 3.** [S.l.], 2019. Disponível em: <<https://www.fibaro.com/pt/products/smart-roller-shutter/>>.

- _____. **Sistemas Sportados.** [S.l.], 2019. Disponível em: <<https://www.fibaro.com/pt/compatibility/>>.
- _____. **Switches.** [S.l.], 2019. Disponível em: <<https://www.fibaro.com/pt/products/switches/>>.
- _____. **Wall Plug.** [S.l.], 2019. Disponível em: <<https://www.fibaro.com/pt/products/wall-plug/>>.
- FILIPEFLOP. Módulo Relé 5V 1 Canal.** [S.l.], 2018. Disponível em: <<https://www.filipeflop.com/produto/modulo-rele-5v-1-canal>>.
- _____. **Módulo Wifi ESP8266 ESP-01.** [S.l.], 2018. Disponível em: <<https://www.filipeflop.com/produto/modulo-wifi-esp8266-esp-01>>.
- _____. **Adaptador USB para Módulo WiFi ESP8266 ESP-01.** [S.l.], 2019. Disponível em: <<https://www.filipeflop.com/produto/adaptador-usb-para-modulo-wifi-esp8266-esp-01>>.
- _____. **Kit Módulo Leitor Rfid Mfrc522 Mifare.** [S.l.], 2019. Disponível em: <<https://www.filipeflop.com/produto/sensor-de-temperatura-ds18b20-a-prova-dagua>>.
- _____. **Mini Trava Elétrica Solenoide 12V.** [S.l.], 2019. Disponível em: <<https://www.filipeflop.com/produto/mini-trava-eletrica-solenoide-12v>>.
- _____. **Sensor de Temperatura Digital DS18B20.** [S.l.], 2019. Disponível em: <<https://www.filipeflop.com/produto/sensor-de-temperatura-digital-ds18b20>>.
- _____. **Sensor de Temperatura DS18B20 a Prova D'água.** [S.l.], 2019. Disponível em: <<https://www.filipeflop.com/produto/sensor-de-temperatura-ds18b20-a-prova-dagua>>.
- FORCELINE. Sensor de Presença de Parede Manual de Instruções:** Manual de instruções. [S.l.]: Force Line Indústria e Comércio de Componentes Eletrônicos., 2018.
- FREIRE, R. Geladeira, lavadora e mais eletrodomésticos deixam sua casa smart.** [S.l.], 2018. Disponível em: <<https://www.techtudo.com.br/noticias/2018/07/geladeira-lavadora-e-mais-eletrodomesticos-deixam-sua-casa-smart.ghtml>>.
- GASTALDI, B. Automação Residencial Usando Microcontroladores de Estado da Arte em Programação Voltada à Internet.** 2008.
- GOOGLE PLAY. e-Control.** [S.l.], 2019. Disponível em: <https://play.google.com/store/apps/details?id=com.broadlink.rmt&hl=pt_BR>.
- GUETTE, L. How To Develop Custom Views With AngularJS Components In Mango Automation.** [S.l.], 2018. Disponível em: <<https://medium.com/typeiqs/how-to-develop-custom-views-with-angularjs-components-in-mango-automation-251e6b74857>>.

HI Tecnologia. **Introdução HIscada Pro.** [S.l.], 2018. Disponível em: <https://doc.hitecnologia.com.br/hiscada_pro/_files_overview/introduction.html>.

INFINITE AUTOMATION. **Mango Automation:** Folha de especificações. [S.l.]: Infinite Automation Systems, 2019.

INJETEL. **Manual Sensor de Presença Para Teto:** Manual de instruções. [S.l.]: Injetel Indústria e comércio de componentes plásticos, 2018.

INVENT ELECTRONICS. **Light Dependent Resistor (LDR).** [S.l.], 2018. Disponível em: <<https://www.inventelectronics.com/product/light-dependent-resistor>>.

