

INSTITUTO FEDERAL DE SANTA CATARINA

KLEISON ANTONIO POSSENTI

PROJETO DE CONTROLE E AUTOMAÇÃO RESIDENCIAL ATRAVÉS DO
MONITORAMENTO DE UMA PÁGINA WEB

Joinville

2022

KLEISON ANTONIO POSSENTI

PROJETO DE CONTROLE E AUTOMAÇÃO RESIDENCIAL ATRAVÉS DO
MONITORAMENTO DE UMA PÁGINA WEB

Monografia apresentada ao
Bacharelado em
Engenharia Elétrica do
Câmpus Joinville do
Instituto Federal de Santa
Catarina para a obtenção do
diploma de Bacharel em
Engenharia Elétrica

Orientador: Nivaldo
Theodoro Schiefler Junior

Joinville

2022

Possenti, Kleison Antonio.

Projeto de controle e automação residencial através do monitoramento de uma página WEB / Kleison Antonio Possenti. – Joinville, SC, 2022.

113 p.

Trabalho de Conclusão de Curso (Graduação) - Instituto Federal de Educação Ciência e Tecnologia de Santa Catarina, Curso de Bacharelado em Engenharia Elétrica, Joinville, 2022.

Orientador: Nivaldo Theodoro Schiefler Junior, Dr. Eng.

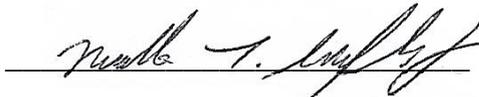
1. Automação Residencial. 2. Domótica. 3. Microcontrolador. I. Instituto Federal de Educação Ciência e Tecnologia de Santa Catarina. II. Título.

KLEISON ANTONIO POSSENTI

PROJETO DE CONTROLE E AUTOMAÇÃO RESIDENCIAL ATRAVÉS DO
MONITORAMENTO DE UMA PÁGINA WEB

Este trabalho foi julgado adequado para obtenção do título em Bacharel em Engenharia Elétrica, pelo Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina, e aprovado na sua forma final pela comissão avaliadora abaixo indicada.

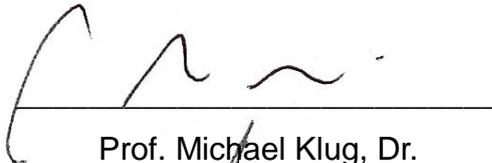
Joinville, 22/02/2022.



Prof. Nivaldo Theodoro Schiefler Junior, Dr.

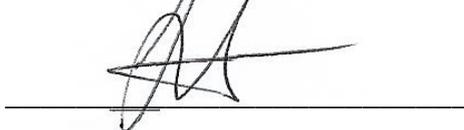
Orientador

Instituto Federal de Santa Catarina



Prof. Michael Klug, Dr.

Instituto Federal de Santa Catarina



Prof. Stefano Romeu Zeplin, Me.

Instituto Federal de Santa Catarina

Dedico este trabalho a Deus, a meu pai, a minha mãe, a minha irmã, meus avós
amigos e professores que acreditaram e apoiaram
durante todo o desenvolvimento
deste sonho.

AGRADECIMENTOS

Primeiramente a Deus, por me dar saúde, sabedoria, inteligência, entendimento, paciência e permitir realizar este trabalho de conclusão de curso e finalizar este curso de Engenharia Elétrica que é mais um sonho da minha vida.

Aos meus pais, por sempre ajudar, apoiar, acreditar e incentivar durante toda a jornada da vida, principalmente durante o período do curso de engenharia, fazendo com que eu nunca desistisse e seguisse perseverante na busca dos objetivos.

Ao Instituto Federal de Santa Catarina, que me deu a oportunidade de me tornar engenheiro eletricitista, além de fornecer grande parte do material científico para elaborar esse trabalho.

Ao professor Dr. Michael Klug por auxiliar na elaboração de algumas etapas do trabalho, além de sanar dúvidas referente ao desenvolvimento do trabalho.

Ao professor Msc. Stefano Romeu Zeplin pelos diversos materiais fornecidos sobre redes de computadores e todo o apoio no desenvolvimento dessa etapa.

Ao professor Dr. Nivaldo Theodoro Schiefler Junior por orientar o trabalho, fornecendo as diretrizes necessárias para atingir os objetivos.

“Nunca deixe ninguém te dizer que você não pode fazer alguma coisa.
Se você tem um sonho, tem que correr atrás dele,
As pessoas não conseguem vencer e dizem
que você também não vai vencer.
Se quer alguma coisa,
corra atrás.”

(Passagem do Filme: À Procura da Felicidade, 2006)

RESUMO

A automação residencial ou domótica pode ser definida como a tecnologia que permite efetuar uma dada tarefa de forma automática em um ambiente residencial ou doméstico. Sua implementação permite ao usuário um maior conforto e segurança em sua residência. Apesar disso, seu uso ainda é pouco explorado e pouco presente nas casas brasileiras, devido ao elevado custo dos sistemas apresentados no mercado. Em vista disso, o presente trabalho demonstra o desenvolvimento do projeto de controle e automação do sistema de iluminação e alarme de uma residência, através do monitoramento de uma página WEB via acesso externo, utilizando um servidor DNS dinâmico gratuito. Para isso, é utilizado um microcontrolador PIC16F877A para atuar como uma central de controle e o módulo ESP-8266 para criar uma rede Wi-Fi e permitir o microcontrolador de se comunicar com o usuário através de outros dispositivos. O sistema conta com monitoramento via página WEB para controle a distância e também uma central confeccionada em placa de circuito impresso, composta por um *display* LCD, LED's e *push buttons*, capazes de efetuar o controle de modo manual. Para o desenvolvimento do projeto foi utilizado a linguagem de programação C e o *software* de programação MPLAB X IDE. Também foram realizadas simulações do projeto no *software* Proteus e testes do sistema de iluminação, alarme e comunicação em *protoboard*. Por fim, foram realizados testes via placa de circuito impresso e via página WEB e verificou-se que o projeto desenvolvido possibilita o controle do sistema de iluminação e alarme de forma remota e presencial de modo satisfatório, além de ser um sistema de baixo custo e fácil aplicação para uma residência.

Palavras-Chave: Automação Residencial. Domótica. Microcontrolador. Programação. Página WEB.

ABSTRACT

Home automation or domotic can be defined as the technology that allows you to perform a given task automatically in a residential or home environment. Its implementation allows the user greater comfort and safety at home. Despite this, its use is still little explored and little present in Brazilian homes, due to the high cost of the systems presented in the market. In view of this, the present work demonstrates the development of the control and automation project of the lighting and alarm system of a residence, through the monitoring of a WEB page via external access, using a free dynamic DNS server. For this, a PIC16F877A microcontroller is used to act as a control center and the ESP-8266 module to create a Wi-Fi network and allow the microcontroller to communicate with the user through other devices. The system has monitoring via WEB page for remote control and also a central made of printed circuit board, composed of an LCD *display*, LED's and *push buttons*, capable of carrying out the control manually. For the development of the project, the C programming language and the MPLAB X IDE programming software were used. Simulations of the project were also carried out in the Proteus software and tests of the lighting, alarm and communication system in protoboard. Finally, tests were carried out via the printed circuit board and via the WEB page and it was found that the developed project allows the remote and face control of the lighting and alarm system to be satisfactorily, in addition to being a low cost system and easy application for a residence.

Keywords: Home automation. Domotic. Microcontroller. Programming. Web page.

LISTA DE ILUSTRAÇÕES

Figura 1 – Rede doméstica	22
Figura 2 – Microcontrolador PIC16F877A	23
Figura 3 – Pinagem do microcontrolador PIC16F877A	24
Figura 4 – Módulo ESP-01	26
Figura 5 – Módulo ESP-01 com microcontrolador ESP-8266 (Pinagem).....	26
Figura 6 – Resolução de problemas com o uso de linguagem de programação	28
Figura 7 – Estrutura básica de linguagem compilada	29
Figura 8 – Sintaxe básica de um documento HTML5	31
Figura 9 – Exemplo de componentes presentes em uma rede.....	32
Figura 10 – Interconexão de redes.....	33
Figura 11 – Rede local (LAN)	34
Figura 12 – Rede de longa distância (WAN)	35
Figura 13 – Hierarquia no sistema DNS	37
Figura 14 – Exemplo de funcionamento de um servidor DDNS.....	38
Figura 15 – Exemplo de direcionamento NAT	40
Figura 16 – Distribuição dos endereços IP's através de um servidor DHCP	42
Figura 17 – Fluxograma das etapas da pesquisa	44
Figura 18 – Fluxograma do acionamento por botões	45
Figura 19 – Estrutura de programação utilizada	46
Figura 20 – Simulação da primeira etapa no Proteus 8	47
Figura 21 – Menu principal.....	47
Figura 22 – Mensagens da função alarme	48
Figura 23 – Mensagens da função lâmpadas.....	49
Figura 24 – Simulação da comunicação entre o PIC16F877A e o ESP-8266 através dos comandos AT.....	50
Figura 25 – Testes da comunicação entre o PIC16F877A e o ESP-8266 através dos comandos AT	51
Figura 26 – Simulação da fonte de alimentação com saída fixa de 5V	52
Figura 27 – Simulação da fonte de alimentação com saída fixa de 3,3V.....	53
Figura 28 – Diagrama completo do projeto	54
Figura 29 – <i>Layout</i> da placa de circuito impresso	55
Figura 30 – Visualização 3D da placa de circuito impresso.....	56
Figura 31 – Código base para programação das páginas WEB	57
Figura 32 – Criando um domínio no site NO-IP	58
Figura 33 – Roteador TP-Link TL-WR829N.....	59
Figura 34 – Configuração inicial do roteador	59
Figura 35 – Configuração LAN	60
Figura 36 – Configuração do Servidor DHCP	61
Figura 37 – Configuração do DNS	61
Figura 38 – Configuração de redirecionamento de porta.....	62
Figura 39 – Função para informar os parâmetros de rede para o ESP	63
Figura 40 – Diagrama da rede	64
Figura 41 – Teste da fonte de tensão de 5V em protoboard	65
Figura 42 – Teste da fonte de tensão de 3,3V em protoboard	66
Figura 43 – Montagem da primeira etapa em protoboard.....	67
Figura 44 – Menu principal em testes no protoboard	67
Figura 45 – Mensagens de alarme em testes no protoboard.....	68

Figura 46 – Teste do sistema de alarme no protoboard.....	69
Figura 47 – Teste do comando das lâmpadas no protoboard.....	69
Figura 48 – Montagem do projeto completo em protoboard.....	70
Figura 49 – Teste de conexão entre PIC e ESP em protoboard.....	71
Figura 50 – Configuração da rede em protoboard.....	72
Figura 51 – Teste de rede através de um computador.....	72
Figura 52 – Status de conexão entre o ESP e o computador.....	73
Figura 53 – Placa de circuito impresso após usinagem das trilhas.....	73
Figura 54 – Placa de circuito impresso com componentes soldados.....	74
Figura 55 – Endereço IP para conexão via WEB.....	75
Figura 56 – Página WEB – Menu Principal.....	75
Figura 57 – Aviso ao usuário que o modo WEB está sendo executado.....	76
Figura 58 – Página WEB – Menu Alarme.....	77
Figura 59 – Página WEB – Alarme disparou.....	77
Figura 60 – Mensagem de “alarme Disparou”, sinalização sonora e visual.....	78
Figura 61 – Menu Lâmpadas.....	79
Figura 62 – Configuração da rede na placa.....	80
Figura 63 – Configuração de endereço IP na placa.....	80
Figura 64 – Menu de <i>Login</i>	81
Figura 65 – Menu principal.....	81
Figura 66 – Botão de alteração do modo WEB para modo manual.....	82
Figura 67 – Menu principal – Opção de retorno para modo WEB.....	83

LISTA DE TABELAS

Tabela 1 – Principais comandos AT utilizados no ESP8266	27
Tabela 2 – Características de redes por extensão geográfica	33
Tabela 3 – Entradas/Saídas do PIC16F877A com os devidos componentes inseridos	53
Tabela 4 – Lista de materiais e orçamento	84

LISTA DE ABREVIATURAS E SIGLAS

DDNS – *Dynamic Domain Name System*
DHCP – *Dynamic Host Configuration Protocol*
DNS – *Domain Name System*
DUC – *Dynamic DNS Update Client*
GM – *General Motors*
HTML – *HyperText Markup Language*
HTTP – *HyperText Transfer Protocol*
IDE – *Integrated Development Environment*
IFSC – *Instituto Federal de Santa Catarina*
IP – *Internet Protocol*
ISP – *Internet Service Providers*
LAN – *Local Area Network*
LCD – *Liquid Crystal Display*
LED – *Light Emitting Diode*
MAN – *Metropolitan Area Network*
NAT – *Network Address Translation*
PAN – *Personal Area Network*
PCI – *Placa de circuito impresso*
PIC – *Peripheral Interface Controller*
RAM – *Random Access Memory*
SEBRAE – *Serviço Brasileiro de Apoio às Micro e Pequenas Empresas*
TCC – *Trabalho de conclusão de curso*
TDL – *Top Level Domains*
XML – *Extensible Markup Language*
WAN – *Wide Area Network*
WEB – *World Wide Web*

LISTA DE SÍMBOLOS

A – Ampère

B – Byte

H – Hertz

Ω – Ohm

V – Volt

W – Watt

SUMÁRIO

1 INTRODUÇÃO	17
1.1. Justificativa	18
1.2 Objetivos	19
1.2.1 Objetivo geral.....	19
1.2.2 Objetivo específico.....	19
2 REVISÃO DA LITERATURA	21
2.1 Automação Residencial	21
2.2 Microcontroladores	23
2.2.1 PIC16F877A	23
2.2.2 ESP-8266.....	25
2.3 Linguagens de programação	28
2.3.1 Linguagem C.....	29
2.4 Linguagens de marcação – HTML5	30
2.5 Redes de computadores	31
2.5.1 DNS	35
2.5.2 DDNS.....	37
2.5.3 NAT	39
2.5.4 DHCP	41
3 METODOLOGIA	43
3.1 Caracterização da pesquisa	43
3.2 Etapas da pesquisa	43
3.2.1 Lógica de programação para acionamento por botões.....	44
3.2.2 Comunicação entre o PIC16F877A e o ESP-8266	49
3.2.3 Fontes de alimentação dos circuitos	51
3.2.4 Diagrama do projeto.....	53
3.2.5 Desenvolvimento da placa	54
3.2.6 Desenvolvimento das páginas de comando via WEB.....	56
3.2.7 Configuração do acesso externo	57
3.2.8 Lógica de programação para alteração dos modos	64
4 ANÁLISE E DISCUSSÃO DOS RESULTADOS	65
4.1 Fontes de alimentação do projeto	65
4.2 Etapa 01 – Acionamentos por botões	66

4.3 Teste de comunicação entre PIC16F877A e o ESP-8266	70
4.4 Configuração da rede	71
4.5 Teste de conexão entre o PIC e outros dispositivos	72
4.6 Placa de circuito impresso	73
4.7 Etapa 02 – Acionamentos via página WEB – Rede Local.....	74
4.8 Etapa 02 – Acionamentos via página WEB – Acesso Externo	79
4.9 Modo WEB para modo Manual	82
4.10 Modo Manual para modo WEB.....	83
4.11 Lista de materiais e orçamento	84
5 CONCLUSÃO	85
6 SUGESTÕES PARA TRABALHOS FUTUROS	87
REFERÊNCIAS	89
APÊNDICE A – Código de programação (Apenas estrutura principal)	93
APÊNDICE B – Código de programação (Biblioteca ESP-8266)	101
ANEXO A – Esquemático do ESP-01	113

1 INTRODUÇÃO

Desde o início da história, o ser humano busca maneiras de inventar e criar ferramentas que possam facilitar e suprir suas necessidades. No século XVIII a invenção da máquina a vapor proporcionou a substituição da mão de obra humana pela de máquinas, o que proporcionou o início do conceito de automação. Em 1894, Nikola Tesla obteve por meio de seus experimentos, o primeiro controle de um barco por rádio sem fio, que 56 anos depois resultou em um controle remoto, objeto indispensável quando se pensa em um ambiente residencial. Com o passar dos anos e dos avanços, novas tecnologias passaram a integrar as residências, trazendo o conceito da domótica, ou seja, a automação do ambiente (STEVAN JR.; FARINELLI, 2019).

Pode-se observar que a busca por um sistema de automação residencial tem crescido cada vez mais no cenário mundial, visto que o uso dessa tecnologia proporciona aos usuários um maior conforto, segurança, versatilidade e economia. Esses benefícios podem ser notados no aumento da praticidade para acionar equipamentos elétricos como lâmpadas e eletrodomésticos, elevar a segurança de uma residência, através de um sistema de monitoramento com alarmes e câmeras além de também contribuir para a eficiência energética, visto que é possível desativar equipamentos que não estão sendo utilizados a distância (PRUDENTE, 2017).

Segundo pesquisas realizadas pela *Business Insider*, no âmbito global, o mercado da automação residencial e da internet das coisas deverá movimentar cerca de 3 trilhões de dólares até 2026, com uma quantia de 64 bilhões de dispositivos conectados, contra os 10 bilhões de dispositivos no ano de 2018, representando um crescimento de 640% (BUSINESS INSIDER, 2019).

Ao observarmos o cenário nacional, de acordo com o Serviço Brasileiro de Apoio às Micro e Pequenas Empresas (SEBRAE), existe um potencial de 1,8 milhão de residências para fornecimento de equipamentos, e destas, apenas 300 mil já possuem automação residencial. Além disso, cerca de 78% dos consumidores brasileiros demonstram interesse em sistemas de automação residencial, o que é um número elevado de potenciais consumidores do projeto desenvolvido desse trabalho (SEBRAE, 2015).

Diante dessa lacuna de mercado apresentada no mercado brasileiro, este trabalho traz o projeto de controle e automação voltado a uma residência, contendo

funções para melhorar a segurança e o conforto do usuário. Para isso será utilizando o microcontrolador PIC16F877A, um adaptador para módulo WiFi ESP-8266, uma página WEB e um servidor DNS dinâmico gratuito, com o objetivo de efetuar o controle.

1.1. Justificativa

Apesar da evolução dos componentes eletrônicos, da boa perspectiva das pesquisas em relação ao crescimento global e nacional e do interesse dos brasileiros, o uso da automação residencial ainda é pouco explorado e pouco presente nas residências brasileiras, principalmente em cidades de pequeno porte. O principal fator se dá devido ao custo elevado para implementação deste sistema, o que faz com que os consumidores apesar de interessados, não busquem pela implementação dessa tecnologia (MARQUES, 2018).

Do ponto de vista social, a automação residencial também propicia muitos benefícios aos usuários portadores de deficiência física. Ela oferece alternativas para minimizar as barreiras enfrentadas por esses usuários ao executar tarefas simples, como por exemplo ligar uma lâmpada ou abrir um portão (PAULUS et al., 2017).

Em vista disso, busca-se atender uma lacuna de mercado em que há uma necessidade de explorar novos métodos de automação residencial, utilizando componentes de baixo custo e com uma acessibilidade maior aos usuários, visto que os apresentados nos cenários atuais requerem um alto investimento por parte do consumidor final.

Além disso, busca-se oferecer nesse projeto, um sistema integrado que permita o controle do sistema de segurança, de aspectos de conforto e economia em um único meio, a página WEB, fazendo com que o usuário tenha o máximo de funções integradas.

O trabalho também pretende contribuir para a pesquisa de sistemas de automação utilizando microcontroladores, visto que os sistemas de baixo custo encontrados atualmente na literatura utilizam a plataforma arduino com bibliotecas prontas.

1.2 Objetivos

1.2.1 Objetivo geral

Em vista dos problemas observados e da lacuna de mercado encontrada, esse trabalho apresenta por objetivo geral desenvolver um projeto de controle e automação residencial, englobando aspectos relacionados à segurança como monitoramento de alarmes e conforto, proporcionando aos usuários efetuar um controle de iluminação de sua residência a longa distância. Para atingir tal objetivo, será utilizado o microcontrolador PIC16F877A e será desenvolvido uma página WEB para controle através de smartphone e computadores.

1.2.2 Objetivo específico

Para atingir o objetivo principal, foram determinados os seguintes objetivos específicos, a fim de estruturar as principais etapas a serem cumpridas ao longo do desenvolvimento do projeto.

a) Realizar uma revisão bibliográfica em automação residencial, linguagem de programação, microcontroladores e protocolos de comunicação, a fim de auxiliar no desenvolvimento do projeto.

b) Desenvolver a lógica de programação para o microcontrolador em linguagem C através do ambiente de desenvolvimento integrado MPLAB X IDE e através do simulador Proteus.

c) Testar o sistema de monitoramento de segurança e acionamento dos pontos de iluminação através de montagem em placa de ensaio (*protoboard*) com acionamento por botões do tipo *push button*.

d) Desenvolver páginas WEB através da linguagem HTML para efetuar os comandos dos acionamentos.

e) Criar um protocolo de comunicação entre o aplicativo desenvolvido e o sistema de monitoramento desenvolvido no microcontrolador através do módulo ESP8266.

f) Utilizar um serviço de DNS dinâmico e realizar as configurações de rede necessárias para proporcionar o acesso externo.

g) Testar o sistema de controle e automação proposto através da página WEB com o objetivo de verificar e validar as ideias propostas.

2 REVISÃO DA LITERATURA

Neste capítulo são apresentados os embasamentos teóricos e científicos necessários para a elaboração do projeto de controle e automação residencial utilizando o PIC16F877A através do monitoramento de uma página WEB. São introduzidos conceitos relacionados a automação residencial, a linguagem utilizada, rede de computadores e principais protocolos de comunicação. Estes tópicos são de suma importância para comprovação e justificativa dos resultados obtidos após o desenvolvimento do estudo do caso.

2.1 Automação Residencial

O conceito de automação foi proposto inicialmente no ano de 1940 por um engenheiro da Ford, em que dizia que automação podia ser definida como a substituição de um esforço humano por ações ou controles de forma automática. Na década de 70 e 80, a *General Motors* (GM) se tornou a pioneira a utilizar controladores lógicos programáveis em sua indústria, alterando assim, o conceito base da automação. Atualmente, pode-se defini-la como sendo o uso de comandos lógicos programáveis ou automatizados que substituem a ação humana que envolvem tomadas de decisões. Dentre inúmeras vantagens, a automação proporciona benefícios como a substituição do trabalho pesado por máquinas programadas, substituição de tarefas repetitivas ou de alto risco, além de permitir elevar a produtividade e reduzir os custos de uma empresa (LAMB, 2013).

Com a evolução da tecnologia e o barateamento dos componentes eletrônicos, a automação deixou de ser apenas um meio industrial e passou a estar presentes em ambientes residenciais e prediais, com o objetivo de auxiliar em tarefas do cotidiano. Com isso, surgiu-se um novo conceito, denominado de domótica. A automação residencial ou domótica (origem romana: casa automatizada), pode ser definida como a tecnologia que permite efetuar uma dada tarefa de forma automática em um ambiente residencial ou doméstico (STEVAN JR.; FARINELLI, 2019).

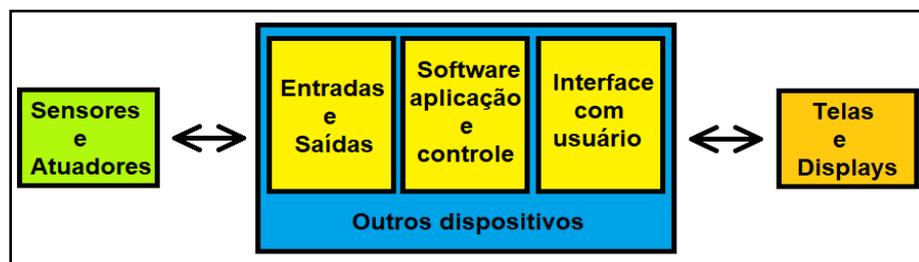
Dentre as ações presentes na domótica, pode-se citar o controle de iluminação, controle de temperatura, controle e comando do sistema de alarme, videocâmera e auxílios a portadores de deficiência física. Essas por sua vez, proporcionam ao usuário vantagens relacionadas ao conforto, como manter a

temperatura ou som ambiente agradável, maior segurança através de uma central de monitoramento, maior economia efetuando o controle de aparelhos e versatilidade. Além disso, a automação ainda permite a interação entre diversos sistemas, controlando-os através de uma única plataforma (PRUDENTE, 2017).

Para que a domótica seja implementada em um ambiente residencial, é necessário utilizar uma rede domótica. Essa rede é composta por dispositivos inteligentes que se comunicam entre si. Fazem parte dessa rede os sensores, os atuadores, os controladores e os dispositivos específicos. Os sensores são responsáveis por coletarem os dados e podem ser utilizados em aplicações como o sistema de alarme ou no controle de temperatura. Os atuadores, são os dispositivos de saída e são utilizados para realizar uma dada tarefa, como acionar o sistema de alarme ou ligar/desligar a iluminação. Os controladores, são os principais componentes, pois recebem as informações dos sensores, processam essa informação e enviam para os atuadores. Exemplos de controladores encontrados nessas aplicações residenciais são os microcontroladores (COELHO; CRUZ, 2017).

A Figura 1 apresenta uma rede domótica de forma simplificada. Os sensores recebem as informações, essas são processadas pelos *softwares*, que por sua vez informam aos usuários através de telas e *displays* e realizam uma ação através dos atuadores.

Figura 1 – Rede domótica



Fonte: Adaptado de Coelho e Cruz (2017).

2.2 Microcontroladores

Os microcontroladores são dispositivos eletrônicos compostos por entradas, saídas, memória e um processador. Apresentam como principais características possuir um tamanho reduzido, uma fácil manutenção e gerenciar tarefas internas de aparelhos eletrônicos. Um dos microcontroladores maior populares são os da série PIC (*Peripheral Interface Controller*), desenvolvidos pela *Microchip Technology Inc.*, devido a facilidade de sua linguagem de programação e a sua ferramenta de auxílio para programar, a *MPLAB Integrated Development Environment (IDE)*. As principais diferenças entre seus diversos modelos são a quantidade de entradas e saídas, a memória interna, a velocidade de processamento e o conjunto de instruções disponíveis (MARTINS, 2005).

2.2.1 PIC16F877A

O microcontrolador PIC 16F877A é um microcontrolador de 40 pinos capaz de suportar projetos complexos e efetuar tarefas simultâneas. Suas principais características são possuir 33 portas que podem ser utilizadas como entrada ou saída, uma memória interna de 256 *bytes* e uma memória de acesso aleatório ou *Random Access Memory (RAM)* de 368 *bytes* além de possuir comunicação serial, conversores e comparadores analógicos e *timers*. Quanto a sua parte elétrica, esse PIC possui uma tensão de trabalho de 4V a 5,5V e uma corrente máxima de entrada e saída das portas de 200 mA, com uma dissipação máxima de energia de 1W (SOUZA; LAVINIA, 2007).

A Figura 2 apresenta o microcontrolador PIC16F877A produzido pela empresa Microchip.

Figura 2 – Microcontrolador PIC16F877A

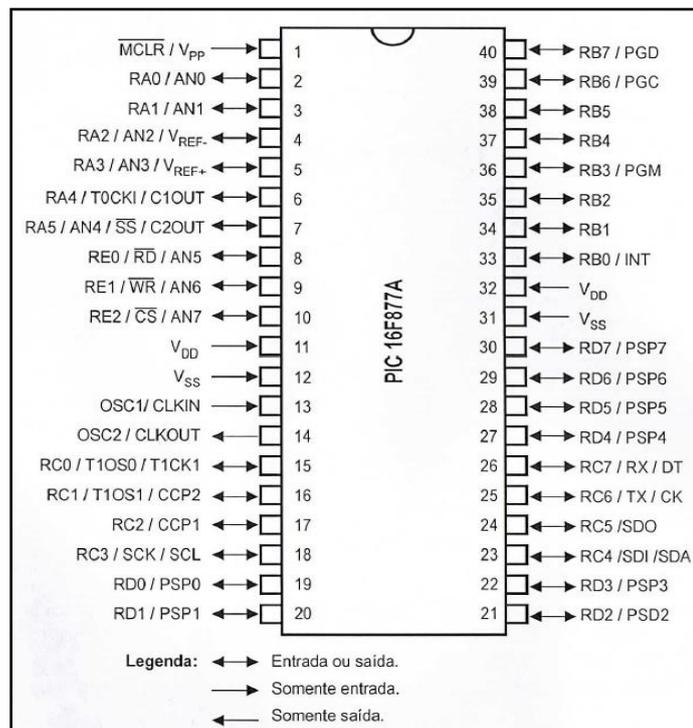


Fonte: Microchip (2020)

Dentre as diversas finalidades e aplicações do PIC16F877A, pode-se citar a adição de periféricos como um teclado matriz elaborado com botões do tipo *push button*, o uso do *display* multiplexado e o uso do *Liquid Crystal Display* (LCD) ou módulo LCD, que permite ao usuário interagir com o PIC e compreender melhor o que está sendo executado. Em uma aplicação de um sistema de alarme, por exemplo, o *display* pode ser útil para compreender se o mesmo está ligado ou não e se há alguma zona em que o alarme disparou. O PIC também permite criar um protocolo de comunicação, permitindo a comunicação entre diversos componentes na mesma placa. Pode-se utilizar vários microcontroladores cada qual designado para uma determinada aplicação. Esses podem ser denominados como *Slave* (escravos) e podem ser controlados por um *Master* (mestre), responsável pela geração do sinal de *clock* (ZANCO, 2006).

A Figura 3 apresenta a pinagem do PIC16F877A, apresentando ao lado uma breve descrição da função de cada pino. As setas direcionadas para o PIC representam as entradas, enquanto que as setas bidirecionais representam as portas que podem ser utilizadas como entrada ou como saída.

Figura 3 – Pinagem do microcontrolador PIC16F877A



Fonte: Souza e Lavinia (2007)

2.2.2 ESP-8266

O módulo ESP-8266 é um microcontrolador da fabricante chinesa *Espressif Systems*, que apresenta por característica oferecer uma solução integrada para projetos que envolvam a criação de redes Wi-Fi. Isso ocorre devido a sua eficiência em relação ao consumo de energia, apresentar um design compacto e um desempenho confiável. Dentre as aplicações do ESP destacam-se seu uso em domótica, eletrodomésticos, lâmpadas inteligentes, controle industrial sem fio, câmeras e redes de sensores (ESPRESSIF, 2020).

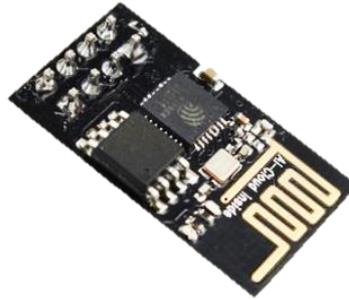
A família ESP-8266 foi projetada para suportar os padrões 802.11 b/g/n com uma velocidade de até 72,2 Mbps para o padrão 802,11 n. Além disso, o ESP conta com os modos BSS *Station mode* (Modo Estação), permitindo que o ESP se conecte a uma rede Wi-Fi criada por um roteador, *SoftAP mode* (Ponto de Acesso), o qual fornece acesso a uma rede Wi-Fi criada pelo próprio ESP permitindo até 5 dispositivos externos de se conectar, ou ambos os modos simultaneamente, operando como estação e ponto de acesso (ESPRESSIF, 2020).

O microcontrolador ESP8266 está presente em diversas placas comerciais, da qual se destacam o ESP-01, *WeeMos D1 R2* (ESP-12E), *NodeMCU* (ESP-12E) e *Witty Cloud* (ESP-12F). O ESP-01 é o modelo mais simples e possui um custo menor quando comparado ao Arduino UNO. Sua alimentação é de 3,3V, devido ao baixo consumo de energia, com 4 GPIOs digitais, 1MB de memória de programa, um oscilador de 80MHz, além de possuir uma antena embutida na placa (BATRINU, 2017).

O módulo ESP-01 foi desenvolvido pela empresa chinesa *Ai-thinker Team* e apresenta dimensões de 14,3mm largura por 24,8mm de comprimento e uma espessura de 3mm, o que o torna muito compacto para projetos que envolvam automação residencial. Além disso, possui uma corrente média de operação de 80mA e um alcance aproximado de 91 metros devido a sua antena embutida no módulo. Seu esquemático pode ser encontrado no anexo A desse trabalho (AI-THINKER, 2015).

A Figura 4 apresenta uma imagem do módulo ESP-01 da fabricante chinesa, em que é possível observar a localização dos componentes e a antena na parte superior.

Figura 4 – Módulo ESP-01

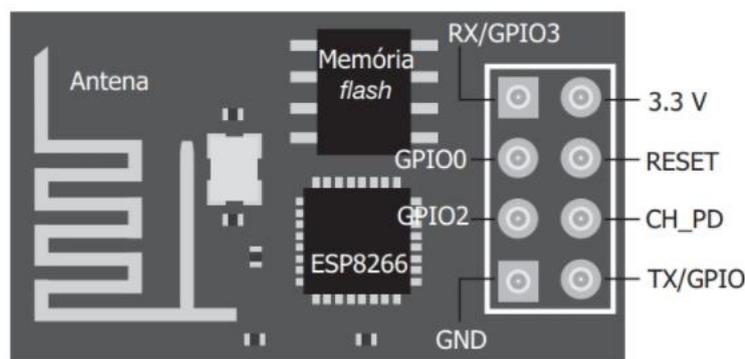


Fonte: AI-Thinker (2015)

O ESP-01 apresenta 8 pinos em seu módulo sendo duas fileiras de 4 pinos cada. O espaçamento entre os pinos é de 2,54mm em ambas as direções. Os pinos encontrados no ESP são os pinos de alimentação (3,3V e GND), os pinos de transmissão e recepção (Tx e Rx respectivamente), o pino de *reset* para limpar dados gravados no módulo (RESET), o pino de habilitação do *chip* (CH_PD), e as entradas e saídas controladas pelo *firmware* (GPIO'S) (STEVAN JR.; FARINELLI, 2019).

A Figura 5 apresenta um esquemático do módulo ESP-01, em que é possível observar a antena, o microcontrolador ESP, a memória *flash* e sua pinagem.

Figura 5 – Módulo ESP-01 com microcontrolador ESP-8266 (Pinagem)



Fonte: Stevan Jr. e Farinelli (2019)

Ao utilizar o ESP como ponto de acesso, ou seja, em modo *SoftAP*, o mesmo começa a operar com os comandos AT, os quais são fornecidos em *libat.a* pela fabricante do microcontrolador. Esses comandos são utilizados para iniciar e reiniciar o ESP, criar uma rede, definir os parâmetros dessa rede, criar páginas na WEB, enviar e receber informações, entre outras funcionalidades. Ao enviar esses comandos ao ESP se faz necessário o uso do formato AT acrescido do comando desejado

(AT+COMANDO), sendo que todas as letras desse comando devem ser maiúsculas. Além disso, todo o texto enviado deve ser entre aspas, e ao final de cada comando é necessário enviar ao microcontrolador os seguintes caracteres “\r\n”, com o objetivo de informar o término do comando (ESPRESSIF, 2018).

A Tabela 1 apresenta os principais comandos AT utilizados no ESP8266 com a respectiva resposta recebida e suas informações.

Tabela 1 – Principais comandos AT utilizados no ESP8266

Comando	Descrição do comando	Resposta
AT	Testa a inicialização do AT. Deve ser o primeiro comando a ser enviado ao ESP.	OK
AT+RST	Utilizado para reiniciar o módulo.	OK
AT+CWMODE	Define o modo de operação do ESP. Para utilizar essa função é necessário utilizar AT+CWMODE:<mode>, sendo: <mode>=1 – modo estação <mode>=2 – modo ponto de acesso (AP) <mode>=3 – ambos os modos	OK
AT+CWSAP	Comando responsável por configurar o modo de ponto de acesso. Para utilizar essa função é necessário utilizar AT+CWSAP:<ssid>,<pwd>,<chl>,<ecn> sendo: <ssid>: nome da rede <pwd>: senha da rede contendo de 8 a 64 caracteres ASCII <chl> : ID do canal <ecn> : método de criptografia da senha (usar o valor 3 para o método WPA2_PSK)	OK
AT+CIPMUX	Habilita/desabilita o modo de múltiplas conexões, o qual permite até 5 dispositivos conectados na rede criada. Para utilizar essa função é necessário utilizar AT+CIPMUX:<mode>, sendo: <mode>=0 – conexão simples (1 dispositivo) <mode>=1 – múltiplas conexões (Até 5 dispositivos)	OK
AT+CIPSERVER	Comando utilizado para criar ou deletar um servidor TCP. Para utilizar essa função é necessário utilizar AT+CIPSERVER=<mode>,<port> sendo: <mode>: usar 0 para deletar o servidor ou 1 para criar o servidor <port> :Define o número da porta. Utilizar 80 para protocolo HTTP e envio de web pages.	OK
AT+CIFSR	Fornece o endereço IP local.	OK
AT+CIPSTART	Configura o tipo de conexão em TCP, UDP ou conexão SSL. Para utilizar essa função é necessário utilizar AT+CIPSTART=<type>: <type>: <i>string</i> que indica tipo de conexão (usar TCP, UDP ou SSL)	OK
AT+CIPSEND	Escreve os dados no buffer e faz o envio para conexão TCP. Utilizado para enviar o código responsável pela criação da página web.	OK
AT+CIPCLOSE	Encerra a conexão TCP/UDP/SSL.	OK

Fonte: Adaptado de Espressif, 2018.

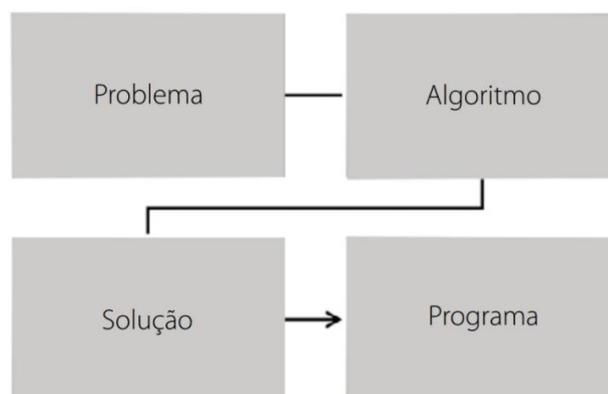
2.3 Linguagens de programação

Uma linguagem de programação pode ser definida como um conjunto de recursos e regras que podem ser utilizados para construir um programa que possa solucionar um problema específico ou efetuar uma dada função programada. Ela é responsável por determinar quais são os recursos disponíveis e como eles devem ser utilizados para executar as tarefas propostas (MELO; SILVA, 2003).

Seu principal objetivo é efetuar a tradução entre a nossa linguagem e a linguagem que o computador compreende, também conhecida como a linguagem de máquina. Pode-se definir que quando uma linguagem é semelhante a linguagem da máquina ela é considerada de baixo nível e quando a linguagem se assemelha a escrita e a fala do ser humano ela é considerada de alto nível. A linguagem de programação também permite sanar problemas através da criação de um programa. Ao analisar o problema, desenvolve-se um algoritmo capaz de resolver esse problema, tendo como saída uma resposta, também denominada de solução (SANTOS; SARAIVA; GONÇALVES, 2018).

A Figura 6 apresenta a sequência utilizada para solucionar um problema através de um programa utilizando uma linguagem de programação.

Figura 6 – Resolução de problemas com o uso de linguagem de programação



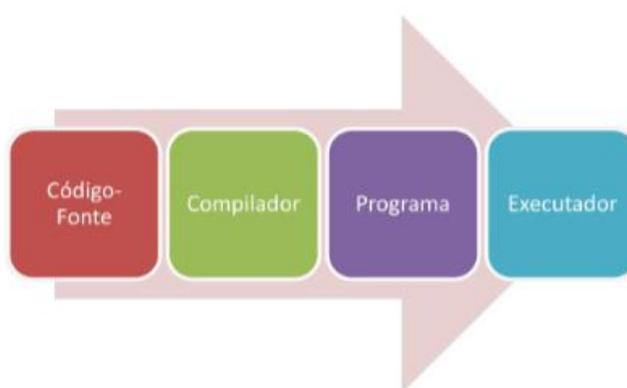
Fonte: Santos, Saraiva e Gonçalves (2018)

A estrutura de uma linguagem de programação pode ser definida em 4 etapas principais. A primeira delas é o código fonte. Ela é o arquivo contendo as informações, operações e funções que foi desenvolvido pelo programador em uma dada linguagem de programação. Em seguida, há o compilador. Ele é o responsável por fazer a

conversa entre a linguagem desenvolvida e a linguagem da máquina, além de ser responsável por verificar a sintaxe e a estrutura do programa. Após compilar o código-fonte, tem-se o programa, cujo é o resultado da compilação em linguagem de máquina. Por fim há o executador, ou seja, o sistema que irá rodar o programa desenvolvido, como por exemplo o sistema Windows (FERREIRA,2020).

A Figura 7 mostra a estrutura básica da linguagem de programação contendo as etapas de código-fonte, compilador, programa e executador.

Figura 7 – Estrutura básica de linguagem compilada



Fonte: Ferreira (2020)

2.3.1 Linguagem C

Criada em 1972 por Dennis Ritchie nos laboratórios Bell, a linguagem C é uma linguagem de programação que foi desenvolvida com o objetivo de permitir a escrita do sistema operacional Unix. Assim como o FORTRAN, PASCAL e COBOL, o C é uma linguagem de alto nível, o que permite aprendizado facilitado e um menor custo na elaboração e manutenção de *softwares*. Além disso, o C permite a maior parte de funcionalidades da linguagem *Assembly*, a qual é uma linguagem de baixo nível. Diferente das outras linguagens que possuem um objetivo específico, o C possui uma ótima adaptação, modelando-se a qualquer projeto e aplicação, o que faz com que seu uso seja amplamente difundido (DAMAS, 2016).

Essa linguagem é amplamente utilizada e seu campo abrange desde programas simples até mesmo a criação de sistemas operacionais, passando por compiladores e editores de texto. Esse fato se dá devido as vantagens apresentadas por essa linguagem como sua portabilidade, visto que os programas em C geram um

único arquivo executável e precisam apenas de um compilador C para executá-las. Além disso, a linguagem possui uma gama de opções que permitem trabalhar com funções e arquivos Dos, apenas adicionando bibliotecas padrões encontradas. O C também é conhecido pela sua sintaxe fácil compreensão, acesso facilitado a memória e hardware e permite a criação de sistemas desacoplados, isolando pequenas partes do programa para a análise de eventuais erros (SANTOS; SARAIVA; GONÇALVES, 2018).

A linguagem C é amplamente utilizada em microcontroladores para o desenvolvimento de *softwares* de aplicação, devido permitir a criação de programas de baixa até alta complexidade, o que é inviável ou muito complexo em algumas linguagens, como o *Assembly*, que não é uma linguagem estruturada. Atualmente, a grande parcela de microcontroladores encontrados no mercado apresentam o compilador C, como é o caso dos microcontroladores PIC's. Além disso, devido a sua facilidade de programação, o usuário não precisa perder muito tempo com a sintaxe da linguagem e pode focar no que realmente é importante para ele, a criação de novos projetos (PEREIRA, 2003).

2.4 Linguagens de marcação – HTML5

As linguagens de marcação são um tipo de linguagem destinadas a anotações de texto, de modo que seja possível adicionar características e informações específicas sobre esse texto. Elas são amplamente utilizadas na web, devido permitir o trabalho em conjunto entre diferentes plataformas, facilitando a comunicação entre os autores, editores e gráficas. Esse tipo de linguagem pode ou não conter uma formatação semântica de apresentação. A *HyperText Markup Language* (HTML) ou Linguagem HTML é um exemplo de linguagem de marcação que apresenta essa formatação, visto que permite ao usuário que está visualizando, compreender a informação que está escrita, diferentemente de outras, como é o caso da *Extensible Markup Language* (XML) ou linguagem XML (FLATSCHART, 2011).

O HTML fornece a estrutura necessária a uma página web, porém não garante sua aparência visual. Outra característica é que ao utilizar o HTML deve-se começar pela adição dos textos na página e em seguida ir acrescentando as características que irão definir aquele texto, como por exemplo se este é um título ou um parágrafo, o seu tamanho e a sua posição na página (LEMAY, 2003).

Uma sintaxe básica de um documento HTML5 compreende a definição do documento e três elementos essenciais, sendo eles, o elemento `<html>`, o elemento `<head>` e o elemento `<body>`. A definição do documento deve ser obrigatoriamente a primeira instrução, visto que ele informa ao navegador a versão da linguagem que está sendo utilizada. O elemento `<html>` compreende toda a estrutura do programa e é responsável por indicar que a página é um documento HTML. O elemento `<head>` compreende todas as informações relacionadas ao cabeçalho da página, como o título do navegador, cujo é obrigatório constar no interior do elemento `<head>`. Por fim o elemento `<body>` define o corpo do documento e é nessa etapa que são inseridos os textos presentes na página, as imagens, hiperlinks, listas e tabelas e todo o restante do conteúdo visível ao usuário (TERUEL, 2014).

A Figura 8 apresenta uma sintaxe básica para um documento HTML5, contendo a informação de que é um a linguagem de marcação HTML5 (primeira linha do código). É possível observar o título, elemento obrigatório da página, e a meta com a codificação UTF-8, o qual permite representar qualquer caractere padrão do Unicode3 e ASC II. O atributo *lang* também fornece ao navegador o idioma que está sendo utilizado na página.

Figura 8 – Sintaxe básica de um documento HTML5

```
<!DOCTYPE HTML>
<html lang= "pt-br">>
  <head>
    <meta charset="UTF-8">
    <title>Estrutura básica de uma página HTML5</title>
  </head>
  <body>
    <p>Parágrafo aqui...</p>
  </body>
</html>
```

Fonte: Teruel (2014)

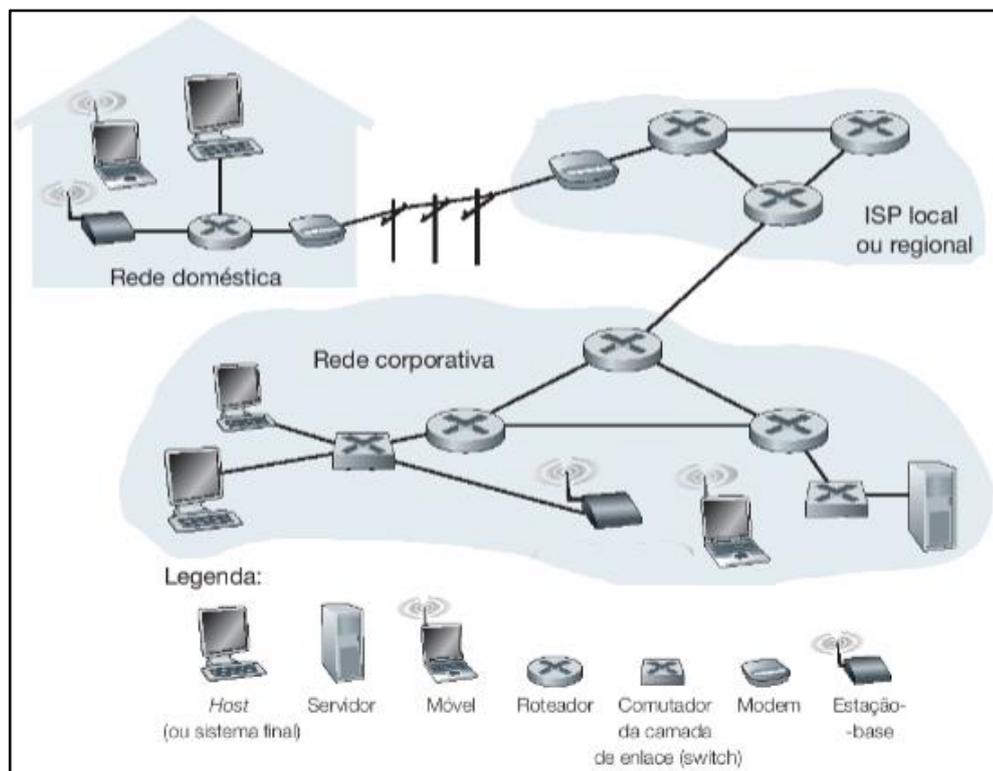
2.5 Redes de computadores

A internet é uma rede de computadores que conecta diferentes aparelhos e dispositivos ao redor de todo o mundo com o principal objetivo de transmitir e receber dados. Essa rede abrange desde uma comunicação entre um aparelho doméstico

(televisão, celular, *notebook*) com um roteador, até mesmo a comunicação entre dois servidores em diferentes continentes. Cada equipamento conectado à rede pode ser chamado de hospedeiro ou sistema final (computadores, aparelhos celulares, etc.) e a conexão entre cada um deles pode ser feita por enlaces de comunicação (cabos coaxiais, fibra ótica, ondas de rádio) ou comutadores de pacotes (*switches*, roteadores). Para que esses hospedeiros acessem a Internet, é utilizado um provedor de serviço de internet, também conhecido como *Internet Service Providers (ISP)* (KUROSE; ROSS, 2013).

A Figura 9 mostra exemplo de uma rede contendo diferentes tipos de dispositivos conectados. É possível observar hospedeiros como os computadores (*Hosts*), *notebooks* e comutadores como os *switches*, roteadores e modems. Também se verifica a presença de um provedor de internet local conectando redes menores como uma rede doméstica e uma rede corporativa.