ITEAD INTELLIGENT SYSTEMS. **EWeLink.** [S.l.], 2016. Disponível em: <<https://sonoff.itead.cc/en/ewelink>>.

_____. **SONOFF Basic.** [S.l.], 2016. Disponível em: <<https://sonoff.itead.cc/en/products/sonoff/sonoff-basic>>.

_____. **SONOFF TH10/TH16.** [S.l.], 2016. Disponível em: <<https://sonoff.itead.cc/en/products/sonoff/sonoff-th>>.

JALOUDI, S. Communication protocols of an industrial internet of things environment: A comparative study. **Future Internet**, v. 11, 03 2019.

JURIZATO, L. A.; PEREIRA, P. S. R. Sistemas Supervisórios. **Network Technologies**, v. 1/2, n. 1/2, p. 105–114, 2003. ISSN 1677-7778.

MADAKAM, S.; RAMASWAMY, R.; TRIPATHI, S. Internet of Things (IoT): A Literature to review. **Journal of Computer and Communications**, 2015. Disponível em: <http://file.scirp.org/pdf/JCC_2015052516013923.pdf>.

MASTER WALKER SHOP. **Base Adaptador para NodeMCU V3.** [S.l.], 2019. Disponível em: <<https://www.masterwalkershop.com.br/base-adaptador-para-nodemcu-v3>>.

_____. **NodeMCU v3 Lolin WiFi ESP8266 ESP-12E.** [S.l.], 2019. Disponível em: <<https://www.masterwalkershop.com.br/nodemcu-v3-lolin-wifi-esp8266-esp-12e>>.

MAXIM INTEGRATED. **DS18B20 Programmable Resolution 1-Wire Digital Thermometer:** Folha de especificações. [S.l.]: Maxim Integrated Products, Inc., 2018.

MODBUS ORGANIZATION. **MODBUS Application Protocol Specification V1.1b3:** Folha de especificações. [S.l.]: Modbus Organization, Inc., 2019.

MORAES, C. C.; CASTRUCCI, P. L. **Engenharia de Automação Industrial.** [S.l.]: LTC, 2010.

MOTT, A. **O que são Sistemas Supervisórios**, 2012. Disponível em: <<https://www.automacaoindustrial.info/o-que-sao-sistemas-supervisorios/>>.

- NXP SEMICONDUCTORS. **MFRC522 Standard performance MIFARE and NTAG frontend**: Folha de especificações. [S.l.]: NXP Semiconductors N.V., 2016.
- PELLENZ, A. **Sistema de Automação Residencial**. Relatório de estágio — Universidade de Caxias do Sul - UCS, 2017.
- PHILIPS SEMICONDUCTORS. **Triacs BT137 series**: Folha de especificações. [S.l.]: Philips Electronics, 1997.
- PROSOFT TECHNOLOGY. **Modbus & Modbus TCP Protocol Solutions**. [S.l.], 2019. Disponível em: <<https://br.prosoft-technology.com/Landing-Pages/Protocol/Modbus-and-Modbus-TCP-Protocol#&slider1=2>>.
- RANGER, S. **What's the IoT? Everything you need to know about the internet of things right now**. [S.l.], 2018. Disponível em: <<https://www.zdnet.com/article/what-is-the-internet-of-things-everything-you-need-to-know-about-the-iot-right-now>>.
- RAPID. **Termocouples and Thermistors**: Folha de especificações. [S.l.]: Rapid, 2007.
- ROGGIA, L.; FUENTES, R. C. **Automação Industrial**. [S.l.]: Rede e-Tec Brasil, 2016.
- ROURE, M. D. **Pirâmide da Automação Industrial Entenda de uma vez por todas**. [S.l.], 2018. Disponível em: <<https://instrumentacaoecontrole.com.br/piramide-da-automacao-industrial>>.
- SENSORWEB. **ScadaBR**. [S.l.], 2018. Disponível em: <<http://www.scadabr.com.br>>.
- SUNROM. **Light Dependent Resistor - LDR**: Folha de especificações. [S.l.]: Sunrom Technologies, 2008.
- SURESH, P. et al. A state of the art review on the Internet of Things (IoT) history, technology and fields of deployment. **International Conference on Science, Engineering and Management Research (ICSEMR)**, 2014. Disponível em: <<https://ieeexplore.ieee.org/abstract/document/7043637>>.
- THOMSEN, A. **Como comunicar com o Arduino Ethernet Shield W5100**. [S.l.], 2014. Disponível em: <<https://www.filipeflop.com/blog/tutorial-ethernet-shield-w5100>>.
- USINAINFO. **Módulo Dimmer para Arduino / Pic MC-8A com sinal Zero Cross**. [S.l.], 2018. Disponível em: <<https://www.usinainfo.com.br/dimmer-arduino/modulo-dimmer-para-arduino-pic-mc-8a-com-sinal-zero-cross-2799.html>>.
- _____. **Adaptador ESP8266 ESP-01 com Relé 5V 10A 1 Canal**. [S.l.], 2019. Disponível em: <<https://www.usinainfo.com.br/esp8266/adaptador-esp8266-esp-01-com-rele-5v-10a-1-canal-5149.html>>.
- VAISALA. **YL-83 Rain Detector**: Folha de especificações. [S.l.]: Vaisala, 2015.