Figura 9 – Exemplo de componentes presentes em uma rede



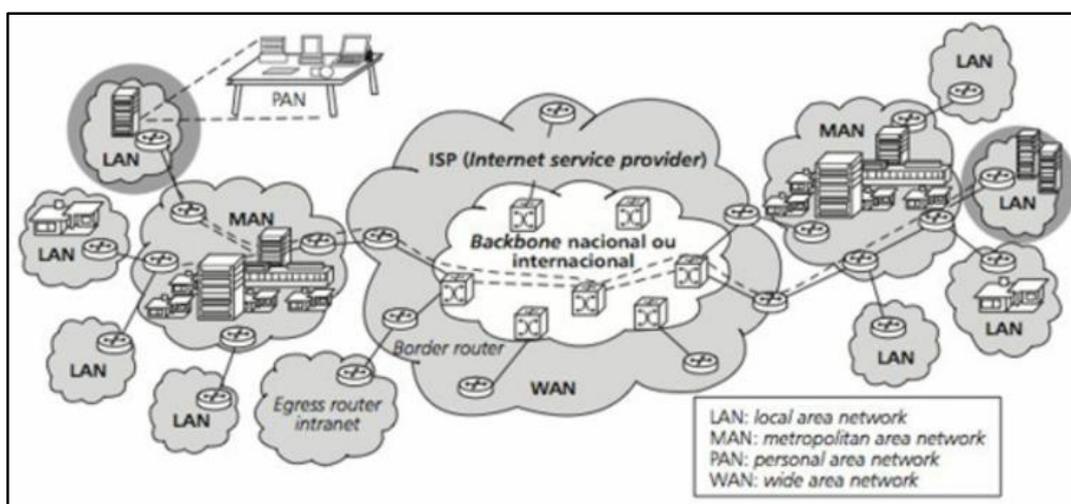
Fonte: Adaptado de Kurose e Ross (2013)

Dentre as características de uma rede, destacam-se o seu desempenho, buscando sempre diminuir o tempo de transito para o envio e recebimento de uma

mensagem, a confiabilidade entregando a mensagem por completo ao destino final e sua segurança, protegendo a mensagem para que somente o emissor e o receptor tenham acesso a ela. As redes podem ser divididas em 4 categorias, de acordo com a distância física entre os dispositivos conectados a ela. Dentre essas categorias se destacam as redes locais, denominadas de *Local Area Network* (LAN) e as redes globais, denominadas de *Wide Area Network* (WAN). Há ainda as redes metropolitanas, *Metropolitan Area Network* (MAN) e as redes pessoais ou *Personal Area Network* (PAN) (FOROUZAN, 2006).

A Figura 10 exemplifica as 4 categorias de rede descritas anteriormente. É possível observar que a rede WAN (internacional ou nacional) conecta diferentes redes MAN (cidades) que por sua vez alimentam as redes LAN (residências e empresas).

Figura 10 – Interconexão de redes



Fonte: Carissimi, Rochol e Granville, (2009)

A Tabela 2 apresenta as principais características como área de cobertura, os meios utilizados para a transmissão de dados e as taxas de comunicação para as 4 categorias de redes.

Tabela 2 – Características de redes por extensão geográfica

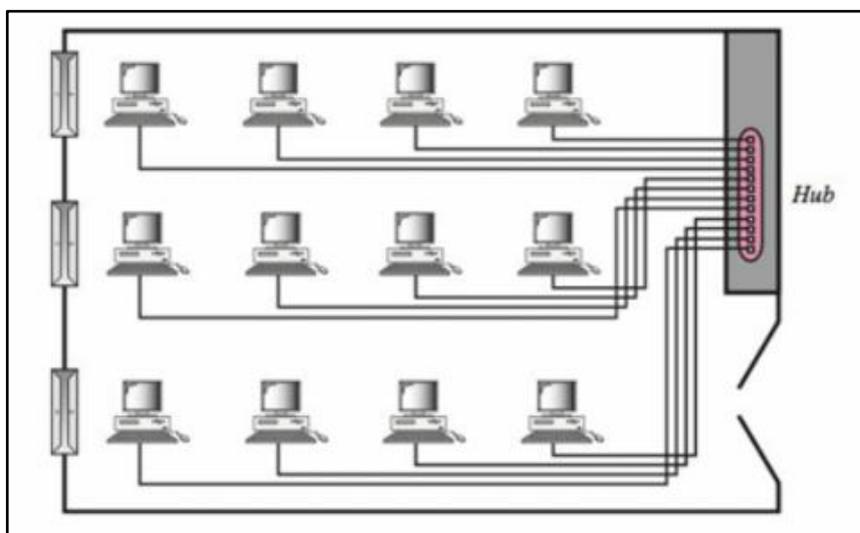
Tipo de rede	Cobertura	Meios	Taxas
PAN	Alguns metros	Wireless	2 Mbit/s
LAN	Alguns quilômetros	Par trançado e fibra óptica	10Mbit/s a 10Gbit/s
MAN	Centenas de quilômetros	Fibra óptica e rádio frequência	155Mbit/s a 10Gbit/s
WAN	Nacional e internacional	Fibra óptica	64kbit/s a Tbit/s

Fonte: Adaptado de Carissimi, Rochol e Granville, 2009.

As redes locais (LAN) são padronizadas internacionalmente pelo IEEE 802 e são projetadas para conectar dispositivos que estejam próximos fisicamente permitindo o compartilhamento de informações entre *softwares* (aplicativos) e *hardwares* (computadores e impressoras). Dentre suas características destacam-se a alta taxa de transmissão de dados, grande escalabilidade, baixo custo e as baixas taxas de erros. Além disso, elas abrangem desde ambientes residenciais até diversos ambientes empresariais (MAIA, 2013).

A Figura 11 apresenta um exemplo de rede local com 12 computadores sendo conectados a um *Hub* em um gabinete.

Figura 11 – Rede local (LAN)



Fonte: Forouzan (2006)

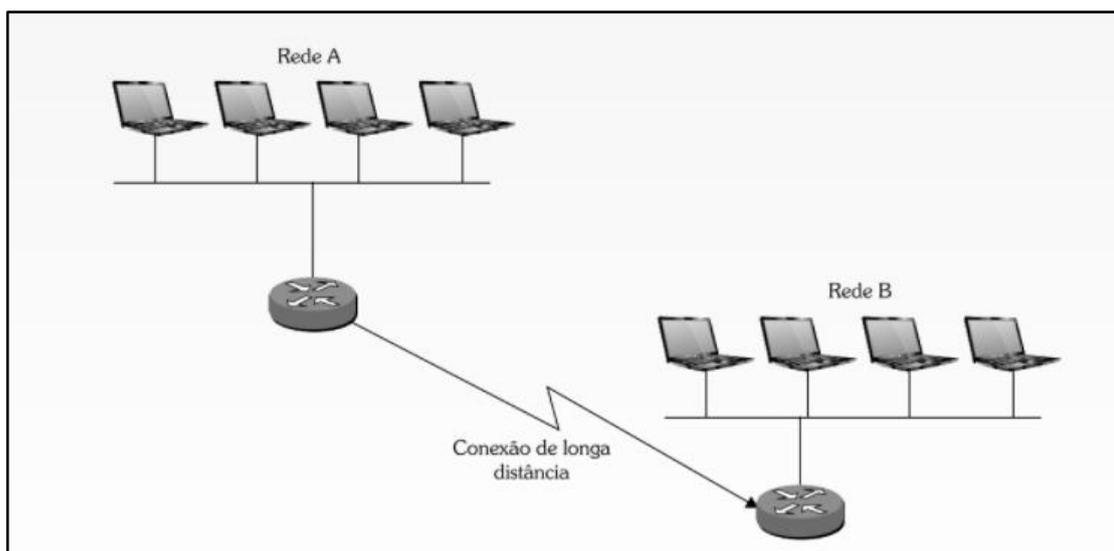
As redes WAN são aquelas que possibilitam a transmissão de dados a grandes distâncias, podendo envolver cidades, estados, países e continentes. Esse tipo de rede apresenta velocidades inferiores às das redes LAN, em virtude da distância e da quantidade de dados transmitidos. Além disso, possuem protocolos específicos de comunicação. Elas podem ser classificadas em WAN comutada, sendo responsável por conectar os sistemas finais (conexão de um roteador a outra LAN ou WAN) e WAN ponto a ponto, responsável por conectar um computador ou uma pequena LAN a um provedor de internet (FOROUZAN, 2006).

A primeira WAN comutada desenvolvida foi a X.25. Ela foi implementada em 1980 e utilizava cabos de cobre, o que não permitia altas taxas de transmissão. Além disso, eram sujeitas a altas taxas de erros, o que necessitava de muitos mecanismos

para corrigir e tratar os dados. Com o surgimento dos cabos de fibra óptica, foi adotado o *frame relay*, permitindo uma maior velocidade e eficiência na transmissão de dados. Esse novo método também não necessita de todo tratamento de dados da X.25 e é amplamente utilizado atualmente (MORAES, 2020).

A Figura 12 mostra um exemplo de rede WAN comutada, interligando dois roteadores que alimentam suas redes locais.

Figura 12 – Rede de longa distância (WAN)



Fonte: Moraes (2020)

2.5.1 DNS

Ao utilizar um serviço de buscas na internet para encontrar um site ou um e-mail (hospedeiros), utiliza-se um nome de hospedeiro (*Hostname*). Esses nomes, geralmente são fáceis de serem lembrados, devido estarem associados a pessoas, marcas ou empresas, porém não fornecem informações sobre sua localização, além de não apresentarem uma padronização, o que torna difícil de serem armazenados e processados por equipamentos presentes na rede, como servidores, roteadores e modems. Para solucionar esse problema, as máquinas presentes na rede possuem uma numeração única para identificação através do *Internet Protocol* (IP), denominado de endereço IP. (KUROSE; ROSS, 2013).

Um endereço IP pode ser definido como um conjunto de 4 números decimais separadas por pontos, sendo que cada um deles apresenta um valor correspondente entre 0 e 255. Cada um desses números é chamado de octeto e representa 1 *byte* ou

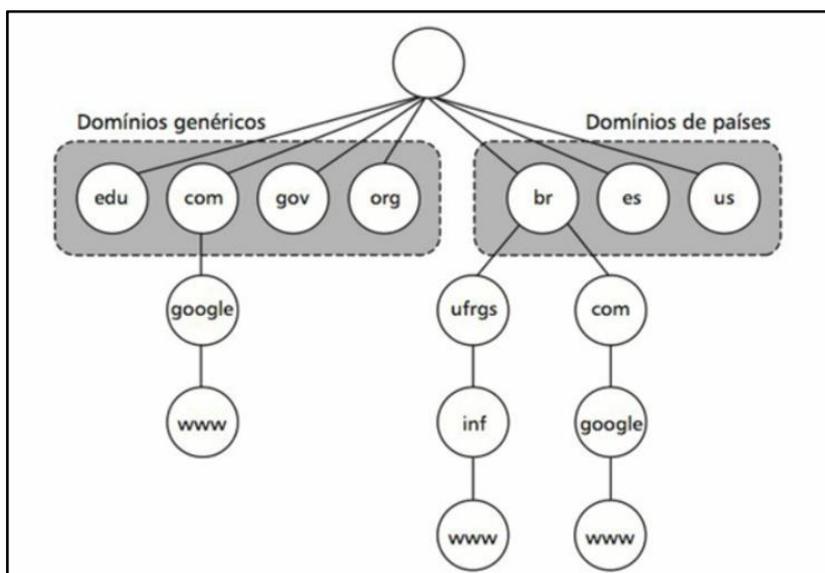
8 bits. Pode-se citar exemplos de endereços IP como sendo 200.132.6.22, 200.132.1.1 ou 143.55.1.1. Suas principais funções são identificar a localização do hospedeiro e da interface de hospedeiro. A análise do endereço é sempre feita da esquerda para direita, obtendo cada vez mais informações do hospedeiro (SCHMITT; PERES; LOUREIRO, 2013).

O Sistema de Nomes de Domínios ou *Domain Name System* (DNS), é um sistema complexo e sofisticado que busca mapear o nome do hospedeiro ao seu endereço IP na internet. Assim, ao digitar o nome de um site em um navegador, uma requisição é feita ao serviço DNS, que informa o endereço IP correspondente. Esse mapeamento por sua vez, é feito por vários servidores diferentes ao redor do mundo. Quando um cliente solicita um endereço x para um servidor y, ele envia uma mensagem de resposta caso encontre o endereço solicitado. Caso isso não ocorra, o servidor y se torna um novo cliente temporário e solicita o endereço x a outro servidor z, e o processo se repete até que algum servidor encontre o endereço desejado (COMER, 2016).

Com o objetivo de organizar e tornar o processo de busca mais rápido, foram criados níveis de domínios, através de uma hierarquia de domínios. Os domínios de mais alto nível são encontrados ao final dos endereços (à direita) e podem ser chamadas de *Top Level Domains* (TDL). Eles podem ser genéricos se referindo a instituições educacionais (.edu), organizações comerciais (.com), governos (.gov), organizações não comerciais (.org) entre outros ou domínios relacionados a países como Brasil (.br), Estados Unidos (.us) e Espanha (.es). A esquerda dos domínios, são encontrados os subdomínios, em que esses pertencem ao domínio especificado. Os subdomínios por sua vez, podem conter outros subdomínios sem uma limitação de quantidade. Por fim, há as folhas do sistema de nomes que representam o nome de máquina em que está hospedado o site (CARISSIMI; ROCHOL; GRANVILLE, 2009).

A Figura 13 apresenta exemplos de domínios de mais alto nível e hierarquias de domínios encontradas em buscas na internet. Pode-se citar como exemplo o endereço www.google.com.br, em que é possível perceber que o país em que o site se encontra é o Brasil (.br), que se trata de uma organização comercial (.com), com o nome da empresa Google (.google) e que se refere a uma máquina (www) em que o site está hospedado.

Figura 13 – Hierarquia no sistema DNS



Fonte: Carissimi, Rochol e Granville (2009)

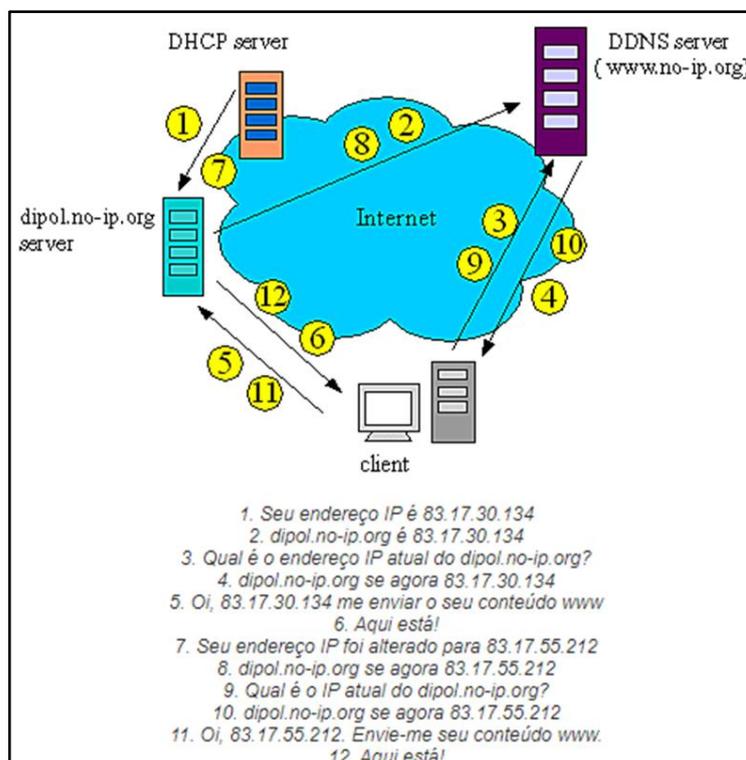
2.5.2 DDNS

Quando pretende-se realizar alterações nos endereços, como um acréscimo ou remoção de host ou até mesmo uma mudança de endereço IP, há a necessidade de alterar essas informações no arquivo mestre do DNS, porém isso envolve um processo manual e custoso, devido ao crescimento da internet e ao número de endereços existentes atualmente. Em vista disso, surgiu o *Dynamic Domain Name System* (DDNS), um serviço que atualiza o serviço DNS de forma dinâmica, ou seja, automática. Assim, ao realizar uma alteração de um endereço IP, as informações são transmitidas ao servidor primário DNS, através do *Dynamic Host Configuration Protocol* (DHCP) ou protocolo DHCP, que por sua vez atualiza as informações. Em seguida o servidor primário notifica os servidores secundários sobre a atualização. Todo o processo é realizado com mecanismos de autenticação, a fim de evitar fraudes (FOROUZAN, 2006).

Um servidor DDNS apresenta um funcionamento semelhante a um servidor DNS, mapeando os domínios cadastrados aos endereços IP's, porém apresenta como vantagem que esses endereços podem ser atualizados. Um serviço DDNS pode ser utilizado quando há a necessidade de hospedar um site, uma câmera ou um aplicativo e não há um endereço IP estático. Isso ocorre devido ao servidor DDNS garantir que o site estará disponível no domínio cadastrado independentemente do número de IP ter sido alterado ou não (DIPOL, 2013).

A Figura 14 apresenta um exemplo de como um servidor DDNS se comporta na atualização do endereço IP de um site.

Figura 14 – Exemplo de funcionamento de um servidor DDNS



Fonte: Dipol (2013)

Para auxiliar na interpretação da Figura 14, os passos a seguir descrevem as etapas demonstradas para uma requisição do domínio dipol.no-ip.org, antes e depois da alteração do endereço IP:

1 – Um servidor DHCP mapeia um endereço IP 83.17.30.134 ao responsável por criar o site dipol.no-ip.org.

2 – O servidor DDNS recebe a informação de endereço IP e domínio do site criado.

3 – O cliente solicita o endereço IP referente ao site dipol.no-ip.org ao servidor DDNS.

4 – O servidor DDNS informa o endereço IP 83.17.30.134 ao cliente.

5 – O cliente solicita a página ao endereço IP informado pelo servidor DDNS.

6 – O cliente recebe a página referente ao site dipol.no-ip.org.

7 – O endereço IP é atualizado para 83.17.55.212 pelo servidor DHCP após um determinado período de tempo, devido ao usuário possuir um endereço IP dinâmico.

Os passos 8 a 12 repetem os passos 2 a 6 para o novo endereço IP 83.17.55.212 fornecido pelo servidor DHCP.

Dentre as diversas empresas fornecedoras de um serviço de DNS dinâmico, pode-se citar a NO-IP, a DYN e a Zoneedit. A NO-IP se destaca por apresentar um plano gratuito destinado para pequenos projetos e experimentos, fornecendo até 3 domínios por um prazo de 30 dias. Esses domínios podem ser renovados após esse período por mais 30 dias. Além disso, apresenta o *Dynamic DNS Update Client* (DUC) ou em português, Cliente de Atualização Dinâmica, um aplicativo fornecido pelo site que verifica frequentemente as mudanças de endereço IP e os atualizava automaticamente. O NO-IP é compatível com diversas marcas de roteadores, dentre elas a TP-Link (NO-IP, 2021).

A DYN é outra empresa que fornece esse tipo de serviço, porém é possível efetuar um teste gratuito apenas pelo período de 7 dias. Em seu plano pago, fornece até 30 nomes de host por pacote comprado, por um período de 1 ano (DYN, 2021). A Zoneedit também possui um plano gratuito para uma zona DNS, sendo possível aumentar esse número para três. Também fornece DNS dinâmico e controle total de zona. Além disso, disponibiliza créditos mensais para o gerenciamento de DNS dinâmico (ZONEEDIT, 2021).

2.5.3 NAT

Uma rede de uma empresa ou residência possui diversos computadores, celulares e equipamentos que necessitam se comunicar entre si e com a internet, a fim de realizar trocas de dados e informações. Para isso se faz necessário a utilização de endereços IP's, porém esses apresentam uma quantia limitada de endereços. Devido a essa limitação na internet, utiliza-se três conjuntos de numerações reservadas para redes internas. Assim, diferentes redes locais podem utilizar o mesmo endereço. Quando um computador dessa rede interna necessita acessar a internet, se faz necessário associar esse endereço IP da rede interna a um endereço IP válido na internet (SOUSA, 2009).

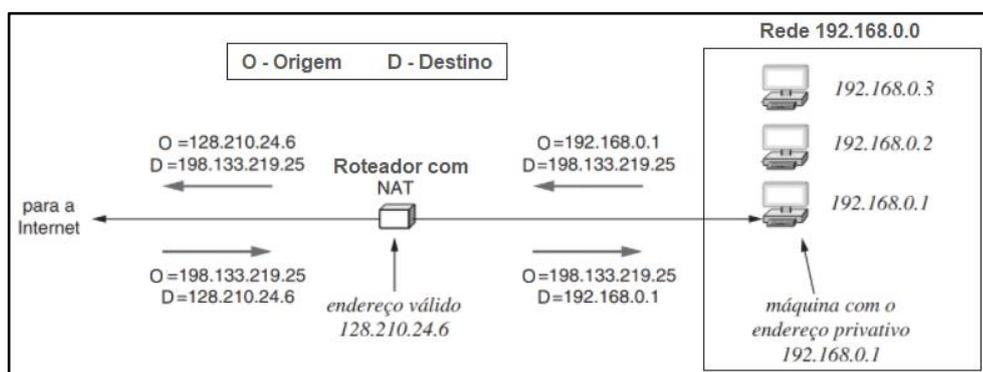
O *Network Address Translation* (NAT) é uma técnica de tradução de endereços IP introduzida pela *Cisco Systems* que teve sua utilização iniciada a partir de 1990 e que permite a conversão de endereços IP's válidos (IP's registrados que possuem acesso à internet) em endereços IP's inválidos (IP's de uma rede interna criada pelo

roteador para a conexão de equipamentos como computadores e dispositivos) e vice-versa. Para isso, o roteador ao receber um pacote de dados de um endereço IP inválido presente em sua rede interna, troca o endereço de origem do pacote para o seu endereço de IP válido e envia esse pacote para a internet. Ao receber o pacote da internet, o roteador faz o processo inverso e retorna o pacote para o endereço IP inválido de origem. (MORAES, 2020).

Esse mapeamento de endereços pode ocorrer de um ou mais endereços inválidos para um único endereço válido, devido ao *software* de NAT possuir um cache relacionando os pacotes enviados aos seus endereços IP's internos. Assim, ao receber o pacote da rede o NAT remove o endereço IP global e insere o endereço IP da estação do usuário que solicitou aquela informação. Dentre suas vantagens, pode-se destacar o aumento da segurança interna da rede, visto que os endereços das máquinas presentes na rede não podem ser vistos, pois ficam mascarados (WHITE, 2015).

A Figura 15 mostra um exemplo de direcionamento NAT em uma rede 192.168.0.0 conectada à internet através de um roteador de endereço global válido 128.210.24.6. Ao considerar um pacote de dados sendo enviado de uma máquina dessa rede para um destino na internet, tem-se que o pacote sai da máquina de origem com endereço IP 192.168.0.1 e chega ao roteador com endereço IP 128.210.24.6. O roteador traduz o IP privado 192.168.0.1 para o endereço válido 128.210.24.6 e envia a mensagem para o novo destino 198.133.219.25. O Destino de endereço IP 198.133.219.25 desconhece a existência da rede 192.168.0.0 e da máquina 192.168.0.1, visto que quem fez o direcionamento foi o roteador presente no endereço 128.210.24.6.

Figura 15 – Exemplo de direcionamento NAT



Fonte: Adaptado de Comer (2016)

2.5.4 DHCP

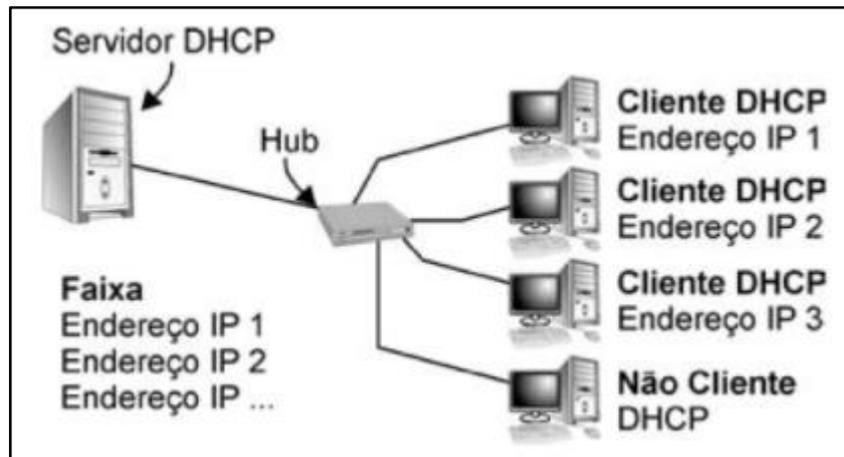
Os computadores presentes em uma rede podem ser conectados através de uma configuração estática ou uma configuração dinâmica. A primeira opção é realizada de forma manual por um profissional na área e se faz necessário ter acesso ao computador, visto que não é possível fazer um acesso remoto. A segunda opção ocorre de forma automática, através do *Dynamic Host Configuration Protocol* (DHCP), permitindo que diversos dispositivos se conectem à rede de forma rápida e eficiente. O protocolo DHCP apresenta como objetivo distribuir os endereços IP's, máscaras, *Gateways* e outras configurações de forma automática a cada dispositivo conectado na rede. (SCHMITT; PERES; LOUREIRO, 2013).

Entre as principais características do protocolo DHCP destaca-se a recuperação da perda de dados e a exclusão de duplicação de pacotes. Caso uma máquina não receba uma resposta por esse protocolo, a requisição será retransmitida e se uma resposta repetida for recebida, ela será ignorada e excluída. Outro fator importante é que esse protocolo armazena em cache o endereço IP do servidor, tornando a renovação de endereços mais eficiente. Além disso, o DHCP prevê a inundação sincronizada, ou seja, caso vários clientes se conectem simultaneamente, haverá um tempo de atraso entre eles para transmitir as requisições (COMER, 2016).

Um servidor DHCP apresenta uma faixa de endereços IP's predefinidos. Assim que um cliente se conecta a esse servidor através de uma ponte, o servidor fornece um endereço de IP válido automaticamente a esse cliente. Esse processo se repete para os demais clientes que se conectam ao servidor. Com isso, o servidor faz a alocação dinâmica dos endereços e das configurações desses computadores conectados a ele. Caso esse servidor atenda a um outro segmento de rede, como um roteador, esse deve apresentar o agente DHCP (SOUSA, 2009).

A Figura 16 demonstra a distribuição de uma faixa de endereços IP's do servidor DHCP através de um Hub. É possível verificar que esse servidor atende 3 clientes DHCP, e um segmento de rede (roteador), fornecendo um endereço IP válido. Esse roteador poderá conectar vários novos clientes a rede através do agente DHCP. Se algum cliente desse roteador quiser enviar um pacote de dados para a internet, poderá ser utilizado o NAT.

Figura 16 – Distribuição dos endereços IP's através de um servidor DHCP



Fonte: Sousa (2009)

Na grande maioria das redes domésticas, é comum utilizar serviços provenientes de um servidor DHCP. Isso ocorre devido as provedoras de internet utilizarem esse tipo de servidor, fornecendo um endereço IP temporário aos clientes pelo período em que eles estão conectados. Assim que o cliente desconecta o seu aparelho (roteador, modem, computador), o endereço IP fornecido pela provedora volta a ficar disponível para outro cliente. Isso faz com que a provedora possa reutilizar aquele endereço caso o cliente deixe de adquirir os serviços prestados por ela (WHITE, 2015).

3 METODOLOGIA

Este capítulo apresenta por objetivo demonstrar as metodologias utilizadas para a elaboração do projeto de forma detalhada a fim de promover ao leitor condições mínimas para poder repetir o trabalho desenvolvido. Ele apresenta tópicos como a caracterização da pesquisa, os experimentos realizados e as etapas de fabricação do projeto.

3.1 Caracterização da pesquisa

A pesquisa pode ser caracterizada segundo o seu fim, como uma pesquisa aplicada, visto que foram estudados conceitos teóricos e esses foram empregados na prática para a elaboração do projeto de controle e automação residencial. Um exemplo desse tipo de pesquisa observado no trabalho foi dos conhecimentos teóricos sobre o funcionamento do PIC que contribuíram para efetuar a programação do sistema de automação.

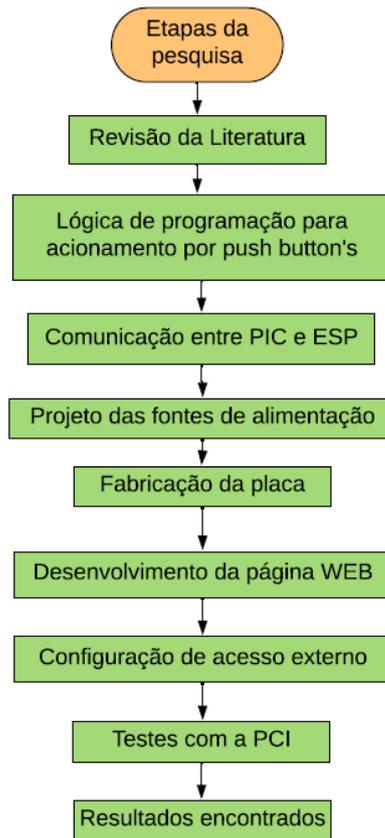
Quanto a caracterização segundo os objetivos, a pesquisa foi classificada em exploratória, em que foi construído uma revisão da literatura do tema e foram efetuadas pesquisas de trabalhos de assuntos similares.

Por fim, a pesquisa realizada nesse trabalho pode ser caracterizada do ponto de vista dos procedimentos técnicos como uma pesquisa experimental, visto que elaborou-se um projeto de controle de um sistema de automação residencial e realizou-se os testes necessários para colocar o projeto em prática.

3.2 Etapas da pesquisa

Com o objetivo de demonstrar de forma breve e simplificada as etapas de pesquisas que são apresentadas neste capítulo, foi apresentado o fluxograma contendo todas as fases necessárias para o projeto de controle e automação residencial utilizando o PIC16F877A através do monitoramento de uma página WEB. Essas etapas foram descritas de forma detalhada nos tópicos a seguir. A Figura 17 demonstra o fluxograma contendo as etapas da pesquisa.

Figura 17 – Fluxograma das etapas da pesquisa



Fonte: O autor (2021)

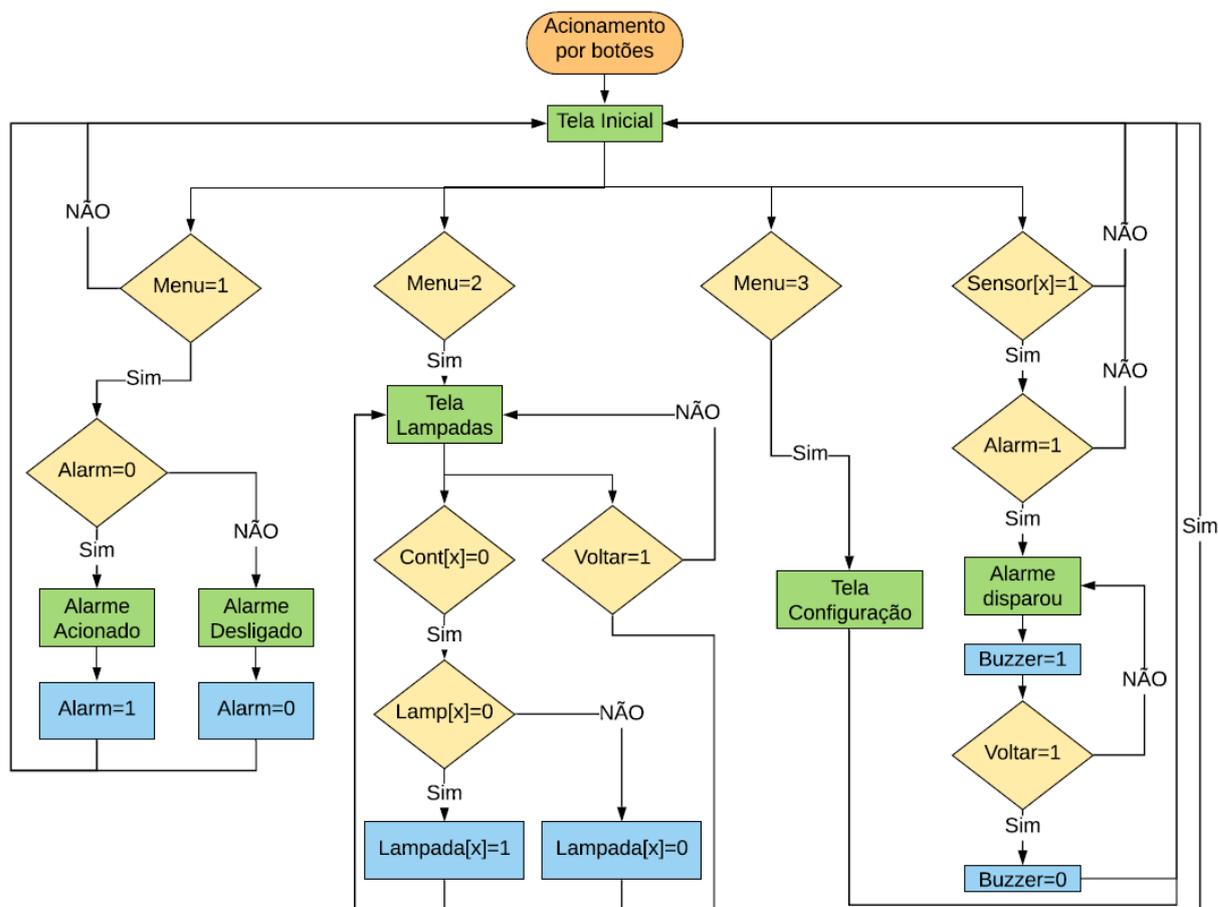
3.2.1 Lógica de programação para acionamento por botões

A primeira etapa foi desenvolver uma lógica de programação responsável por realizar os acionamentos dos LED's e do alarme através de botões físicos. Para isso, elaborou-se um fluxograma contendo as principais etapas do programa. A Figura 18 apresenta o fluxograma desenvolvido.

Inicialmente o fluxograma mostra uma tela inicial que contém as opções de acionamento de alarme, controle das lâmpadas e uma opção de configuração para possíveis ajustes de *login* e senha. Ao selecionar a primeira opção, o alarme pode ser acionado (possibilitando a leitura do sensor de presença) ou desligado. A segunda opção fornece uma tela contendo as lâmpadas programadas e permite o controle dessas, habilitando ou desabilitando cada uma através de um botão de seleção. No fluxograma esta etapa está apresentada de forma genérica utilizando a variável *x*, sendo que *x* pode assumir valores de 1 até o número máximo de lâmpadas. Por fim, a terceira opção leva a um menu de configuração, que permite a alteração do nome do usuário e a senha (implementação na etapa de acionamento por Wi-Fi). Na rotina

principal desenvolvida, o programa verifica se algum sensor está acionado e se a opção de alarme está habilitada. Caso as duas informações sejam verdadeiras, o *buzzer* é acionado, uma mensagem de disparo é apresentada e se faz necessário apertar a tecla voltar para desligar o alarme.

Figura 18 – Fluxograma do acionamento por botões



Fonte: O autor (2021)

Com o Fluxograma elaborado, o código de programação foi desenvolvido através do programa MPLABX IDE v5.00, utilizando a linguagem de programação C. O programa foi estruturado em inclusão de bibliotecas (configuração, LCD, etc., ...), definições para facilitar a programação (entradas nomeadas de acordo com as funções dos botões), declaração das funções auxiliares como as mensagens apresentadas no *Display* LCD, definição das variáveis, o programa principal e as funções auxiliares. A Figura 19 apresenta a estruturação utilizada no código. A lógica de programação para essa etapa pode ser observada no Apêndice A.

Figura 19 – Estrutura de programação utilizada

```

//-----Bibliotecas-----
#include "config.h"
#include "lcd4_4L.h"
//-----Definicoes-----
#define CIMA RB0
#define BAIXO RB1
//-----Declaracao de funcoes-----
void TELA_INICIAL(int POS);
void TELA_LAMP(int POS);
//-----Variaveis globais-----
int CONT;
char lamp[4];
//-----Programa principal-----
void main(void)
{
    .
    .
    .
}
//-----Funções-----

void TELA_INICIAL(int POS) //MOSTRA MENSAGENS DO MENU CONFIGURACAO
{
    limpa_lcd4();
    .
    .
    .
}

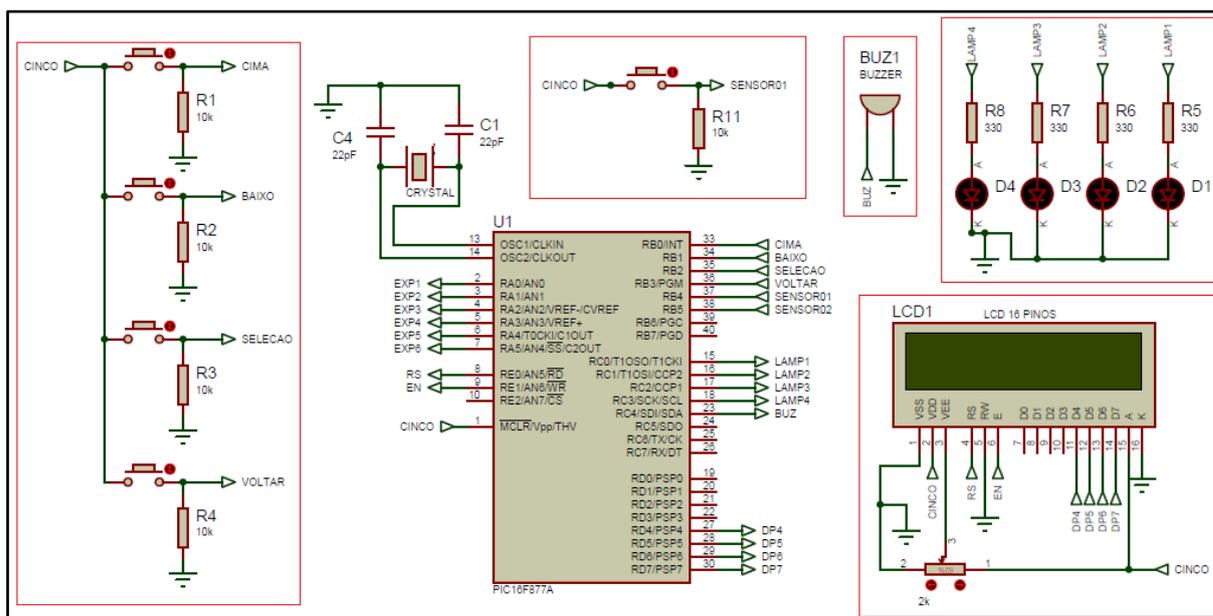
```

Fonte: O autor (2021)

Para auxiliar no desenvolvimento do código e verificar se a programação estava correta, foi utilizado o arquivo .hex, gerado pelo MPLABX IDE v5.00 para realizar a simulação do PIC16F877A e dos demais componentes através do *software* Proteus 8. Os componentes inseridos na simulação foram o PIC16F877A, seis *push buttons* (que representavam os botões de navegação do menu para cima e para baixo, o botão seleção, o botão voltar, o sensor de presença para simular o acionamento do alarme e o botão para alterar para o modo manual), o *buzzer* (representando o alarme), quatro LED's (representando as lâmpadas de uma casa), e um *Display* LCD, para apresentar as mensagens inseridas nos menus.

A Figura 20 mostra o projeto de simulação elaborado no *software* Proteus 8 com todos os componentes necessários para o teste dessa etapa. Para facilitar na visualização, organizou-se o projeto de modo que é possível observar os acionamentos a esquerda, o microcontrolador ao centro e os periféricos a direita indicando os sinais visuais e sonoros ao usuário sobre o funcionamento do programa.

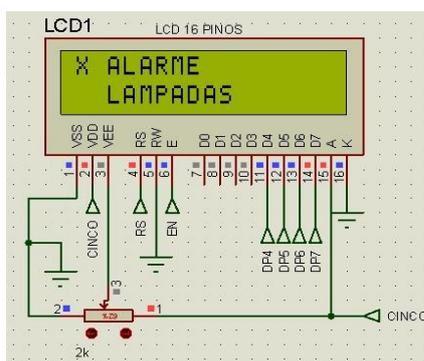
Figura 20 – Simulação da primeira etapa no Proteus 8



Fonte: O autor (2021)

A primeira etapa desenvolvida no código de programação foi a criação do menu principal do programa com uma opção de seleção, contendo as opções de alarme, lâmpadas e configuração. Essa foi verificada na simulação através do funcionamento do *display* LCD. Para isso, foi utilizado uma função para informar o texto que seria apresentado no *display*, a linha em que a mensagem seria mostrada (linha 0 ou linha 1) e a posição em x no *display* (valores de 0 a 15). Também foi criada uma variável denominada de “POS”, em que seria colocado um “X” na frente de cada opção quando a mesma estivesse selecionada. Foi elaborado uma rotina para fazer com que “X” alterasse de posição e uma função de limpeza de LCD a cada alteração de posição de “X”. A Figura 21 apresenta o menu principal criado e testado através do Proteus 8.

Figura 21 – Menu principal

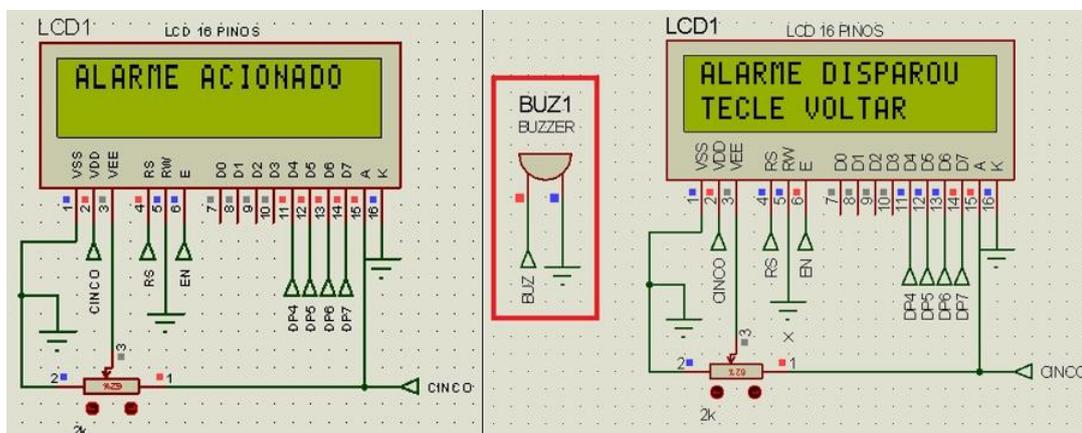


Fonte: O autor (2021)

Após criar o menu principal, foi desenvolvido a lógica de programação para o alarme. Ao selecionar a opção alarme no menu principal uma mensagem de “alarme acionado” era informada no *display* LCD ao usuário, habilitando assim o *buzzer*. Caso o sensor fosse acionado, era enviando um nível lógico alto para as entradas do PIC, a mensagem de “alarme disparou” era informada ao usuário simultaneamente ao disparo do *buzzer*. Ao clicar no botão voltar, o usuário desabilitava o *buzzer* e o sistema retornava ao estado de monitoramento. Para desabilitar o alarme bastava selecionar a opção alarme novamente no menu principal.

A Figura 22 (esq.) apresenta a mensagem de alarme acionado e a Figura 22 (dir.) apresenta o momento em que o alarme dispara, devido estar habilitado e com nível lógico alto recebido do sensor.

Figura 22 – Mensagens da função alarme

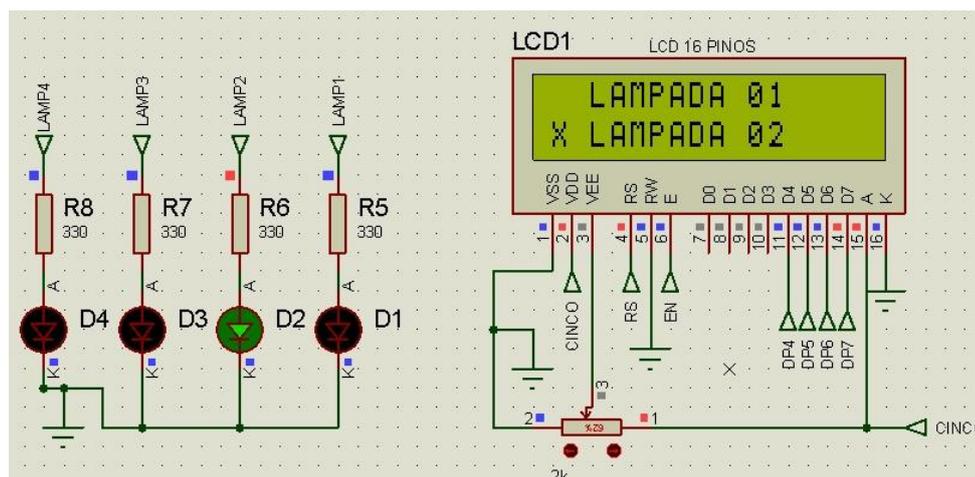


Fonte: O autor (2021)

A segunda opção desenvolvida no menu principal, foi a opção “lâmpadas”. Ao selecionar a opção lâmpadas no menu principal, o usuário era direcionado a um novo menu contendo todas as lâmpadas cadastradas. Ao navegar com os botões “para cima” e “para baixo” era possível acessar qualquer lâmpada e acioná-la com um clique no botão seleção. Ao clicar novamente a lâmpada era desligada. Como o *Display* LCD utilizado apresenta apenas duas linhas, ao selecionar a opção para baixo o menu atualiza demonstrando as outras lâmpadas cadastradas.

A Figura 23 demonstra o procedimento para a lâmpada 02, acionando-a através de um clique no botão seleção, após entrar na opção lâmpadas no menu principal.

Figura 23 – Mensagens da função lâmpadas



Fonte: O autor (2021)

3.2.2 Comunicação entre o PIC16F877A e o ESP-8266

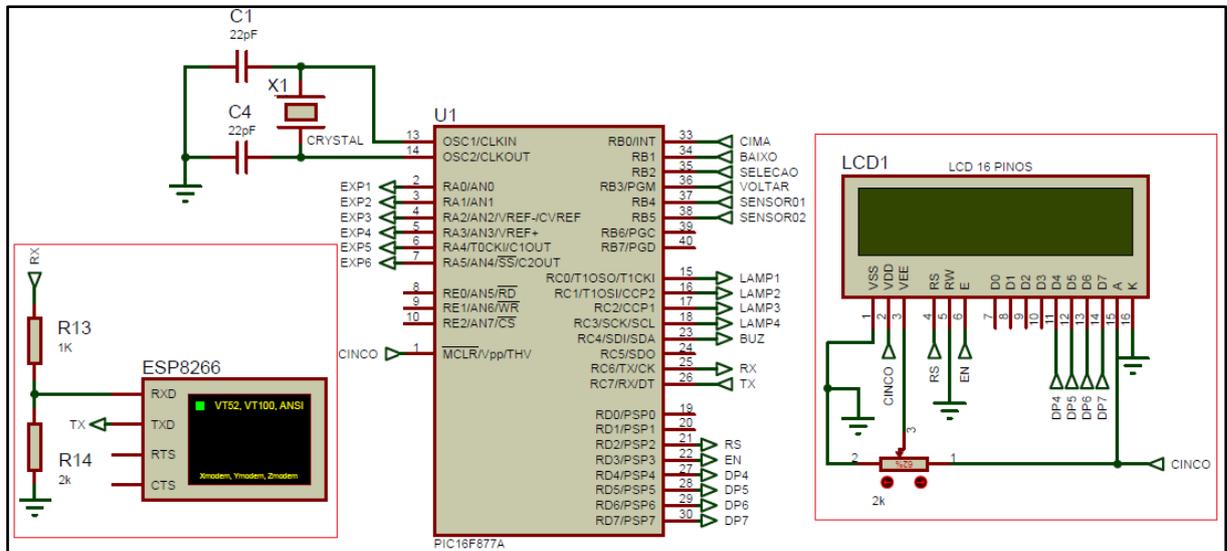
Após efetuar o teste do programa via botões, foi necessário elaborar uma lógica de programação responsável por efetuar a comunicação entre o PIC16F877A e o ESP-8266. Para isso, utilizaram-se os comandos AT do ESP8266 presentes no guia do fabricante (Espressif Systems, 2018). A primeira etapa da programação, foi criar uma função para enviar o comando “AT” para o ESP, a fim de receber a resposta “OK” do ESP, para informar que o ESP estava conectado. Em seguida utilizou-se o comando “AT+CWMODE”, para definir o modo Wi-Fi que o ESP deveria trabalhar (1 – *Station* ou 2 – Modo AP). Por fim, foi utilizado o comando “AT+CWSAP” para definir as configurações do ESP no modo AP, como *login*, senha, ID do canal e o método de criptografia. Todos os comandos utilizados apresentavam como característica fornecer uma resposta “OK” para o código, a fim de verificar se as etapas estavam sendo cumpridas.

Assim como na etapa anterior, utilizou-se o *software* Proteus 8 para efetuar a simulação da comunicação entre o PIC16F877A e o ESP-8266. Para essa etapa foi utilizado o microcontrolador PIC16F877A juntamente com o arquivo .hex obtido no MPLABX IDE, o *display* LCD para visualizar as mensagens de configuração da comunicação entre PIC e ESP, e um terminal virtual com o objetivo de simular o ESP-8266 e enviar as respostas para o PIC.

A Figura 24 mostra o projeto de simulação elaborado no *software* Proteus 8 para a simulação da comunicação do ESP com o PIC, com o terminal virtual, utilizando

os conectores RX (para recebimento de informações) e TX (para o envio dos comandos ao PIC).