WIZNET. **W5100 Datasheet**: Folha de especificações. [S.l.]: WIZnet Co, 2008.

ZURAWSKI, R. **Industrial Communication Technology Handbook**. [S.l.]: CRC Press, 2015.

APÊNDICES

Apêndice A – Código Arduino

Este é o código implementado no Arduino responsável pelo sensoriamento. Possui as funções de realizar as leituras de sensores e envia-lás para o sistema supervisório, assim como recebe o valor de luminosidade da lâmpada dimmerizada e envia o sinal para o módulo NodeMCU ligado ao Dimmer.

```

/*
Modbus-Arduino Example - TempSensor (Modbus IP)
Copyright by Andre Sarmento Barbosa
http://github.com/andresarmento/modbus-arduino
*/
#include <SPI.h>
#include <Ethernet.h>
#include <Modbus.h>
#include <ModbusIP.h>
#include <OneWire.h>
#include <DallasTemperature.h>

OneWire pino(3);
DallasTemperature barramento(&pino);
DeviceAddress sensor;

//Modbus Registers Offsets (0-9999)
const int TEMP_IREG = 0;
const int LDR_IREG = 1;
const int YLA_IREG = 3;
const int YLD_IREG = 4;
const int DIMMER_IREG = 5;
const int LUM_HREG = 0;

//Used Pins
const int LDRPin = A0;
const int YLAPin = A2;
const int YLDPin = 5;
const int PINO_DIM = 9;

//ModbusIP object
ModbusIP mb;
int luminosidade=0;
int luminosidadeLDR;