Figura 24 – Simulação da comunicação entre o PIC16F877A e o ESP-8266 através dos comandos AT



Fonte: O autor (2021)

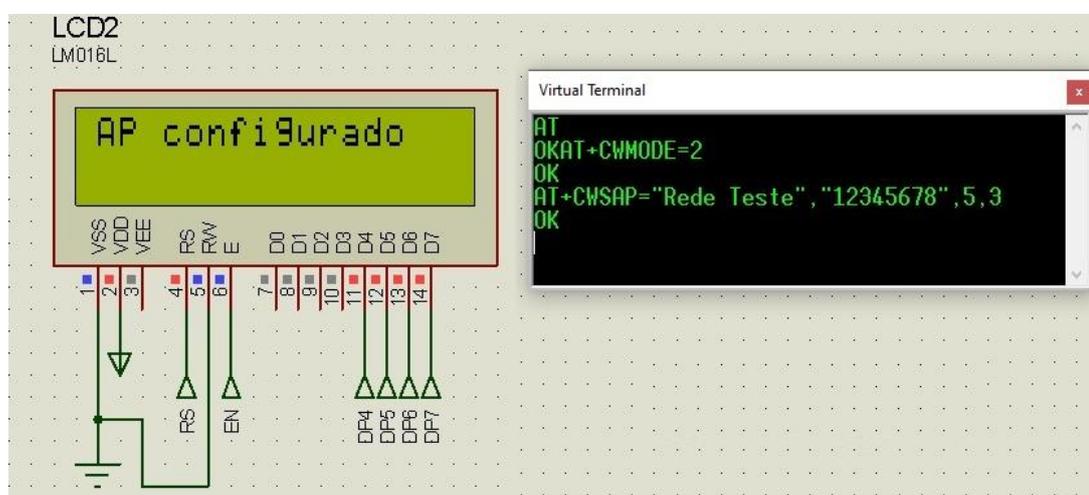
Ao iniciar a simulação, foi verificado que o PIC enviou o comando “AT” para o terminal virtual (Representa o ESP-8266). Assim como apresenta o guia utilizado, o ESP ao receber o comando AT retornou a resposta “OK” (inserido manualmente no terminal). Com essa resposta, uma mensagem de ESP conectado foi inserida no *display*, para informar ao usuário que o ESP estava conectado no PIC. Caso a resposta não fosse enviada pelo ESP, uma mensagem de ESP desconectado apareceria no *display*, informando ao usuário um erro na conexão entre o ESP e o PIC. Com isso, o programa ficaria em um *loop* aguardando a conexão para prosseguir.

Em seguida com o ESP reconhecido pelo PIC, o programa executou a segunda função descrita no código de programação, responsável por utilizar o comando “AT+CWMODE=2”. Esse comando foi responsável por configurar o Wi-Fi do ESP no modo de ponto de acesso (AP), permitindo que o ESP se conectasse a qualquer ponto de Wi-Fi, como um celular, um computador ou um notebook, criando assim, uma rede local. Assim como apresentado no guia, ao executar esse comando o ESP retornou um “OK”, que foi inserido manualmente no terminal virtual para simular o comportamento do ESP.

Com o comando “OK” recebido, o programa executou a terceira função,

utilizando o comando “AT+CWSAP”. Foi definido na programação uma rede chamada de “Rede Teste”, uma senha de oito caracteres “12345678”, o ID do canal como “5” e o método de criptografia “3” (WPA2_PSK). Novamente, um “OK” foi inserido no terminal virtual simulando a resposta do ESP para essa configuração. Com isso, uma mensagem no *display* foi fornecida ao usuário informando que o ESP foi configurado corretamente e uma rede foi criada. A Figura 25 demonstra as etapas descritas anteriormente para a configuração do ESP-8266 no modo AP, através do terminal virtual, criando uma rede teste com uma senha. No *display* é possível verificar a mensagem “AP configurado” informando que a configuração foi um sucesso. A estrutura de programação para a etapa de comunicação entre o PIC e o ESP pode ser encontrada no Apêndice A e as funções utilizadas no Apêndice B do trabalho.

Figura 25 – Testes da comunicação entre o PIC16F877A e o ESP-8266 através dos comandos AT



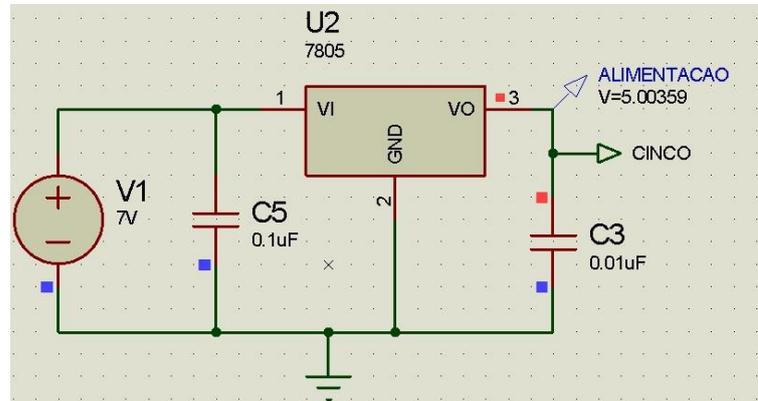
Fonte: O autor (2021)

3.2.3 Fontes de alimentação dos circuitos

Com o objetivo de fornecer uma tensão constante de alimentação para o microcontrolador, o *display* e os botões, elaborou-se uma fonte de tensão contínua com saída fixa de 5V. Para isso, utilizou-se um regulador de tensão LM7805, visto que assim seria possível alimentar o sistema com uma tensão de entrada de 7 a 20V. Assim, qualquer fonte comercial de fácil acesso poderia ser utilizada para alimentar o sistema. Também foram adicionados dois capacitores cerâmicos (entrada e saída), com o objetivo de atuarem como filtros e eliminar ruídos no sistema. A Figura 26 demonstra a simulação da fonte de alimentação com saída de 5V, através do *software*

Proteus 8, em que é possível verificar um valor de 5,00359 V para a saída a partir de uma tensão de entrada de 7V.

Figura 26 – Simulação da fonte de alimentação com saída fixa de 5V



Fonte: O autor (2021)

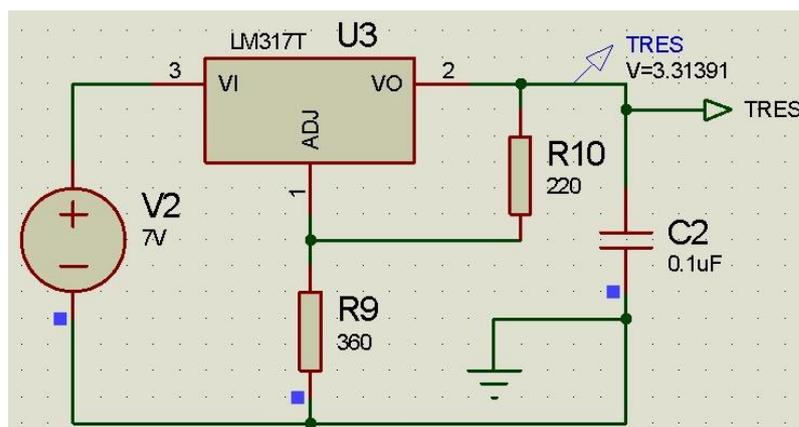
Para a alimentação do ESP-8266, foi necessário elaborar uma fonte de tensão contínua com saída fixa em 3,3 V. Para isso, utilizou-se o regulador de tensão LM317T, o qual permite ajustar a tensão de saída de 1,2 V a 37 V através dos valores dos resistores. Os valores dos resistores utilizados foram de 220 Ω para R1 e 360 Ω para R2. A parcela referente a corrente adjacente em R2 é desprezível, devido ao valor baixo utilizado para o resistor R2 e a aplicação que está sendo utilizado. A Equação (1) é utilizada para efetuar os cálculos. Ao substituir os valores das resistências em (2) tem-se o valor aproximado para a tensão de 3,30 V.

$$V_o = 1,25 \cdot \left(1 + \frac{R_2}{R_1}\right) + I_{adj} \cdot R_2 \quad (1)$$

$$V_o = 1,25 \cdot \left(1 + \frac{360}{220}\right) + 0 \rightarrow V_o \cong 3,30V \quad (2)$$

Para a entrada do regulador, foi utilizado uma tensão de 7 V (mesma alimentação da fonte de 5 V, visto que as duas fontes foram projetadas para a mesma entrada na placa do circuito), dado que o mesmo suporta valores de 3,3 V a 40 V. Também foram adicionados dois capacitores cerâmicos (entrada e saída) com o objetivo de filtrar o sinal. A Figura 27 apresenta a simulação para a fonte utilizando o regulador de tensão LM317T, em que foi obtido uma tensão contínua de saída de 3,31391 V.

Figura 27 – Simulação da fonte de alimentação com saída fixa de 3,3V



Fonte: O autor (2021)

3.2.4 Diagrama do projeto

Após os testes de acionamento por botões e o teste de comunicação entre o PIC e o ESP, desenvolveu-se o diagrama do projeto contendo os componentes e periféricos necessários para a elaboração do projeto e confecção da placa de circuito impresso (PCI). Também foram inseridos no diagrama dois bornes de 3 vias com o objetivo de expandir o projeto no futuro, através da adição de uma leitura analógica de temperatura, o acionamento de um portão eletrônico ou outras funcionalidades desejadas. A Tabela 3 apresenta as entradas/saídas do PIC16F877A utilizadas, com suas respectivas funções e com os devidos componentes inseridos.

Tabela 3 – Entradas/Saídas do PIC16F877A com os devidos componentes inseridos

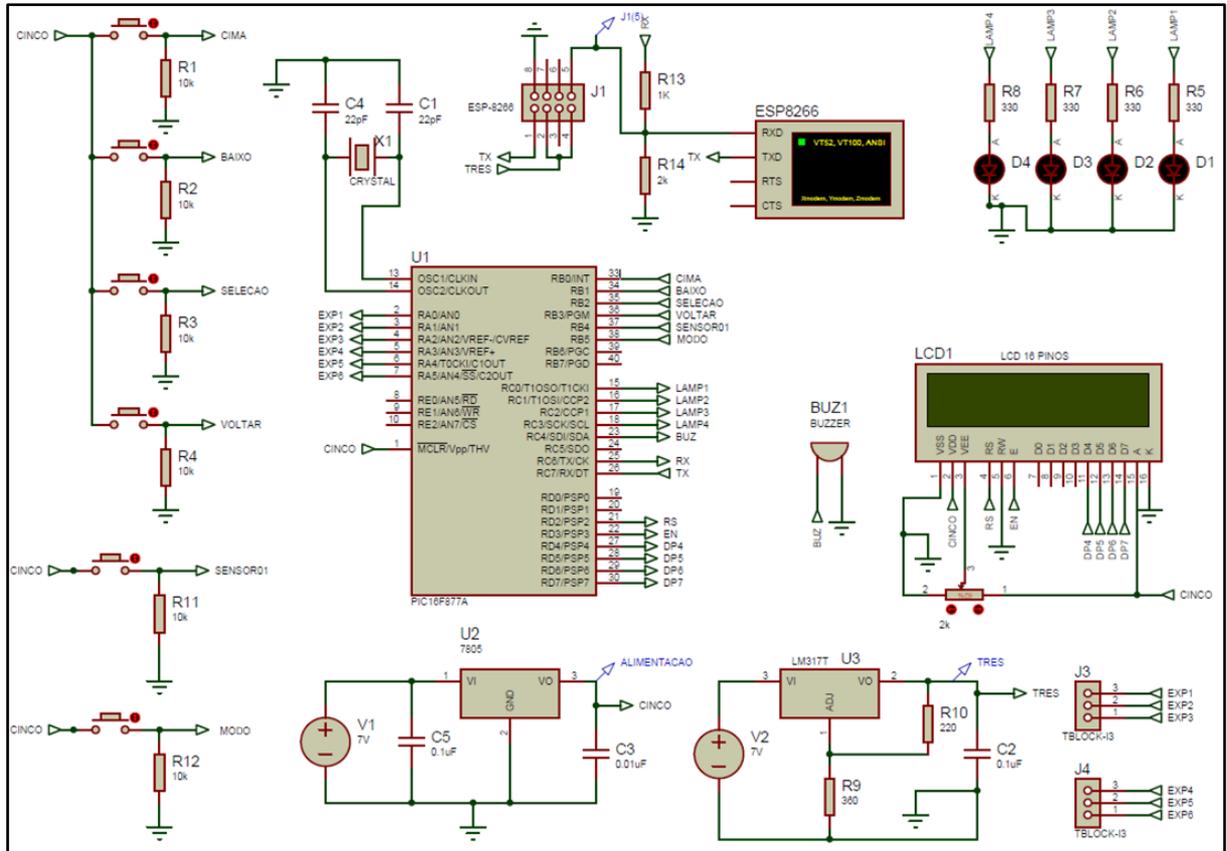
Pino	Função	Componente
RA0 – RA5	Expansão do projeto (relés, sensor temperatura)	Bornes 3 vias
RB0 – RB3	Comando do alarme e das lâmpadas via botões	<i>Push Button</i>
RB4	Sensor do alarme	<i>Push Button</i>
RB5	Botão de alteração do modo WEB para manual	<i>Push Button</i>
RC0 – RC3	Simulação das lâmpadas	LED's
RC4	Simulação do alarme	<i>Buzzer</i>
RC6 – RC7	Comunicação com o ESP-8266	ESP-8266
RD2 – RD7	Interface com o usuário	<i>Display LCD</i>

Fonte: Elaborado pelo autor, 2021.

O diagrama para a elaboração da PCI foi desenvolvido no *software* Proteus 8. Devido ao *software* não apresentar o componente ESP-8266, foi inserido um bloco genérico de 8 pinos (4x2) para a confecção da placa e foi utilizado um terminal virtual na simulação. O *display* LCD de 14 pinos foi adaptado para um *display backlight* 16 pinos, assim seria possível utilizar os dois modelos após confecção da placa. Os

demais componentes utilizados no projeto foram obtidos na biblioteca do *software*. A Figura 28 apresenta o diagrama do projeto desenvolvido.

Figura 28 – Diagrama completo do projeto



Fonte: O autor (2021)

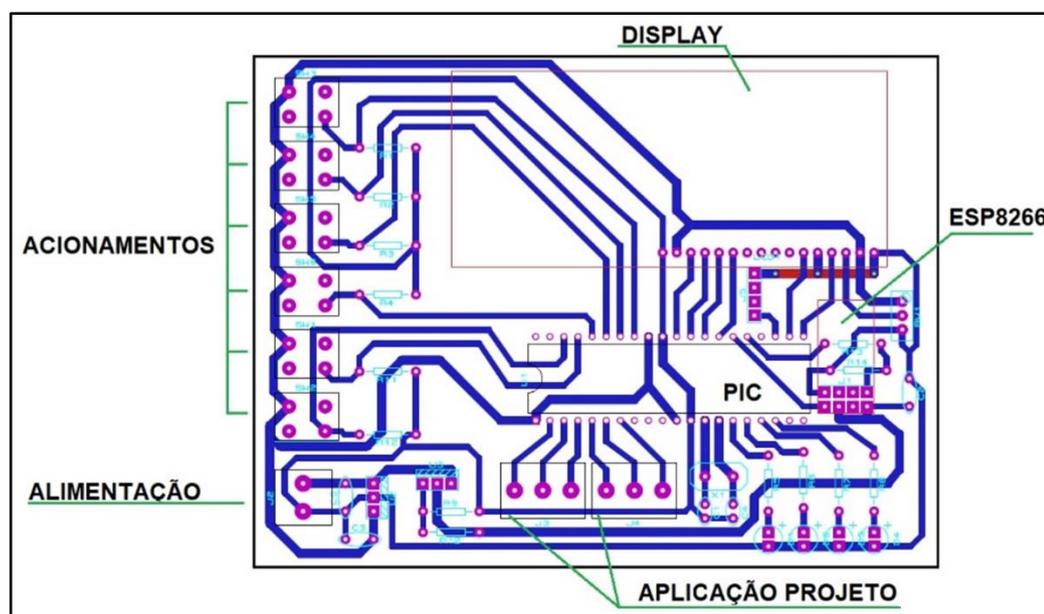
3.2.5 Desenvolvimento da placa

Para o projeto da Placa de Circuito Impresso, também foi utilizado o *software* Proteus 8. O circuito impresso foi gerado na aba ARES do *software* a partir do diagrama elétrico elaborado na aba ISIS. Para o desenvolvimento do *layout*, inicialmente posicionou-se o PIC16F877A ao centro da PCI e adicionaram-se os periféricos (*display*, LED's, botões) de modo com que a criação das trilhas fosse facilitada, evitando o cruzamento dessas. O *display* também foi posicionado de modo que todo o visor ficasse localizado sobre a área da placa. A alimentação e os bornes de expansão foram posicionados próximo as bordas da placa para evitar que as ligações interferissem nos componentes.

Após posicionar todos os componentes, foi configurado no *software* a opção

Bottom copper, com o objetivo de criar trilhas apenas na parte inferior da placa. As trilhas de alimentação (5 V e 3,3 V), foram configuradas com uma largura de trilha T60, enquanto as trilhas de terra e as demais trilhas foram configuradas com uma largura T40. Em seguida, utilizou-se a opção *Auto router* para elaborar as trilhas. Devido as limitações de geometria e posição, algumas trilhas não foram geradas, pois acarretariam em cruzamentos de trilhas. Para solucionar esse problema, o local das trilhas foi alterado de forma manual e alguns componentes foram realocados. Devido ao número de portas elevadas utilizadas no PIC, foi necessário realizar dois *jumpers* para os pinos de terra no *display*. A Figura 29 demonstra o *layout* final com as trilhas e furações da placa de circuito impresso, em que é possível observar o posicionamento dos componentes, as trilhas elaboradas (azul) e os *jumpers* necessários (vermelho).

Figura 29 – *Layout* da placa de circuito impresso

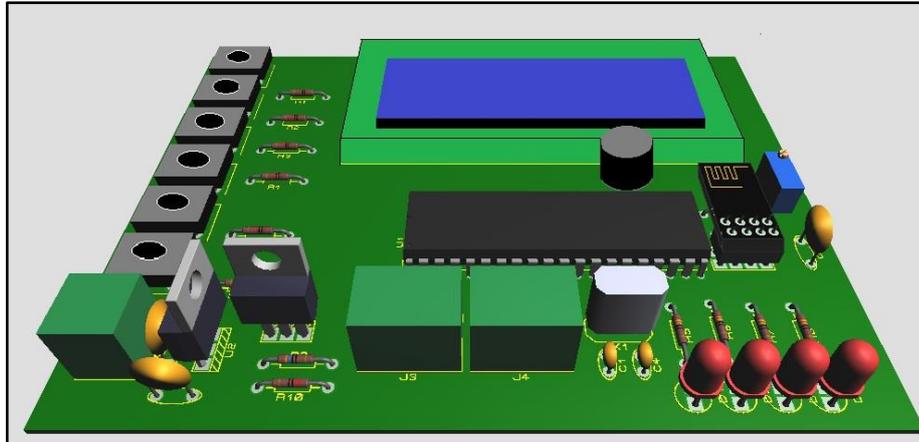


Fonte: O autor (2021)

Com o objetivo de visualizar o posicionamento de cada componente e verificar se esses não estariam ocupando o mesmo espaço físico inviabilizando a fabricação da placa, foi utilizado a ferramenta de visualização 3D. Devido ao *software* não apresentar alguns componentes, foi utilizado um *software* de modelamento 3D para criar esses componentes de modo simplificado, adicionando-os ao *software* Proteus 8.

A Figura 30 mostra o modelo 3D gerado pelo programa, em que é possível verificar que não há componentes sobrepostos.

Figura 30 – Visualização 3D da placa de circuito impresso



Fonte: O autor (2021)

3.2.6 Desenvolvimento das páginas de comando via WEB

O desenvolvimento das páginas WEB para o controle e automação do projeto, foi realizado a partir do programa Sublime Text, visto que esse editor de código fonte possui bibliotecas que auxiliam na programação da linguagem HTML. A primeira etapa, foi criar um arquivo e salvar no formato HTML. Ao salvar um arquivo nesse formato e digitar na primeira linha o código `<html>`, o programa gerou o código base para o desenvolvimento da página, gerando os campos `html`, `head` e `body`.

O campo `html` gerado foi utilizado para delimitar o programa, marcando o início e o término do código de programação da página. No campo `head`, foram inseridas informações sobre o cabeçalho de cada página, como o título utilizado e o meta da página. Foi utilizado a opção “`<meta http-equiv="refresh" content="2">`”, visto que é muito utilizada em páginas que necessitem atualizar automaticamente, como é o caso da página do alarme que necessita receber a informação da condição do monitoramento. A opção “`content`” especificou o intervalo de tempo de cada atualização em segundos.

No campo `body`, foi desenvolvido o corpo da página, como os textos e mensagens apresentados ao usuário, os hiperlinks para acionamentos dos componentes (lâmpadas e alarme) e trocas de página, as cores do texto e do fundo da página e outras informações necessárias para o funcionamento da página. A Figura

31 apresenta o código base de forma genérica, utilizado para a programação das páginas de menu principal, lâmpadas e alarme, no editor Sublime Text. As páginas criadas através do editor estão apresentadas nos resultados do trabalho, em conjunto com as imagens de acionamento da placa. O código desenvolvido para as páginas está presente em funções (envia_página) do Apêndice B do trabalho.

Figura 31 – Código base para programação das páginas WEB



```
C:\Users\kleison\Desktop\Eng Eletrica\TCC ifsc\ex4.html - Sublime Text
File Edit Selection Find View Goto Tools Project Preferences Help

ex1.html
1 <!DOCTYPE html>
2 <html>
3
4 <head>
5
6 <title></title>
7
8 </head>
9
10 <body>
11
12 </body>
13
14 </html>
15
```

Fonte: O autor (2021)

3.2.7 Configuração do acesso externo

Com o objetivo de tornar o projeto acessível de qualquer lugar do mundo, utilizou-se um serviço de DDNS. Para isso, criou-se uma conta através do site “www.noip.com/”, no qual foi associado o endereço IP referente ao roteador em que o ESP estava conectado ao domínio automaçãok13.ddns.net. Foi utilizado o plano gratuito fornecido pelo site NO-IP, o qual permitia a adição de até 3 domínios ativos por um prazo de 30 dias. Após esse período era necessário realizar o cadastro do domínio novamente. Ao acessar o endereço cadastrado (automacaok13.ddns.net), o usuário seria redirecionado ao endereço IP do roteador (192.168.0.26), que por sua vez, iria direcionar ao ESP conectado a um endereço IP (192.168.1.150) e uma porta específica (80) cadastrada nas configurações do roteador.

O serviço de DDNS também permitiu transformar o IP dinâmico do modem, fornecido pela provedora de internet, em um domínio fixo, através do Cliente de

Atualização Dinâmica, um aplicativo fornecido pelo site que verificava frequentemente as mudanças de endereço IP e os atualizava automaticamente. Assim, mesmo que o modem fosse reiniciado e o endereço IP fornecido pela provedora fosse alterado, o DUC atualizaria o novo IP ao domínio informado. A Figura 32 mostra o domínio criado no site NO-IP, o qual foi inserido o endereço automacaok13.ddns.net ao nome do host e o endereço IP 192.168.0.26 no campo IP/Alvo referente ao endereço do roteador utilizado. Também é possível verificar a data e o horário de cadastro e a informação de que os hosts vencem a cada 30 dias. As informações básicas apresentadas, como e-mail, senha e nome de usuário foram utilizadas posteriormente nas configurações do roteador.

Figura 32 – Criando um domínio no site NO-IP

The screenshot displays the NO-IP dynamic DNS management interface. The main content area shows a progress indicator for '1 of 3' hostnames, a 'Criar Hostname' button, and a search bar. Below this is a table of hostnames:

Nome de Host	Last Update	IP / Alvo	Type
automacaok13.ddns.net Vencerá em 19 dias	Jul 7, 2021 15:10 BRT	192.168.0.26	A

Below the table is a 'Basic Info' section with the following details:

- Email: kleisonpossenti@hotmail.com
- Senha: [masked]
- Nome de usuário: kleison1313

Fonte: O autor (2021)

Para a conexão do ESP a internet, utilizou-se um roteador da marca TP-Link, modelo TL-WR829N, com uma velocidade de 300 Mbps e duas antenas Wi-Fi de 5 dBi. Esse roteador foi escolhido devido apresentar um baixo custo no mercado, ser de fácil obtenção e de sua marca ser a mais utilizada no mundo. Esse modelo possui os modos de operação como roteador, repetido de sinal, Access Point e WISP, o que supre as necessidades da aplicação e utiliza o padrão *Wireless* IEEE 802.11b/g/n 2.4

GHz, compatível com o ESP-8266. Além disso, sua configuração possui uma interface Web intuitiva em diversos idiomas, inclusive o português, o que facilita a configuração do mesmo. A Figura 33 apresenta uma imagem ilustrativa do roteador TL-WR829N utilizado para a elaboração do projeto.

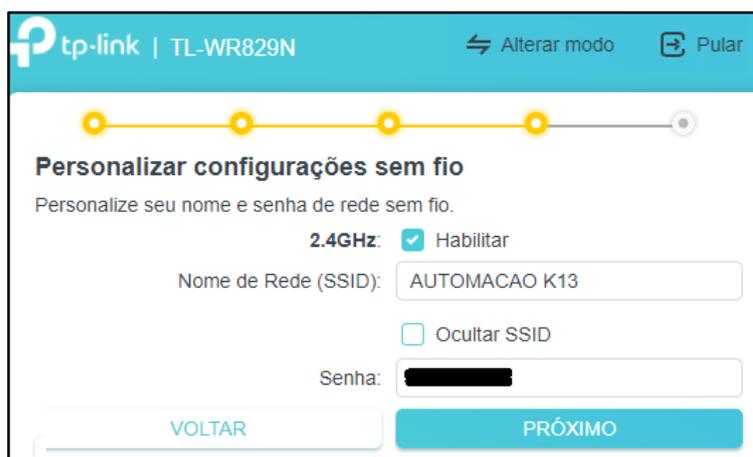
Figura 33 – Roteador TP-Link TL-WR829N



Fonte: TP-Link (2021)

Inicialmente efetuou-se a configuração rápida do roteador TL-WR829N para o modo Access Point, transformando a rede com fio fornecida pelo modem em uma rede sem fio. Assim, seria possível conectar o ESP-8266 ao roteador através da conexão Wi-Fi. Para isso, selecionou-se o modo Access Point nas configurações iniciais do roteador, inserido o tipo de conexão desejada e criado uma rede com o nome “AUTOMACAO K13”, com uma senha de oito dígitos. A Figura 34 apresenta a personalização da rede sem fio criada, com o nome do usuário e senha.

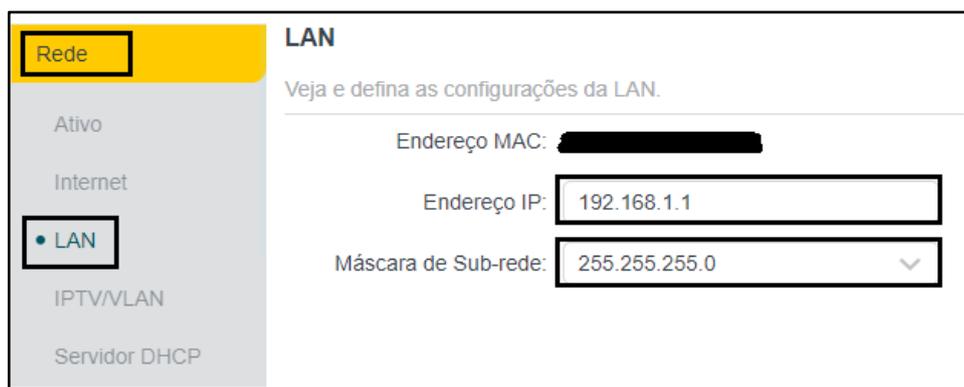
Figura 34 – Configuração inicial do roteador

A imagem é uma captura de tela da interface web de configuração do roteador TP-Link TL-WR829N. No topo, há o logotipo da TP-Link e o modelo do dispositivo. Abaixo, há uma barra de progresso com cinco pontos, onde o primeiro ponto está preenchido. O título da seção é "Personalizar configurações sem fio". Abaixo do título, há uma instrução: "Personalize seu nome e senha de rede sem fio." Há uma opção "2.4GHz:" com um checkbox "Habilitar" selecionado. Abaixo disso, há um campo de texto para "Nome de Rede (SSID):" com o valor "AUTOMACAO K13". Há também um checkbox "Ocultar SSID" desselecionado. Abaixo, há um campo de texto para "Senha:" com uma senha mascarada por pontos. Na base da tela, há dois botões: "VOLTAR" e "PRÓXIMO".

Fonte: O autor (2021)

Após efetuar as configurações iniciais, realizaram-se as configurações de rede. Para isso, foi selecionado a opção Rede no menu principal e depois a opção LAN. O endereço IP padrão fornecido pela fabricante era de 192.168.0.1. O Endereço IP da LAN foi redefinido para 192.168.1.1, visto que foi verificado que o endereço IP da LAN do modem estava configurado pela provedora de internet em 192.168.0.1. A máscara de sub-rede utilizada foi a da classe C, visto que o modem adotava o mesmo padrão, sendo 255.255.255.0 permitindo a conexão de até 256 Hosts ao roteador. A Figura 35 apresenta a configuração LAN para o roteador utilizado.

Figura 35 – Configuração LAN



Rede	LAN
Ativo	Veja e defina as configurações da LAN.
Internet	Endereço MAC: [REDACTED]
• LAN	Endereço IP: 192.168.1.1
IPTV/VLAN	Máscara de Sub-rede: 255.255.255.0
Servidor DHCP	

Fonte: O autor (2021)

A segunda configuração de rede ajustada, foi em relação ao *Dynamic Host Configuration Protocol* (DHCP), ou seja, o intervalo de endereços de rede de modo dinâmico. Para isso, selecionou-se a opção servidor DHCP, também disponível no menu Rede. Devido a alteração do endereço IP da LAN do roteador, o intervalo foi reajustado para um IP inicial de 192.168.1.100 até um endereço IP final de 192.168.1.199, sendo que o tempo de concessão do endereço foi mantido padrão. Com isso todo dispositivo conectado ao roteador teria seu IP atribuído automaticamente a partir do endereço 192.168.1.101 em diante até o ultimo endereço disponível (192.168.1.198). A Figura 36 apresenta as configurações realizadas para o protocolo de configuração dinâmica de endereços de rede, com o intervalo de endereçamento definido e o tempo de verificação para averiguar se aquele dispositivo ainda está conectado ao roteador.

Figura 36 – Configuração do Servidor DHCP

Servidor DHCP

Atribua dinamicamente endereços IP aos dispositivos conectados ao roteador.

Servidor DHCP: Habilitar

Pool Address IP: 192.168.1.100 - 192.168.1.199

Tempo de Concessão de Endereço: 120 minutos

Fonte: O autor (2021)

Por fim, a última configuração de rede realizada foi a configuração do nome de domínio fixo, mais conhecido como *Domain Name System* (DNS). Para isso foi selecionado a opção DNS dinâmico presente no menu Rede. Os roteadores da marca TP-Link apresentam suporte para os provedores de serviço NO-IP e DynDNS, porém a primeira opção permite a criação de um domínio por um período de 30 dias de forma gratuita e é possível renova-lo após esse período, enquanto a segunda permite apenas um teste grátis de 7 dias, sendo necessário adquirir os serviços pagos após esse período. Sendo assim o provedor de serviço selecionado foi a opção NO-IP.

Em seguida, foi necessário preencher as informações cadastradas no site NO-IP, como o nome de usuário, a senha utilizada para criação da conta, o domínio cadastrado e habilitar a opção de vínculo IP WAN permitindo associar o endereço de IP do roteador ao domínio cadastrado. A Figura 37 apresenta as configurações realizadas para o DNS dinâmico, em que é possível verificar na opção ativo que o domínio está conectado.

Figura 37 – Configuração do DNS

DNS Dinâmico

Atribua um nome de host fixo (nome de domínio) para acesso remoto ao seu dispositivo, site ou servidor atrás do roteador.

Provedor de Serviço: NO-IP [Registre-se agora](#)

Nome de usuário: kleison1313

Senha:

Nome de Domínio: automacaok13.ddns.net

Vínculo IP WAN: Habilitar

Ativo: Conectado

LOGIN E SALVAR

Fonte: O autor (2021)

Com todas as configurações de rede definidas e o intervalo IP atribuído, foi possível realizar a próxima etapa, o redirecionamento de porta. Para isso, selecionou-se a opção Direcionamento NAT (também encontrada como *forwarding* em alguns roteadores) e em seguida a opção Redirecionamento de Porta (*Virtual Servers*). Ao selecionar a opção adicionar, uma tela contendo nome de serviço, endereço IP, porta interna e externa e protocolo era exibida. O serviço selecionado foi a opção de *HyperText Transfer Protocol* (HTTP) em virtude das páginas WEB criadas para o projeto, e por consequência a porta interna e externa foram definidas como 80. O endereço IP do dispositivo ESP foi configurado como 192.168.1.150, a fim de evitar que nenhum dispositivo conectado ao roteador atingisse essa porta devido ao protocolo DHCP. Por fim, selecionou-se a opção de habilitar a porta e a configuração foi salva, direcionando a porta do roteador para o endereço IP do ESP. A Figura 38 apresenta a etapa de redirecionamento de porta com a janela da opção adicionar.

Figura 38 – Configuração de redirecionamento de porta

Direcionamento NAT

Redirecionamento de Porta

Porta de Disparo

UPnP

DMZ

Controle dos Pais

Redirecionamento de Porta

Especifique portas para tornar acessíveis dispositivos ou serviços específicos na sua rede local pela Internet.

Nome de Serviço: HTTP

VER SERVIÇOS COMUNS

Endereço IP do dispositivo: 192.168.1.150

VER DISPOSITIVOS CONECTADOS

Porta Externa: 80

Porta Interna: 80

Protocolo: Tudo

Habilitar Esta Entrada

CANCELAR SALVAR

+ Adicionar

Fonte: O autor (2021)

Com o host criado e com a configuração do roteador concluída, foi necessário iniciar a configuração no ESP. Para que o ESP fosse conectado à rede criada, foi utilizado uma função com o comando “AT+CWMODE=1” para que o mesmo trabalhasse em modo estação (*Station*). Em seguida as informações de nome da rede (AUTOMACAO K13) e senha do roteador foram informadas através de outra função com o comando AT+CWJAP. Por fim, uma função com o comando CIPSTA_CUR foi utilizada para informar ao ESP o seu endereço IP fixo, o Gateway da rede e a máscara de sub-rede. A Figura 39 apresenta a função que fornece ao ESP os parâmetros

necessários para conectar à rede. Nela é possível verificar o endereço IP fixo do ESP (192.168.1.150), conforme definido no redirecionamento de porta do roteador, o Gateway padrão (192.168.1.1) e a máscara de sub-rede (255.255.255.0) conforme apresentado nas configurações LAN do roteador.

Figura 39 – Função para informar os parâmetros de rede para o ESP

```

//Conecta ao IP fixo, Gateway e Mascara da sub-rede
esp8266_enderecoIP("192.168.1.150", "192.168.1.1", "255.255.255.0");
texto_lcd4_4L ((char*)"Endereco OK", 0, 0);

void esp8266_enderecoIP(unsigned char* IP, unsigned char* gateway, unsigned char* mascara)
{
    _esp8266_print("AT+CIPSTA_CUR=\");
    _esp8266_print(IP);
    _esp8266_print("\",\");
    _esp8266_print(gateway);
    _esp8266_print("\",\");
    _esp8266_print(mascara);
    _esp8266_print("\r\n");
    _esp8266_waitResponse();
}

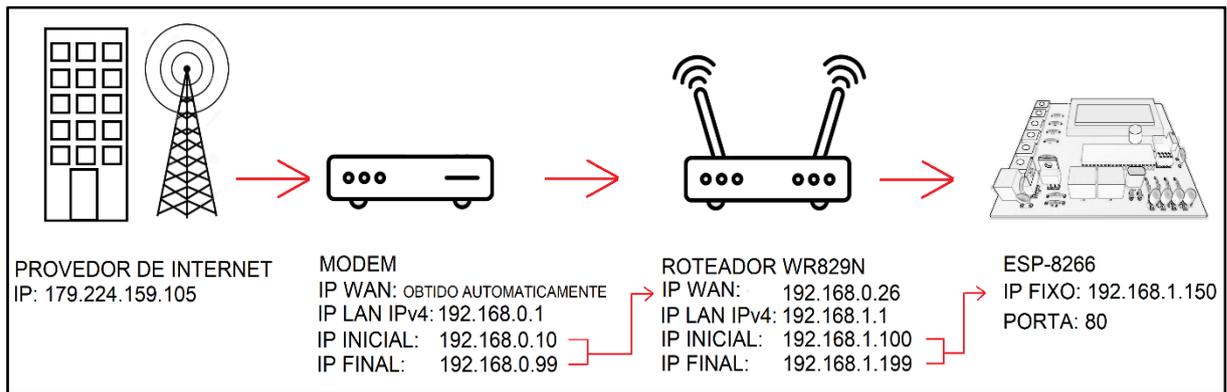
```

Fonte: O autor (2021)

Com isso, as configurações de rede necessárias para que o ESP seja comandado por um acesso externo foram finalizadas. Para auxiliar na compreensão da rede e de todos os endereços de IP envolvidos, elaborou-se um diagrama contendo os principais equipamentos presentes na rede. É possível verificar que o IP WAN do modem é dinâmico, ou seja, é obtido automaticamente pela provedora de internet. Cada vez que o modem é desligado e reiniciado após um determinado período de tempo a provedora de internet fornece um novo IP. Esse IP, está associado ao domínio cadastrado no site NO-IP (automacaok13), e cada vez que ele é alterado o aplicativo DUC atualiza esse novo IP ao domínio cadastrado.

O modem apresenta um IP LAN fixo 192.168.0.1 e distribui os endereços IP de cada aparelho conectado de acordo com o protocolo DHCP na faixa de 192.168.0.10 até 192.168.0.99. O roteador por sua vez, está conectado no endereço 192.168.0.26 do modem e possui um IP LAN fixo 192.168.1.1. Assim como o modem, o roteador distribui os endereços IP de cada aparelho conectado de acordo com o protocolo DHCP na faixa de 192.168.1.101 até 192.168.1.199. Por fim, o ESP está configurado para se conectar no IP 192.168.1.150 do roteador, possuindo assim um IP fixo. A Figura 40 apresenta um diagrama da rede desde o provedor de internet até o ESP-8266 presente na placa, com os respectivos IP's de cada aparelho.

Figura 40 – Diagrama da rede



Fonte: O autor (2021)

3.2.8 Lógica de programação para alteração dos modos

O projeto desenvolvido também foi pensado na possibilidade de o usuário encontrar problemas para utilizar o modo WEB, devido alguma falha em componentes como a queima do ESP ou em impossibilidades de se conectar com a rede criada, ou até mesmo efetuar um comando de forma manual devido estar ser nenhum aparelho com Wi-Fi disponível. Em vista disso, foi necessário possibilitar a utilização do modo por botões em qualquer momento em que o usuário solicitasse. Para isso foi utilizado a interrupção externa do microcontrolador PIC16F877A, a qual permite ao usuário pressionar um botão em qualquer momento e acessar o modo manual, independentemente do modo web estar funcionando ou não. Nesse caso de programação desenvolvida o botão selecionado foi configurado na porta RB5, visto que o PIC16F877a permite a seleção das portas RB4 a RB7 para interrupção externa.

Outra situação prevista na programação, foi o usuário retornar para o modo WEB após efetuar algum comando de forma manual na placa, devido estar impossibilitado naquele momento de utilizar uma conexão Wi-Fi. Para isso, foi criado uma opção do menu principal, a qual permitisse o usuário sair da interrupção externa e retornar ao controle via WEB. Ao selecionar a opção Modo WEB no menu principal, o usuário era direcionado a seguinte mensagem “deseja sair? Tecle seleção”. Ao pressionar seleção o programa saia da interrupção externa e voltava a programação do modo WEB, respondendo normalmente aos comandos da página WEB. Para isso, bastava atualizar a página WEB e o programa retornava a página que foi exibida por último antes da interrupção ser ativada.

4 ANÁLISE E DISCUSSÃO DOS RESULTADOS

Nesse capítulo são apresentados os resultados e discussões do desenvolvimento do projeto de controle e automação residencial utilizando o ESP8266, apresentando os testes práticos em *protoboard* e placa, a confecção da placa, a interface via web com o usuário e o levantamento dos custos envolvidos no projeto.

4.1 Fontes de alimentação do projeto

O primeiro teste realizado em *protoboard*, foi o projeto da fonte de tensão continua com saída fixa de 5V. Para isso, utilizou-se o regulador de tensão LM7805, com dois capacitores cerâmicos. Para efetuar o teste, foi utilizada uma fonte de alimentação comercial com tensão de entrada 220V, frequência de 60 Hz e uma potência de 12W. A saída da fonte comercial era de 6V, para uma corrente de 1000 mA. Para a medição da tensão, foi utilizado um multímetro Digital Ac/dc, da marca ICEL, modelo MD-1000A. A Figura 41 apresenta a medição realizada em *protoboard* em que é possível observar o valor de 5,02 V para a saída da fonte.

Figura 41 – Teste da fonte de tensão de 5V em *protoboard*

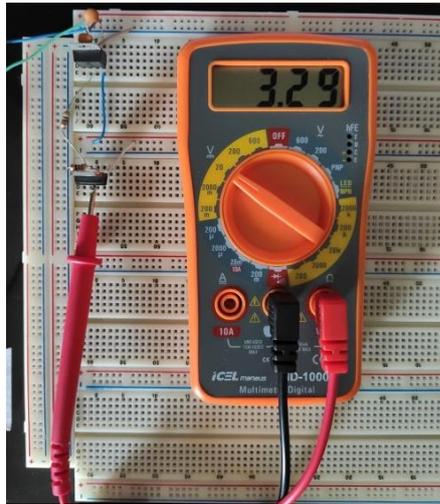


Fonte: O autor (2021)

Além da fonte de 5V, foi realizado o projeto da fonte de tensão continua com saída fixa de 3,3V em *protoboard*. Para isso, foi utilizado o regulador de tensão

LM317T, com um capacitor cerâmico na saída e dois resistores com valores de 220Ω $1/4W$ e 360Ω $1/4W$ para ajustar a tensão de saída conforme a literatura. Para efetuar o teste, foi utilizada a mesma fonte de alimentação especificada anteriormente e o mesmo multímetro. A Figura 42 apresenta a medição realizada em *protoboard* em que é possível observar o valor de 3,29 V para a saída da fonte.

Figura 42 – Teste da fonte de tensão de 3,3V em *protoboard*



Fonte: O autor (2021)

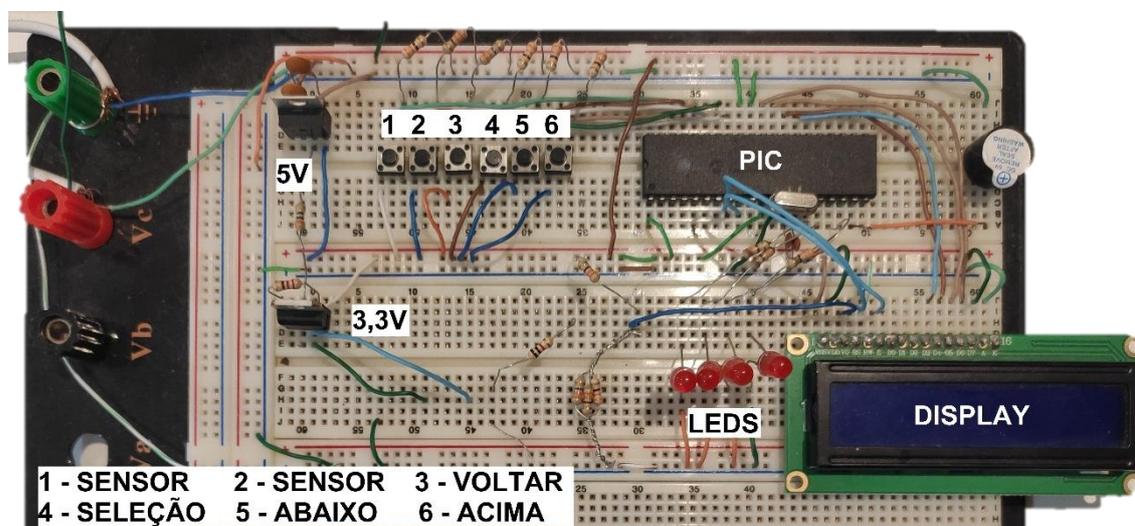
4.2 Etapa 01 – Acionamentos por botões

Após a construção das fontes, elaborou-se a montagem em *protoboard* do projeto de acionamento e controle das lâmpadas e alarme através de *push buttons*. Para isso, foi utilizada a fonte de alimentação de 5V elaborada anteriormente, seis *push buttons*, quatro led's, um *buzzer*, o *display* LCD, o PIC16F877a, *jumpers*, capacitores e resistores. Os *push buttons* apresentaram duas funções, sendo que dois deles foram utilizados para simular sensores de presença enquanto que os demais para simular as opções de interação com o usuário (cima, baixo, voltar e seleção). Os led's apresentaram a função de simular quatro lâmpadas de uma residência e o *buzzer* a função de alarme. O *display* foi utilizado para interagir com o usuário, mostrando os menus e informando o que estava acontecendo em cada etapa.

A Figura 43 apresenta a montagem em *protoboard* do projeto descrito, contendo os principais componentes com suas respectivas informações apresentadas na imagem. A montagem em *protoboard* também proporcionou uma ideia inicial das posições dos componentes no momento de elaborar a placa de circuito impresso, visto

que foi possível perceber os caminhos percorridos pelos *jumpers* após a saída do componente principal, o PIC, a fim de evitar cruzamentos.

Figura 43 – Montagem da primeira etapa em *protoboard*



Fonte: O autor (2021)

Com a montagem em *protoboard* da primeira etapa finalizada, utilizou-se a fonte de alimentação comercial para acionar todo o sistema, com o objetivo de verificar as simulações efetuadas no Proteus 8 e a lógica de programação. Ao alimentar o projeto, a primeira tela informada ao usuário através do *display* LCD foi o menu principal, contendo as opções de alarme, lâmpadas e configuração. Para navegar no menu e selecionar alguma opção foi utilizado os botões acima e abaixo. Devido ao *display* apresentar apenas duas linhas, o usuário necessitava selecionar duas vezes a tecla abaixo para o restante do menu aparecer, neste caso, apenas a opção configuração. A Figura 44 apresenta a tela de menu principal no *display* LCD para as opções de alarme e lâmpadas.

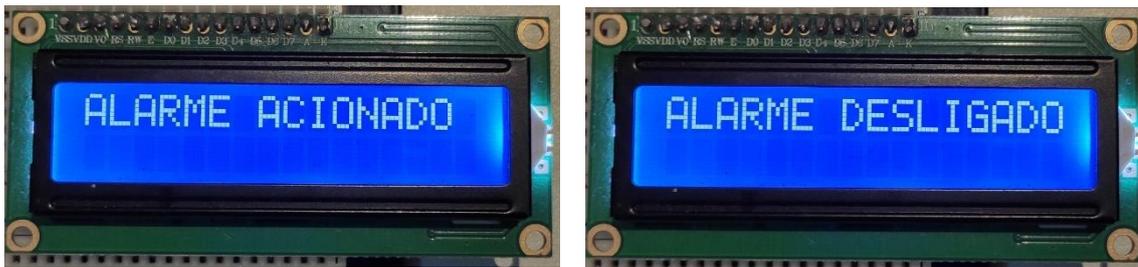
Figura 44 – Menu principal em testes no *protoboard*



Fonte: O autor (2021)

Ao selecionar a opção de alarme no menu principal, verificou-se que uma mensagem de alarme acionado era fornecida ao usuário, habilitando assim o sistema de alarme. Após um tempo de 2 segundos, o *display* era atualizado e novamente o menu principal era informado ao usuário. Ao selecionar novamente a opção alarme, uma nova mensagem de alarme desligado era fornecida ao usuário durante um tempo de 2 segundos, desabilitando o sistema de alarme. A Figura 45 (esq.) apresenta a mensagem de alarme acionado e a Figura 45 (dir.) apresenta a mensagem de alarme desligado, em testes realizados em *protoboard*.

Figura 45 – Mensagens de alarme em testes no *protoboard*

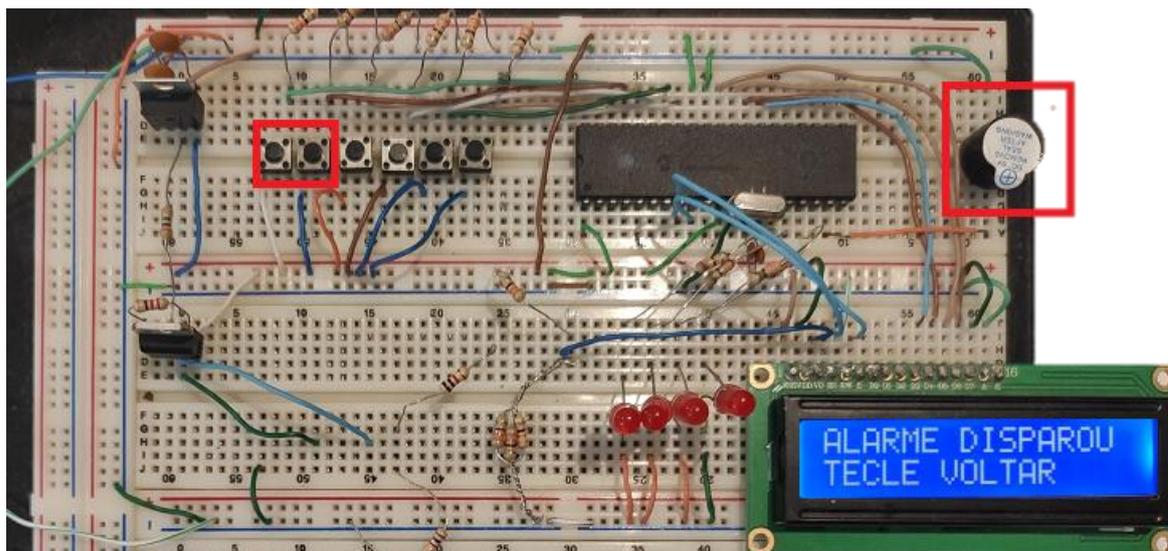


Fonte: O autor (2021)

Para efetuar o teste do *buzzer* e dos *push buttons* que simulavam os sensores, selecionou-se a opção alarme no menu principal, até que a mensagem de alarme acionado fosse informada. Ao clicar em um dos sensores, o PIC enviou nível lógico alto ao *buzzer* e o mesmo começou a emitir um som. Simultaneamente uma mensagem de “alarme disparou, tecler voltar” foi informada, solicitando ao usuário selecionar o botão voltar para desabilitar o alarme e por consequência desativar o *buzzer*.