```

```
int escuro;
int chuva;
int chuva_intens;
float temperatura = 0;
int PWM;
int PWM3;

void setup() {
    pinMode(A0, INPUT);
    pinMode(A1, INPUT);
    pinMode(A2, INPUT);
    pinMode(4, INPUT);
    pinMode(9, OUTPUT);
    barramento.begin();
    barramento.getAddress(sensor, 0);

    // The media access control (ethernet hardware) address for the shield
    byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
    // The IP address for the shield
    byte ip[] = { 192, 168, 0, 108 };
    //Config Modbus IP
    mb.config(mac, ip);

    mb.addIreg(TEMP_IREG);
    mb.addIreg(LDR_IREG);
    mb.addIreg(POT_IREG);
    mb.addIreg(YLA_IREG);
    mb.addIreg(YLD_IREG);
    mb.addIreg(DIMMER_IREG);
    mb.addHreg(LUM_HREG);
}

void loop() {
    //Call once inside loop() - all magic here
    mb.task();

    barramento.requestTemperatures();
    temperatura = barramento.getTempC(sensor);
    luminosidadeLDR=map(analogRead(A0),0,1024,0,100);

    if(luminosidadeLDR<50){
        escuro=0;
```

```
 }else{
    escuro=1;
}

chuva_intens=map(analogRead(A2),0,1024,100,0);

if(digitalRead(YLDPin)==1){
    chuva=0;
}else{
    chuva=1;
}

if (mb.Hreg(LUM_HREG)!=luminosidade) {
    luminosidade=mb.Hreg(LUM_HREG);
}

PWM=map(luminosidade,0,100,0,255);

if(PWM>255){
    PWM=255;
}
if(PWM<0){
    PWM=0;
}
PWM3=(PWM*0.66);

analogWrite(PINO_DIM,PWM3);

if(luminosidade>1){
    mb.Ireg(DIMMER_IREG, 1);
}else{
    mb.Ireg(DIMMER_IREG, 0);
}

mb.Ireg(TEMP_IREG, temperatura);
mb.Ireg(LDR_IREG, escuro);
mb.Ireg(YLD_IREG, chuva);

delay(100);
}
```

Apêndice B – Código Acionamento ON/OFF

Esse é o código implementado em módulos ESP-01 com a finalidade de acionamentos do tipo On/Off. Possui somente uma variável em comunicação com o sistema supervisório a fim de acionar o módulo relé.

```

/*
Modbus-Arduino Example - Test Led (Modbus IP ESP8266)
Control a Led on GPIO0 pin using Write Single Coil Modbus Function
Copyright by Andre Sarmento Barbosa
http://github.com/andresarmento/modbus-arduino
*/
#include <ESP8266WiFi.h>
#include <Modbus.h>
#include <ModbusIP_ESP8266.h>

//Modbus Registers Offsets (0-9999)
const int LED_COIL = 0;
//Used Pins
const int ledPin = 0; //GPIO0

//ModbusIP object
ModbusIP mb;

void setup() {

    mb.config("123456789", "qweasdzc");
    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
    }
    pinMode(ledPin, OUTPUT);
    mb.addCoil(LED_COIL);
}

void loop() {
    //Call once inside loop() - all magic here
    mb.task();

    if (mb.Coil(LED_COIL)==1){
        digitalWrite(ledPin, 0);
    }
}

```

```
if (mb.Coil(LED_COIL)==0){  
    digitalWrite(ledPin, 1);  
}  
}
```

Apêndice C – Código Dimmer

Esse é o código implementado no módulo NodeMCU responsável pelo controle do módulo Dimmer. Possui a função de realizar uma leitura analógica e a partir dela define o nível de luminosidade com o módulo Dimmer.

```
#define PINO_ZC 5 //D1
#define PINO_DIM 4 //D2
#define DIM_IN A0

volatile long luminosidade = 0; // 0 a 100
int valordim = 0;
int i = 0;

void zeroCross() {
    if (luminosidade>100) luminosidade=100;
    if (luminosidade<0) luminosidade=0;
    long t1 = 8200L * (100L - luminosidade) / 100L;
    delayMicroseconds(t1);
    digitalWrite(PINO_DIM, HIGH);
    delayMicroseconds(6); // t2
    digitalWrite(PINO_DIM, LOW);
}

void setup() {
    pinMode(PINO_DIM, OUTPUT);
    attachInterrupt(5, zeroCross, RISING);
}

void loop() {
    valordim = analogRead(DIM_IN);
    luminosidade = map(valordim,0,1023,0,95);
}
```