A Figura 46 ilustra o momento em que o usuário seleciona um dos botões (retângulo vermelho a esquerda) que representam os sensores, e o sistema de alarme dispara, informando ao usuário a mensagem no *display* LCD. Nesse momento o *buzzer* está acionado (quadrado vermelho a direita) emitindo um barulho ao usuário. Ao selecionar a tecla voltar o alarme desabilita e o menu principal é exibido novamente ao usuário.

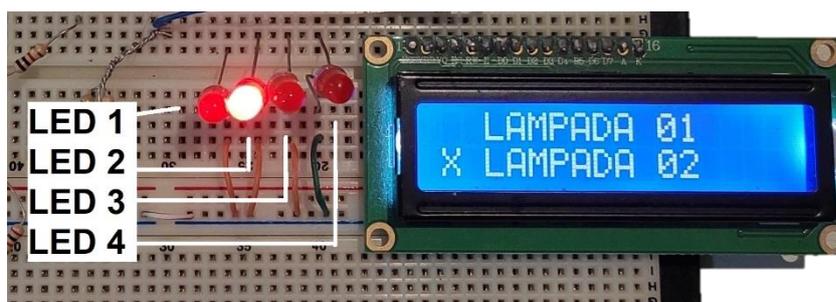
Figura 46 – Teste do sistema de alarme no *protoboard*



Fonte: O autor (2021)

Por fim, efetuou-se o teste do comando das lâmpadas. Para isso, selecionou-se a opção lâmpadas no menu principal, o que retornou ao usuário através do *display* um novo menu secundário, informando as lâmpadas presentes para o controle. Ao navegar com os botões (acima e abaixo) era possível selecionar a lâmpada que se desejava ligar e ao dar dois cliques na opção abaixo as lâmpadas três e quatro eram mostradas no *display*. Ao utilizar o botão seleção, a lâmpada era acionada ou desligada dependendo do nível lógico e ao selecionar o botão voltar, o menu principal era exibido novamente ao usuário. A Figura 47 apresenta o teste efetuado para controlar a lâmpada 02, em que ao selecionar a opção abaixo e depois a opção seleção a lâmpada 02 acendeu (representada pelo LED 2 informado na imagem).

Figura 47 – Teste do comando das lâmpadas no *protoboard*

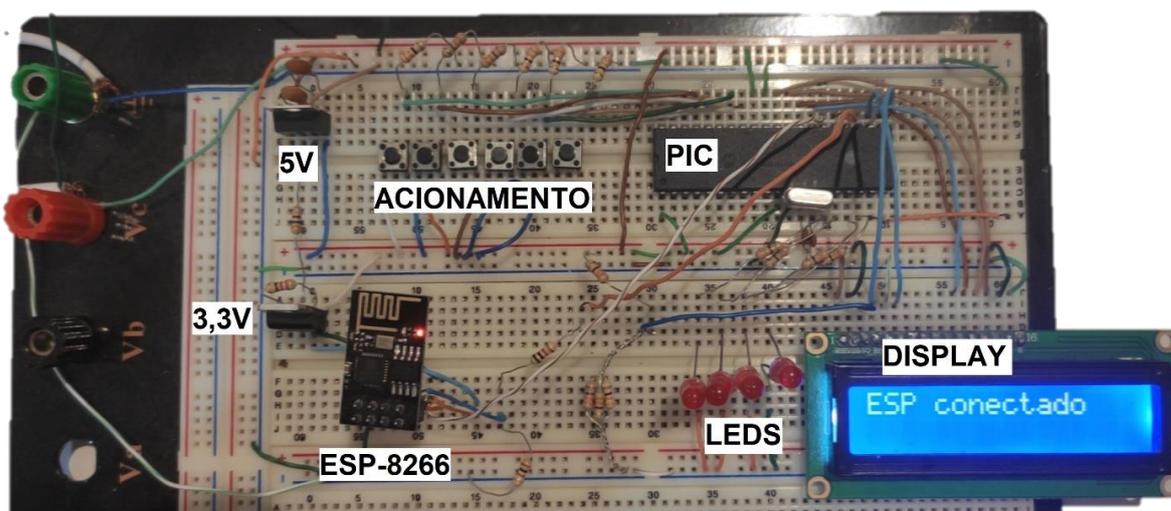


Fonte: O autor (2021)

4.3 Teste de comunicação entre PIC16F877A e o ESP-8266

Com os testes da etapa 01 finalizados, efetuou-se a montagem do restante do circuito para o funcionamento do ESP-8266. Para isso, foi adicionado ao projeto o ESP, dois resistores para que o retorno do sinal do PIC fosse de 3,3V, *jumpers* e a fonte de 3,3V para a alimentação do ESP. O *display* LCD foi utilizado novamente para a interface entre o programa e o usuário e os botões de acionamento e os LEDs não foram utilizados nessa etapa de teste. A Figura 48 mostra o projeto completo montado em *protoboard* para efetuar os testes de comunicação e criação de rede, com os principais componentes identificados na imagem.

Figura 48 – Montagem do projeto completo em *protoboard*

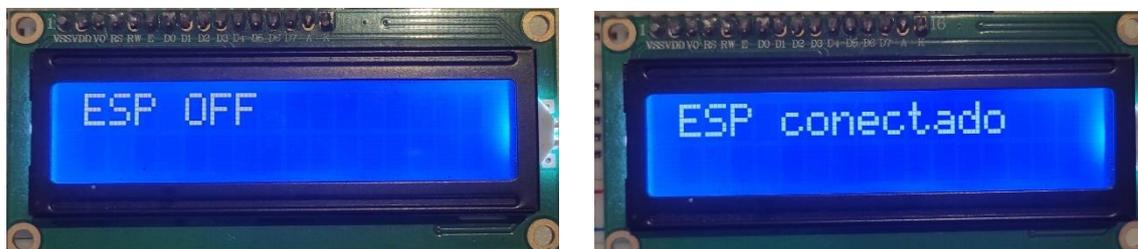


Fonte: O autor (2021)

Ao alimentar o projeto em *protoboard*, foi possível verificar se a programação para a comunicação entre o PIC e o ESP estava correta. Foi feito inicialmente, um teste removendo o ESP do *protoboard*. Como esperado, a mensagem de ESP OFF foi informada no *display* LCD informando ao usuário que o ESP estava desconectado e o programa ficou em um loop aguardando a conexão do ESP. Ao inserir o ESP no *protoboard*, foi possível verificar que o ESP ao receber o comando AT do microprocessador, retornou uma resposta "OK", informando que a conexão entre ESP e PIC estava correta. Assim, o programa foi liberado do loop e informou através do *display* LCD a mensagem de ESP conectado, informando ao usuário que o mesmo foi localizado pelo PIC. A Figura 49 (esq.) apresenta a mensagem de ESP OFF, informada

quando o ESP não é encontrado pelo PIC ou quando há algum problema de conexão e a Figura 49 (dir.) apresenta a mensagem quando a conexão entre o PIC e o ESP é estabelecida.

Figura 49 – Teste de conexão entre PIC e ESP em *protoboard*



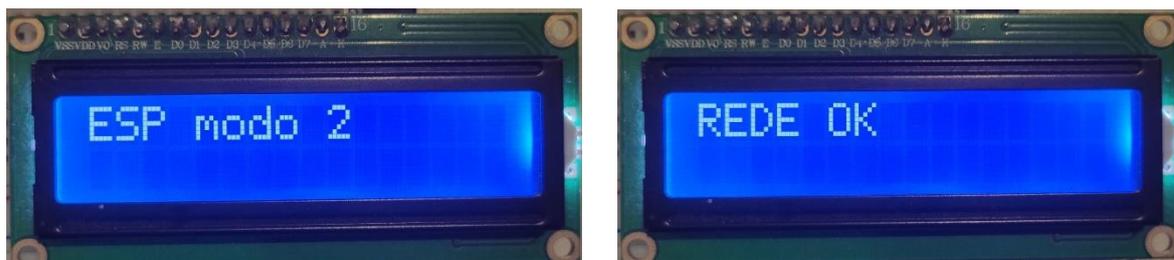
Fonte: O autor (2021)

4.4 Configuração da rede

Após efetuar a conexão entre o PIC e o ESP, foi verificado a configuração e a criação da rede através do ESP. Para isso, foi utilizado o comando “AT+CWMODE=2” no momento da programação, o que permitiu ao ESP se conectar em qualquer ponto de acesso com Wi-Fi (celular, computador, notebook), criando assim uma rede local. Ao executar essa etapa de programação, uma mensagem de ESP modo 2 foi informada ao usuário através do *display* LCD.

Após isso, a programação elaborada executou o comando “AT+CWSAP”, o que fez com que uma rede fosse criada. Para esse teste, utilizou-se o nome de “Rede Teste” e uma senha de oito caracteres foi inserida como chave de acesso. Com a rede criada, uma mensagem de REDE OK foi fornecida ao usuário através do *display* LCD informando que qualquer aparelho pudesse se conectar à rede criada.

A Figura 50 (esq.) apresenta a mensagem informando ao usuário o modo de operação que o ESP está operando e a Figura 50 (dir.) apresenta a mensagem informando ao usuário que a rede foi criada com sucesso. Nos testes em *protoboard*, foi observado que devido aos *jumpers* utilizados apresentarem problemas de conexão, algumas vezes não era possível criar a rede, porém esse problema foi sanado com a criação da placa de circuito impresso posteriormente.

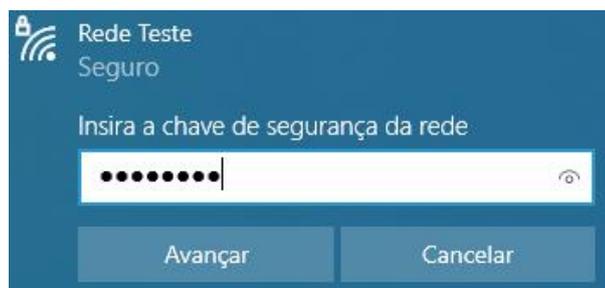
Figura 50 – Configuração da rede em *protoboard*

Fonte: O autor (2021)

4.5 Teste de conexão entre o PIC e outros dispositivos

Com a rede devidamente configurada, realizou-se os testes de conexão entre o projeto e os dispositivos que apresentassem conexão via Wi-Fi. Para isso utilizou-se inicialmente um computador desktop. Ao procurar pela opção “rede e internet” no computador, foi encontrado a rede criada pelo ESP, denominada de Rede Teste. Após clicar na Rede Teste, foi solicitado que a senha cadastrada no programa fosse inserida. A Figura 51 apresenta o momento em que a rede é encontrada pelo computador e uma senha é solicitada.

Figura 51 – Teste de rede através de um computador

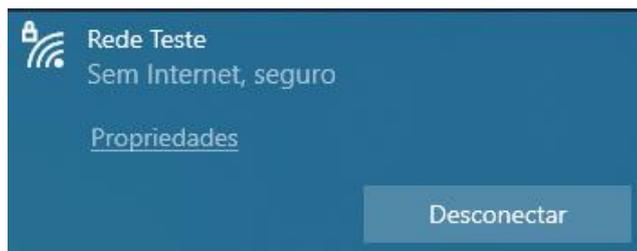


Fonte: O autor (2021)

Ao inserir a senha e selecionar a opção avançar, o computador reconheceu a rede e conectou-se a rede criada. É importante ressaltar que nenhum problema ou configuração extra foi solicitado no momento em que os testes foram efetuados. A Figura 52 apresenta a mensagem exibida pelo computador após a conexão, em que é possível notar uma conexão segura, porém ser internet.

Além do teste realizado com o computador, dois celulares de marcas Xiaomi e Samsung foram utilizados para efetuar o teste de conexão. A conexão entre eles e o ESP ocorreram da forma esperada, sem dificuldades ou falha de reconhecimento.

Figura 52 – Status de conexão entre o ESP e o computador

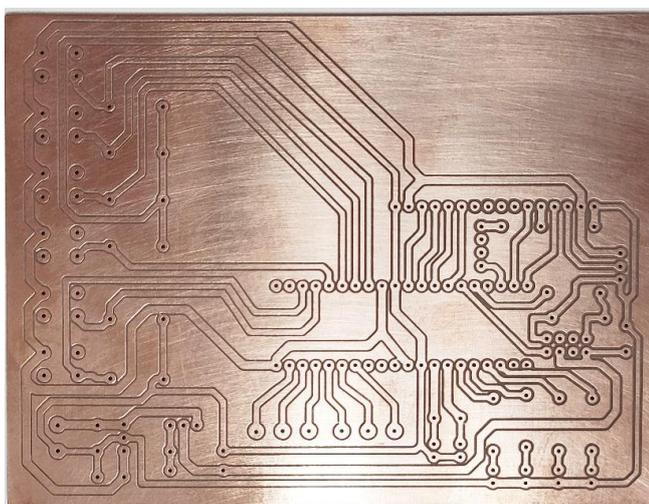


Fonte: O autor (2021)

4.6 Placa de circuito impresso

Com os testes de conexão devidamente realizados, realizou-se a confecção da placa para dar procedimento ao projeto e controlar as lâmpadas e o alarme remotamente. Para a confecção da placa de circuito impresso, foi utilizada uma placa de fenolite com dimensões de 15cm de comprimento por 15cm de largura, com face única. A usinagem foi realizada em uma prototipadora de circuito impresso CNC. Foi utilizado uma fresa de gravação cônica 60° x 0,1mm metal duro para a isolação das trilhas e uma fresa raidada 3mm para o corte externo da placa. As dimensões finais da placa foram de 9,4cm x 12,5cm. Os furos para inserção dos componentes foram realizados com brocas de 0,8mm, 1,0mm e 1,5mm de diâmetro, dependendo da necessidade de cada componente. A Figura 53 demonstra a placa de circuito impresso após o processo de usinagem das trilhas.

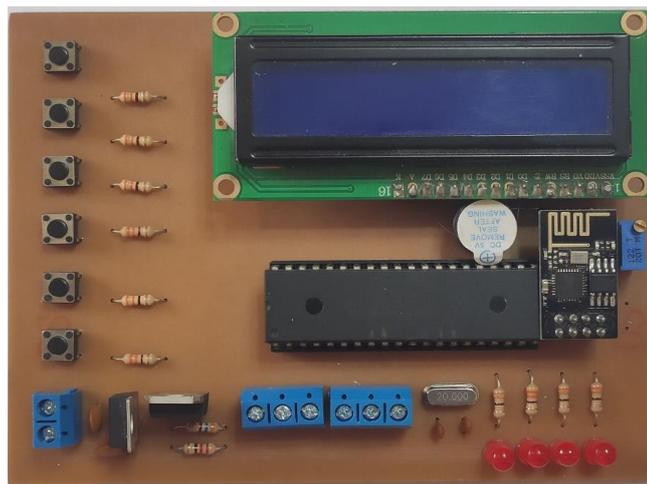
Figura 53 – Placa de circuito impresso após usinagem das trilhas



Fonte: O autor (2021)

Com a placa usinada e perfurada, foi efetuado a solda dos componentes eletrônicos do projeto e dos soquetes para a inserção do *display*, do microcontrolador e do ESP-8266. Para isso, foi utilizado fio de solda 63Sn/37pb com resina de 1mm de diâmetro e um ferro de solda 60W. Também foram inseridos bornes para a alimentação e para as portas de expansão do PIC, facilitando a montagem e a desmontagem. A Figura 54 apresenta a placa com todos os componentes utilizados no projeto, juntamente com o *display*, o microcontrolador e o módulo ESP.

Figura 54 – Placa de circuito impresso com componentes soldados



Fonte: O autor (2021)

4.7 Etapa 02 – Acionamentos via página WEB – Rede Local

Os testes efetuados para os acionamentos via página WEB, para uma rede local, foram realizados na placa de circuito impresso desenvolvida. Ao efetuar a ligação da placa, a etapa de conexão entre o ESP e o PIC, conforme apresentado na seção 4.5 foi iniciada, informando que o ESP estava conectado em modo 2 e a rede estava disponível para qualquer dispositivo com Wi-Fi se conectar. Posteriormente a essa etapa, foi apresentado ao usuário o endereço de IP no *display* LCD para que o mesmo pudesse acessar pela primeira vez o menu principal via página WEB. O endereço IP apresentado para os testes foi o de número 192.168.1.4. Esse endereço foi fornecido ao usuário até o momento em que algum dispositivo acessou o endereço através de um navegador. Também foi utilizado um LED externo a placa na porta RA0 para sinalizar em que etapa ou menu o programa se encontrava no momento do acionamento via página WEB. A Figura 55 mostra a placa desenvolvida em

funcionamento com o endereço IP apresentado no *display* LCD aguardando o acesso do endereço fornecido através de um navegador do dispositivo conectado à rede.

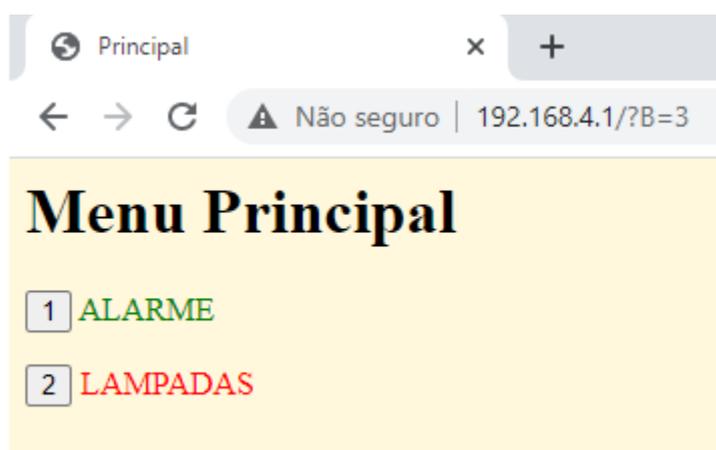
Figura 55 – Endereço IP para conexão via WEB



Fonte: O autor (2021)

Para efetuar os testes, foi utilizado o navegador Google Chrome Versão 88.0.4324.146 de 64bits. Ao digitar o endereço IP 192.168.4.1 no navegador, a página de Menu Principal foi carregada e fornecida ao usuário. Nessa página, as opções de Alarme e Lâmpadas foram apresentadas ao usuário. Outras opções como controle de temperatura e configuração de senha podem vir a ser adicionadas a essa página futuramente para expansão do projeto. A Figura 56 apresenta a página de Menu principal fornecida ao usuário no navegador WEB.

Figura 56 – Página WEB – Menu Principal



Fonte: O autor (2021)

Simultaneamente ao acesso do endereço IP via navegador, o LED auxiliar piscou 5 vezes durante 2,5 segundos na PCI, informando que um acesso via WEB foi feito e que o mesmo se encontrava na página de Menu Principal. Uma mensagem de Modo WEB também foi exibida no *display* LCD, com o objetivo de informar ao usuário que um acesso via página WEB estava sendo executado.

Figura 57 – Aviso ao usuário que o modo WEB está sendo executado



Fonte: O autor (2021)

Ao selecionar a opção Alarme no Menu Principal via página WEB, o usuário era direcionado para outra página contendo as opções de ativar o alarme, desativar o alarme ou voltar ao menu principal novamente. O endereço da página continuava sendo o mesmo, porém a página carregada ao usuário era atualizada na programação e no navegador. Ao entrar no menu alarme, novamente o LED auxiliar informava ao usuário que o modo WEB estava sendo utilizado e piscava por 3 vezes durante 1,5 segundos. Assim era possível saber que o menu do alarme estava sendo acionado via página WEB.

Ao selecionar a opção Ativar Alarme, a mensagem de “Monitorando Alarme ON” era fornecida em tempo real ao usuário via página WEB, do mesmo modo que se a opção desativar alarme era selecionada, uma mensagem de “Alarme Desativado OFF” era fornecida ao usuário. Para ativar ou desativar o alarme bastava ao usuário selecionar a opção 1 ou 2 respectivamente na página, sendo que a condição inicial do programa era com a opção de alarme desativado. A Figura 58 (esq.) apresenta o menu alarme com o alarme desativado (condição inicial ao utilizar o programa) e a Figura 58 (dir.) mostra o menu alarme com o alarme ativado, devido ao usuário

selecionar a opção 1 (ativar alarme) na página WEB.

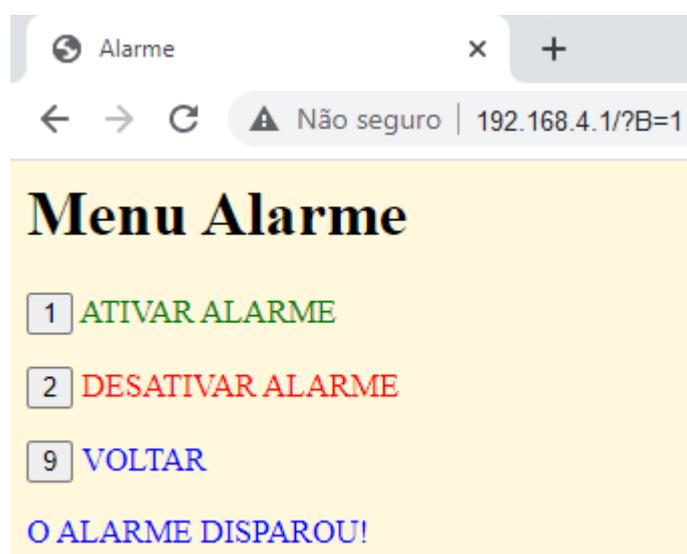
Figura 58 – Página WEB – Menu Alarme



Fonte: O autor (2021)

Com a opção 1 do menu alarme selecionada, o monitoramento do alarme foi iniciado. Independente do usuário estar na página de alarme ou não, o monitoramento funcionou corretamente, visto que foi utilizado a opção de interrupção externa durante a programação dando prioridade ao controle de alarme. Assim, no momento em que o alarme disparou, o usuário foi redirecionado para o menu alarme com a mensagem de “ Alarme Disparou! ”, conforme apresenta a Figura 59.

Figura 59 – Página WEB – Alarme disparou



Fonte: O autor (2021)

Simultaneamente a página WEB apresentada Figura 59, a informação do disparo do alarme também foi apresentada na placa desenvolvida. Para isso foi utilizado o *display* LCD informando a mesma mensagem presente no modo por botões, o acionamento do *Buzzer* e o LED 4 (acionado apenas para exemplificar na apresentação desta etapa no trabalho). Para desativar o alarme, foi possível utilizar duas opções, sendo a primeira delas de selecionar o botão voltar na placa, assim como apresentado no acionamento por botões ou a segunda opção, selecionando a opção desativar alarme conforme apresentado na página WEB na Figura 58 (esq.). Ao selecionar uma das duas opções, o alarme foi desativado e as mensagens via página WEB e *display* LCD foram atualizadas. Na Figura 60 é possível verificar o momento em que o alarme é acionado e a mensagem é apresentada ao usuário, juntamente com o sinal visual e sonoro.

Figura 60 – Mensagem de “alarme Disparou”, sinalização sonora e visual



Fonte: O autor (2021)

Por fim, ao selecionar a opção Lâmpadas no menu principal, o usuário era direcionado a um novo menu, denominado de menu lâmpadas. Assim como no caso anterior, o endereço da página continuava sendo o mesmo, porém era atualizada ao usuário devido a opção escolhida.

Ao entrar no menu lâmpadas, o usuário encontrava as opções de ligar ou desligar as quatro lâmpadas cadastradas (LED's), além da opção voltar, a qual direcionava o usuário novamente ao menu principal. Para acionar a lâmpada 2, por exemplo, bastava selecionar a opção “LED2 ON” e para desligá-la bastava selecionar

a opção “LED2 OFF”. Cada vez que um botão era acionado a página era atualizada e recarregada ao usuário. Assim como no menu principal e no menu alarme, o LED auxiliar informava ao usuário que o modo WEB estava sendo utilizado na opção lâmpadas e piscava por 4 vezes durante 2,0 segundos. A Figura 61 apresenta a página WEB para o menu lâmpadas, contendo as nove opções descritas anteriormente.