Apêndice D – Código Ar Condicionado

Este é o código implementado em módulo ESP-01 para acionamento de ar condicionado. Este possui a função de receber os dados de temperatura ambiente e temperatura que o ar condicionado deve ligar e desligar. A partir desses dados ele realiza a comparação das temperaturas, liga ou desliga o ar condicionado e retorna para o sistema supervisório se o estado do equipamento.

```
564, 616, 560, 588, 564, 588, 560, 588, 564, 616, 560, 588, 560, 588, 564,
1696, 564, 616, 560, 588, 564, 584, 564, 588, 560, 616, 564, 588, 560,
1696, 564, 612, 540, 616, 560, 588, 560, 592, 560, 584, 564, 1724, 564,
588, 564, 588, 560, 588, 564, 612, 564, 588, 560, 592, 560, 588, 560, 616,
560, 588, 564, 588, 560, 588, 564, 616, 560, 584, 564, 592, 560, 588, 560,
580, 564};

uint16_t desliga[99] = {8868, 4532, 520, 1740, 520, 1740, 520, 656, 496, 656,
520, 628, 520, 632, 520, 656, 536, 1720, 524, 632, 520, 1736, 524, 1736,
524, 656, 520, 632, 516, 632, 520, 628, 524, 656, 520, 628, 520, 632, 520,
1736, 524, 656, 520, 628, 524, 632, 516, 632, 520, 656, 520, 632, 516,
1740, 524, 628, 520, 656, 520, 628, 524, 652, 496, 632, 520, 1764, 524,
628, 520, 656, 496, 628, 520, 656, 520, 656, 496, 628, 520, 628, 524, 652,
524, 652, 496, 628, 524, 628, 520, 656, 524, 628, 520, 628, 524, 624, 524,
644, 496};

void setup() {

    irsend.begin();

    mb.config("123456789", "qweasdzc");

    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
    }

    mb.addHreg(TEMP_HREG);
    mb.addHreg(LIGA_HREG,40);
    mb.addHreg(DESLIGA_HREG,30);
    mb.addIsts(LIGADO_ISTS);
}

void loop() {
    //Call once inside loop() - all magic here
    mb.task();

    temp=mb.Hreg(TEMP_HREG);

    SPliga=mb.Hreg(LIGA_HREG);
    SPdesliga=mb.Hreg(DESLIGA_HREG);

    mb.Ists(LIGADO_ISTS, ligado);
}
```

```
if (temp>SPliga and ligado==0){  
    irsend.sendRaw(liga, 67, 38); // Send a raw data capture at 38kHz.  
    ligado=1;  
}  
if (temp<SPdesliga and ligado==1){  
    irsend.sendRaw(desliga, 67, 38); // Send a raw data capture at 38kHz.  
    ligado=0;  
}  
}
```

Apêndice E – Código Fechadura

Este é o código implementado em módulo NodeMCU para o controle da fechadura. Este realiza a leitura do cartão/chaveiro aproximado do sensor e envia para o sistema supervisório se o cartão/chaveiro lido é válido ou não. Ainda faz o acionamento da fechadura eletrônica.

```
#include <Wire.h> //INCLUSAO DA BIBLIOTECA NECESSARIA
#include <SPI.h> //INCLUSAO DE BIBLIOTECA
#include <MFRC522.h> //INCLUSAO DE BIBLIOTECA
#include <ESP8266WiFi.h>
#include <Modbus.h>
#include <ModbusIP_ESP8266.h>

#define SS_PIN 2 // D4 - PINO SDA
#define RST_PIN 0 // D3 - PINO DE RESET

MFRC522 rfid(SS_PIN, RST_PIN); //PASSAGEM DE PARAMETROS REFERENTE AOS PINOS

const int pinoLedVerde = D1; //PINO DIGITAL REFERENTE AO LED VERDE
const int pinoLedVermelho = D2; //PINO DIGITAL REFERENTE AO LED VERMELHO
const int pinoReleFechadura = D8; //PINO DIGITAL REFERENTE A FECHADURA

const int CERTO_ISTS = 0;
const int ERRADO_ISTS = 1;
int cartao_lido = 0;
volatile int certo = 0;
volatile int errado = 0;
long ts;
long reset_ts;

ModbusIP mb;

void setup(){
    Wire.begin(); //INICIALIZA A BIBLIOTECA WIRE
    SPI.begin(); //INICIALIZA O BARRAMENTO SPI
    rfid.PCD_Init(); //INICIALIZA MFRC522

    mb.config("123456789", "qweasdzc");
}

while (WiFi.status() != WL_CONNECTED) {
```

```
delay(500);
}

mb.addIsts(CERTO_ISTS, 0);
mb.addIsts(ERRADO_ISTS, 0);

pinMode(pinoLedVerde, OUTPUT); //DEFINE O PINO COMO SAIDA
pinMode(pinoLedVermelho, OUTPUT); //DEFINE O PINO COMO SAIDA
pinMode(pinoReleFechadura, OUTPUT); //DEFINE O PINO COMO SAIDA

digitalWrite(pinoLedVerde, LOW); //LED INICIA DESLIGADO
digitalWrite(pinoLedVermelho, LOW); //LED INICIA DESLIGADO
digitalWrite(pinoReleFechadura, LOW); //FECHADURA INICIA TRANCADA
reset_ts=0;
}

void loop() {
    mb.task();
    if (millis()>reset_ts){
        reset_ts=millis();
    }else{
        reset_ts=0;
        ts=0;
    }
    if (certo==0 and errado==0){
        leituraRfid(); //CHAMA A FUNCAO RESPONSABEL PELA VALIDACAO DA TAG RFID
    }else{
        if(cartao_lido==0){
            ts=millis();
            cartao_lido=1;
        }
        if (millis()>(ts+3000) and certo==1){
            certo=0;
            cartao_lido=0;
        }
        if (millis()>(ts+3000) and errado==1){
            errado=0;
            cartao_lido=0;
        }
    }
}
```

```
digitalWrite(pinoReleFechadura, certo);
digitalWrite(pinoLedVerde, certo);
digitalWrite(pinoLedVermelho, errado);
mb.Ists(CERTO_ISTS, certo);
mb.Ists(ERRADO_ISTS, errado);
delay(200);
}

//FUNCAO DE VALIDACAO DA TAG RFID
void leituraRfid(){
    if (!rfid.PICC_IsNewCardPresent() || !rfid.PICC_ReadCardSerial()) //VERIFICA
        SE O CARTAO PRESENTE NO LEITOR E DIFERENTE DO ULTIMO CARTAO LIDO. CASO
        NAO SEJA, FAZ
    return; //RETORNA PARA LER NOVAMENTE
/**INICIO BLOCO DE CODIGO RESPONSAVEL POR GERAR A TAG RFID LIDA***/
String strID = "";
for (byte i = 0; i < 4; i++) {
    strID +=
        (rfid.uid.uidByte[i] < 0x10 ? "0" : "") +
        String(rfid.uid.uidByte[i], HEX) +
        (i!=3 ? ":" : "");
}
strID.toUpperCase();
/**FIM DO BLOCO DE CODIGO RESPONSAVEL POR GERAR A TAG RFID LIDA***/
//O ENDERECHO "2B:7C:96:BB" DEVERA SER ALTERADO PARA O ENDERECHO DA SUA TAG
//RFID QUE CAPTUROU ANTERIORMENTE
if (strID.indexOf("2B:7C:96:BB") >= 0) { //SE O ENDERECHO DA TAG LIDA FOR
    IGUAL AO ENDERECHO INFORMADO, FAZ
    certo=1;
}else{ //SENAO, FAZ (CASO A TAG LIDA NAO SEJA VALIDA)
    errado=1;
}
rfid.PICC_HaltA(); //PARADA DA LEITURA DO CARTAO
rfid.PCD_StopCrypto1(); //PARADA DA CRIPTOGRAFIA NO PCD
}
```