Figura 61 – Menu Lâmpadas



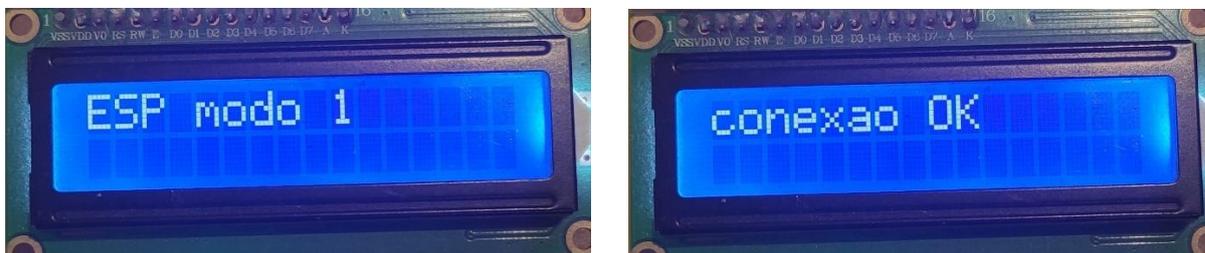
Fonte: O autor (2021)

4.8 Etapa 02 – Acionamentos via página WEB – Acesso Externo

Os testes efetuados para os acionamentos via página WEB, para o acesso externo, também foram realizados na placa de circuito impresso desenvolvida. Para que o acesso externo fosse possível, foram efetuadas as alterações presentes na seção 3.2.7 desse trabalho e a função presente na Figura 39, foi adicionada na programação, a fim de informar ao ESP, seu endereço IP fixo, o *Gateway* padrão e a máscara de sub-rede. Ao efetuar a ligação da placa, a mensagem de ESP modo 1 era fornecida no *display* LCD, conforme apresentado na Figura 62 (esq.), informando que

o ESP estava conectado em modo estação (*Station*). Após o ESP se conectar na rede do roteador configurado, uma mensagem de Conexão OK era fornecida ao usuário, conforme mostra a Figura 62 (dir.).

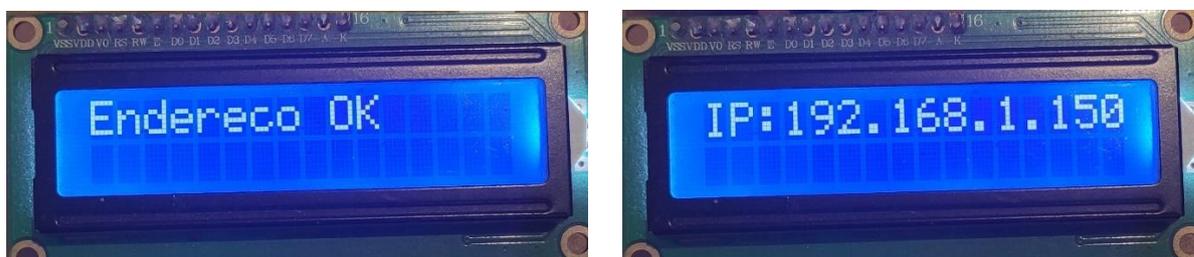
Figura 62 – Configuração da rede na placa



Fonte: O autor (2021)

Em seguida, ao configurar os parâmetros de rede, como seu endereço IP, Gateway e máscara, uma mensagem de endereço OK era apresentada no *display*, conforme mostra Figura 63 (esq.). Por fim, uma mensagem era apresentada ao usuário contendo o endereço de IP (192.168.1.150) no *display* LCD, conforme Figura 63 (dir.). Esse endereço informado no *display* era o mesmo cadastrado no roteador e apresentava por objetivo realizar um acesso local, caso o usuário apresentasse problemas com o acesso externo através do domínio cadastrado no site NO-IP (automacaok13).

Figura 63 – Configuração de endereço IP na placa

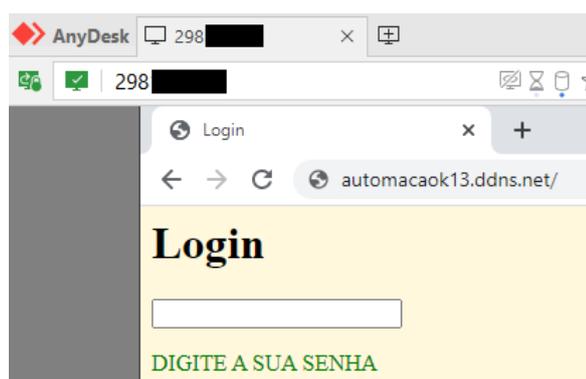


Fonte: O autor (2021)

Para efetuar os testes via página WEB através do acesso externo, foi utilizado o aplicativo de desktop remoto AnyDesk v6.3.2, distribuído pela AnyDesk software GmbH. Com ele, foi possível acessar o navegador WEB de um computador presente em outra localidade, conectado em outra rede, e controlar o que estava acontecendo em um computador na mesma localidade da placa, controlando assim o ESP a

distância, porém, acompanhando o seu funcionamento no local. Devido a utilização do acesso externo, foi necessário elaborar uma tela inicial de *login* para garantir a segurança do projeto, solicitando ao usuário uma senha de acesso. Na Figura 64 é possível verificar a tela inicial de *login* fornecida ao usuário ao acessar o domínio “automacaok13.ddns.net/”. Também é possível verificar que o aplicativo AnyDesk está sendo utilizado para efetuar o *login* a distância de um computador presente em outra rede.

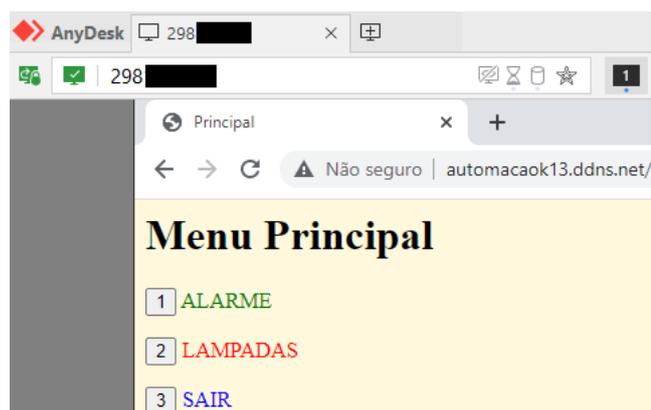
Figura 64 – Menu de *Login*



Fonte: O autor (2021)

Ao efetuar o *Login*, o usuário era direcionado para a página de Menu Principal. Nessa página, além das opções de Alarme e Lâmpadas fornecidas em um acesso local, a opção de sair também era fornecida ao usuário. Ao selecionar essa opção o usuário era direcionado a tela de *Login* inicial, solicitando novamente a senha de acesso. A Figura 65 apresenta o menu principal para o acesso externo.

Figura 65 – Menu principal



Fonte: O autor (2021)

4.9 Modo WEB para modo Manual

O teste de alteração do modo WEB para o modo de operação por botões foi realizado de dois modos. Inicialmente, o teste foi realizado com o ESP-8266 presente na placa, com o objetivo de simular que um usuário presente na casa e sem o celular ou computador no momento pudesse controlar o sistema de alarme e iluminação manualmente. Foi efetuado a alteração do modo através do botão RB5, o qual ativou a interrupção externa do microcontrolador e habilitou a etapa do programa que funciona de modo manual.

Em seguida o teste foi repetido, porém sem a presença do ESP na placa, o qual simulava um eventual problema no sistema que o modo WEB estivesse indisponível para uma situação de emergência, como desativar o alarme. Esse teste também funcionou corretamente e permitiu a entrada no modo manual. Independentemente do local em que o programa se encontrava, foi possível realizar a alteração do modo WEB para o modo manual, devido a interrupção externa utilizada durante a programação. A Figura 66 apresenta o teste com o ESP presente na placa, para o modo de operação manual com os botões, o qual foi permitido após pressionar a chave que está presente no retângulo vermelho da imagem.

Figura 66 – Botão de alteração do modo WEB para modo manual



Fonte: O autor (2021)

4.10 Modo Manual para modo WEB

Para realizar esse teste, inicialmente foi efetuado algumas operações no modo WEB, como ligar e desligar lâmpadas e ativar o alarme. Em seguida o modo manual foi ativado conforme apresentado na seção anterior e as mesmas operações foram repetidas, a fim de verificar se o programa não apresentaria erros ou ficaria preso em loops. Por fim, a opção de “modo WEB” no menu principal foi selecionada. Ao selecioná-la, a mensagem de “deseja sair? Tecle seleção” foi fornecida no *display* LCD ao usuário. Ao clicar novamente em seleção o programa encerrou o loop na interrupção externa e retornou ao funcionamento no modo WEB. Mais alguns testes foram realizados no modo WEB após retornar do modo manual, a fim de verificar se o programa não apresentaria problemas de programação. A Figura 67 (esq.) apresenta o programa operando em modo manual no menu principal com a opção de “modo WEB” para selecionar e a Figura 67 (dir.) apresenta a mensagem solicitando a tecla seleção para retornar ao modo WEB. Após pressionar a tecla seleção o programa retornou ao modo WEB e a mensagem de modo WEB apresentada na Figura 57 retornou ao *display*. Para manusear o programa no modo WEB novamente, foi necessário atualizar a página do navegador.

Figura 67 – Menu principal – Opção de retorno para modo WEB



Fonte: O autor (2021)

4.11 Lista de materiais e orçamento

Uma lista de materiais e componentes necessários para elaborar o projeto foi criada, com o objetivo de possibilitar a replicação do projeto futuramente, além de fornecer uma base para efetuar o orçamento do projeto. Em vista disso, a Tabela 4 apresenta os componentes e materiais utilizados para a confecção do projeto, com suas quantidades, especificações, valor unitário e valor total. Os valores foram obtidos de um fornecedor de componentes eletrônicos da cidade de Joinville-SC, o que pode alterar o valor de projeto dependendo da localidade. O Valor do roteador WR829N foi de R\$ 79,99, porém não foi incluído no orçamento, pois na grande maioria dos casos é possível utilizar o roteador presente nas residências fornecido pela provedora de internet, não sendo necessário a aquisição do mesmo.

Tabela 4 – Lista de materiais e orçamento

Quantidade	Componente	RS Unitário	R\$ Total
01	Placa fenolite cobreada 1 face 15x15 cm	8,50	8,50
01	Soquete C.I. 40 pinos	1,50	1,50
01	Microcontrolador PIC16F877A Microchip	27,00	27,00
01	Barra soquete fêmea 1x40 pinos MCI 180°	2,00	2,00
01	Barra pinos macho 1x40 180°	1,50	1,50
01	Display LCD 16x2 Backlight Azul	21,90	21,90
01	Módulo Wi-Fi ESP-8266 ESP-01	25,00	25,00
01	Buzzer DC 5V	3,00	3,00
01	Terminal Borne KF301 2 vias Azul	1,00	1,00
02	Terminal Borne KF301 3 vias Azul	1,50	3,00
01	Trimpot 3296 1kΩ vertical multivoltas	3,00	3,00
01	Regulador de tensão L7805	1,90	1,90
01	Regulador de tensão ajustável LM317T	2,50	2,50
06	Chave táctil 6x6x5mm 4 terminais	0,50	3,00
01	Cristal oscilador 20 MHz caneca baixa	1,50	1,50
04	LED difuso 5mm vermelho	0,40	1,60
02	Capacitor cerâmico 22pF 50V	0,20	0,40
01	Capacitor cerâmico 10nF 50V	0,10	0,10
02	Capacitor cerâmico 100nF (0,1uF) 4	0,15	0,30
01	Resistor 1kΩ 5% (1/4W)	0,15	0,15
01	Resistor 2kΩ 5% (1/4W)	0,15	0,15
10	Resistor 330Ω 5% (1/4W)	0,15	1,50
01	Resistor 220Ω 5% (1/4W)	0,15	0,15
01	Resistor 360Ω 5% (1/4W)	0,15	0,15
Total			110,80

Fonte: O autor, 2021.

5 CONCLUSÃO

A automação residencial é um campo da engenharia elétrica que apresenta um elevado potencial de investimento e crescimento, devido proporcionar grandes vantagens relacionadas ao conforto e segurança dos usuários quando comparada com sistemas manuais. No entanto, devido aos elevados custos dos sistemas atuais encontrados no mercado, a busca pela implementação nas residências brasileiras ainda é muita baixa.

Em vista disso, foi realizado um estudo sobre os principais conceitos relacionados a automação residencial e elaborado um embasamento teórico contendo os tópicos necessários para elaboração desse projeto. A partir do estudo realizado, foram levantados os objetivos necessários para elaborar o projeto. Assim, foi realizado o desenvolvimento de um sistema de automação residencial capaz de efetuar o monitoramento e o controle do sistema de iluminação e alarme de uma residência utilizando o PIC16F877A e o módulo ESP-8266 através de uma interface web e acesso externo.

Para isso, foi elaborado a lógica de programação do sistema de iluminação e alarme através da linguagem C no ambiente de desenvolvimento integrado MPLAB X IDE e testado no simulador Proteus. Foi verificado a necessidade de utilizar interrupções no programa, a fim de garantir o pleno funcionamento do sistema de alarme, independente da etapa de programa que estava sendo executada. Também foi realizado a montagem do projeto em *protoboard*, em que foi possível validar cada etapa do projeto de modo individual, corrigindo os erros encontrados, evitando assim, problemas no projeto final na placa de circuito impresso.

Após a elaboração da placa, foi constatado que todas as funcionalidades propostas apresentaram um resultado positivo. Também foi verificado que em caso do não funcionamento do módulo ESP, o usuário poderá efetuar o controle de modo manual, apertando apenas um botão na placa desenvolvida. Isso garantiu que o usuário não ficasse refém apenas do sistema automatizado, podendo efetuar os controles da iluminação e alarme do modo tradicional.

Foi constatado que o sistema de automação residencial proposto apresentou um custo de R\$ 110,80 para compra dos componentes, sendo necessário apenas acrescentar um valor para instalação do mesmo em um ambiente residencial. Esse valor é relativamente baixo em comparação com os atuais valores encontrados no

mercado, e viabiliza a produção e instalação do mesmo em uma residência. Também foi verificado que a memória de programa do PIC16F877A utilizada no projeto ficou em 57%, o que permite a adição de novas funcionalidades e conseqüentemente a expansão do projeto, porém a memória de dados resultou em uma ocupação de 98%, principalmente devido a quantia de textos criados para ser mostrado no *display* LCD, o que pode ser um possível problema, necessitando assim utilizar um microcontrolador mais robusto, como o PIC18F4550.

Em vista disso, pode-se verificar que o projeto elaborado atingiu seu objetivo principal e todos os objetivos específicos, resultando em um sistema de controle e automação de baixo custo para uma residência. Também pode-se verificar a flexibilidade do projeto desenvolvido, permitindo a expansão do sistema de alarme e iluminação ou a adição de outras funcionalidades como o controle de temperatura ou abertura e fechamento de um portão eletrônico através dos bornes deixados na placa.

6 SUGESTÕES PARA TRABALHOS FUTUROS

Apresentam-se como sugestões para trabalhos futuros, baseado no projeto desenvolvido:

Desenvolvimento de um aplicativo android para controle e automação residencial utilizando o PIC16F877A.

Desenvolvimento do projeto de controle e automação residencial utilizando o PIC16F877A e um módulo Bluetooth.

Desenvolvimento de um sistema de monitoramento e controle de temperatura e umidade ambiente utilizando o PIC16F877A e um controlador PID.

Implementação de um sistema de alarme com a adição de canais de áudio e vídeo com base no sistema desenvolvido.

Implementação do projeto de controle e automação residencial utilizando o PIC16F877A via página WEB em um ambiente residencial.

REFERÊNCIAS

- AI-THINKER. **ESP-01 WiFi Module**. Shenzhen: Ai-Thinker Team, 2015. 19 p. Disponível em: <http://www.microchip.ua/wireless/esp01.pdf>. Acesso em: 10 jan. 2021.
- BATRINU, Catalin. **ESP8266 home automation projects**. Birmingham, Mumbai: Packt Publishing Ltd., 2017. 196 p.
- BUSINESS INSIDER (Estados Unidos). **The Internet of Things Report**. New York City, 2019.
- CARISSIMI, Alexandre da Silva; ROCHOL, Juergen; GRANVILLE, Lisandro Zambenedetti. **Redes de computadores**. Porto Alegre: Editora Bookman, 2009. 391 p.
- COELHO, Darlene Figueiredo Borges; CRUZ, Victor Hugo do Nascimento. **Edifícios inteligentes: uma visão das tecnologias aplicadas**. São Paulo: Editora Blucher, 2017. 136 p.
- COMER, Douglas E.. **Redes de Computadores e Internet**. 6. ed. Porto Alegre: Editora Bookman, 2016. 557 p. Tradução de: José Valdeni de Lima e Valter Roesler.
- DAMAS, Luís. **Linguagem C**. 10. ed. Rio de Janeiro: Editora Ltc, 2016. 374 p. Tradução de: João Araújo Ribeiro. Disponível em: <https://integrada.minhabiblioteca.com.br/#/books/9788521632474/epubcfi/6/8%5B%3Bvnd.vst.idref%3Dtitle%5D!/4/2/4%5Bvst-image-button-989004%5D%400:41.8>. Acesso em: 01 nov. 2020.
- DIPOL. **O que é DDNS (Dynamic Domain Name System) e como usá-lo?** 2013. Disponível em: https://www.dipol.pt/o_que_e_ddns_dynamic_domain_name_system_e_como_usa-lo__bib93.htm. Acesso em: 05 jun. 2021.
- DYN. **Dynamic DNS**. 2021. Disponível em: <https://pages.dyn.com/dyndns-remote-access/>. Acesso em: 15 jun. 2021.
- ESPRESSIF (China). **ESP8266 AT Instruction Set**. [S. l.]: Espressif Inc, 2018. 70 p. Disponível em: <https://www.espressif.com/en/support/documents/technical-documents>. Acesso em: 10 jul. 2020.
- ESPRESSIF (China). **ESP8266EX: datasheet**. [S. l.]: Espressif Inc, 2020. 31 p. (Versão 6.6). Disponível em: <https://www.espressif.com/en/support/documents/technical-documents>. Acesso em: 22 nov. 2020.
- FERREIRA, Ronaldo Rodrigues. **Linguagem de programação**. Curitiba: Editora Contentus, 2020. 56 p. Disponível em: <https://plataforma.bvirtual.com.br/Acervo/Publicacao/182483>. Acesso em: 04 set. 2020.

FLATSCHART, Fábio. **HTML5: embarque imediato**. Rio de Janeiro: Editora Brasport, 2011. 256 p. Disponível em: <https://plataforma.bvirtual.com.br/Acervo/Publicacao/160673>. Acesso em: 10 out. 2020.

FOROUZAN, Behrouz A.. **Data Communications and Networking**. 4. ed. New York City: Editora Mc Graw Hill, 2006. 1168 p.

KUROSE, James F; ROSS, Keith W. **Redes de computadores e a internet: uma abordagem top-down**. 6. ed. São Paulo: Editora Pearson Education do Brasil, 2013. 634 p. Tradução de: Daniel Vieira.

LAMB, Frank. **Industrial automation: Hands-on**. New York City: Mc Graw Hill Education, 2013.

LEMAY, Laura. **Sams Teach Yourself Web Publishing with HTML & XHTML in 21 Days**. 4. ed. Indianapolis: Editora Sams, 2003. 816 p.

MAIA, Luiz Paulo. **Arquitetura de redes de computadores**. 2. ed. Rio de Janeiro: Editora Ltc, 2013.

MARQUES, Mayara Carla. **Pesquisa sobre o interesse em automação residencial nas cidades de pequeno porte, com base na percepção dos potenciais usuários**. 2018. 18 f. TCC (Graduação) - Curso de Tecnólogo em Gestão Comercial, Instituto Federal da Paraíba (IFPB), Guarabira, 2018. Disponível em: <https://repositorio.ifpb.edu.br/xmlui/bitstream/handle/177683/695/Pesquisa%20sobre%20o%20interesse%20em%20automa%c3%a7%c3%a3o%20residencial%20nas%20cidades%20de%20pequeno%20porte%2c%20com%20base%20na%20percep%c3%a7%c3%a3o%20dos%20potencias%20usu%c3%a1rios%20-%20Mayara%20Carla%20Marques.pdf?sequence=1&isAllowed=y>. Acesso em: 05 mar. 2020.

MARTINS, Nardênio Almeida. **Sistemas Microcontrolados: Uma abordagem com o microcontrolador PIC16F84**. São Paulo: Novatec Editora LTDA, 2005. 263 p.

MELO, Ana Cristina Vieira de; SILVA, Flávio Soares Corrêa da. **Princípios de linguagens de programação**. São Paulo: Editora Blucher, 2003. 211 p.

MICROCHIP. **PIC16F877A: newer device available pic16f18877**. Newer Device Available PIC16F18877. 2020. Microchip Technology Inc. Disponível em: <https://www.microchip.com/wwwproducts/en/PIC16F877A>. Acesso em: 02 set. 2020.

MORAES, Alexandre Fernandes de. **Redes de Computadores: fundamentos**. 8. ed. São Paulo: Editora Érica, 2020. 248 p.

NO-IP. **Acesso Remoto com DNS Dinâmico**. 2021. Disponível em: <https://www.noip.com/pt-BR/remote-access>. Acesso em: 15 jun. 2021.

PAULUS, Gustavo Bathu et al. **Sistema de automação residencial: acessibilidade no controle**. XXII jornada de pesquisa, 2017, Ijuí: Salão do Conhecimento, 2017. 13 p. Disponível em:

<https://publicacoeseventos.unijui.edu.br/index.php/salaconhecimento/article/view/7676/6413>. Acesso em: 07 mar. 2020.

PEREIRA, Fábio. **Microcontroladores PIC**: programação em c. São Paulo: Editora Érica, 2003. 355 p.

PRUDENTE, Francesco. **Automação Predial e Residencial**: Uma Introdução. Rio de Janeiro: Editora LTC, 2017. 211 p.

SANTOS, Marcela Gonçalves dos; SARAIVA, Maurício de Oliveira; GONÇALVES, Priscila de Fátima. **Linguagem de Programação**. Porto Alegre: Sagah, 2018.

SEBRAE (org.). **Construção civil: boletim de tendência - automação residencial**. Boletim de Tendência - Automação residencial. Rio de Janeiro: Sebrae Inteligência Setorial, 2015. 5 p. Disponível em: <https://sebraeinteligenciasetorial.com.br/produtos/boletins-de-tendencia/automacao-residencial/55dc60f3c3779d21009a1f2a/arquivo/450cea51-6f88-4324-b36c-d90507545d82.pdf>. Acesso em: 02 mar. 2020.

SCHMITT, Marcelo Augusto Rauh; PERES, André; LOUREIRO, César Augusto Hass. **Redes de computadores**: nível de aplicação e instalação de serviços. Porto Alegre: Editora Bookman, 2013. 173 p.

SOUSA, Lindeberg Barros de. **Redes de computadores**: guia total. São Paulo: Editora Érica, 2009. 334 p.

SOUZA, David José de; LAVINIA, Nicolás César. **Conectando o PIC 16F877A**: Recursos Avançados. 4. ed. São Paulo: Editora Érica, 2007.

STEVAN JR, Sergio Luiz; FARINELLI, Felipe Adalberto. **Domótica**: Automação Residencial e Casas Inteligentes com Arduíno e ESP8266. São Paulo: Editora Érica, 2019.

TERUEL, Evandro Carlos. **HTML5**: guia prático. 2. ed. São Paulo: Editora Érica, 2014. 336 p.

TP-LINK Corporation Limited. **TL-WR829N**: roteador wireless multimodo 300 mbps. Roteador Wireless Multimodo 300 Mbps. 2021. Disponível em: <https://www.tp-link.com/br/home-networking/wifi-router/tl-wr829n/>. Acesso em: 05 jul. 2021.

WHITE, Curt M. **Data Communications and Computer Networks**: a business user's approach. 8. ed. Boston: Editora Cengage Learning, 2015. 456 p.

ZANCO, Wagner da Silva. **Microcontroladores PIC**: técnicas de software e hardware para projetos de circuitos eletrônicos com base no PIC16F877A. São Paulo: Editora Érica, 2006. 390 p.

ZONEEDIT. **Free DNS**. 2021. Disponível em: <https://www.zoneedit.com/free-dns/>. Acesso em: 15 jun. 2021.

APÊNDICE A – Código de programação (Apenas estrutura principal)

```

//-----BIBLIOTECAS-----
#include <string.h>
#include <xc.h>
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
#include <math.h>
#include <stdbool.h>
#include <stdint.h>
#include "lcd4_4L.h"
#include "configuracao.h"
#include "lcd_mensagens.h"
#include "ESP8266_functions.h"
//-----DEFINICOES-----
#define CIMA RB0
#define BAIXO RB1
#define SELECAO RB2
#define VOLTAR RB3
#define SENSOR01 RB4
#define MODOMANUAL RB5
//-----DECLARACAO DE FUNCOES-----
void teste_menu();
void teste_lampadas();
void teste_alarme();
void verifica_alarme(int);
void verifica_alarme_manual(int);
int POSICAO(int POS);
//-----VARIAVEIS GLOBAIS-----
char i;
int CONT;
int MENU; //Informa o menu atual
int AL; //Habilita o alarme
int ALARME; //Auxiliar do alarme
int DISPARO; //Auxiliar do alarme
int ATUALIZAR; //Atualiza o display
int MANUAL; //Habilita modo manual
char lamp[4]; //Acionamento de lâmpadas/LEDS
unsigned char meuip[4]; //IP
unsigned char data[30]; //IP
//-----PROGRAMA PRINCIPAL-----
void main()
{
    configuracao(); //Configuração do PIC
    inicializa_lcd4(); //Inicia LCD
    Initialize_ESP8266(); //Inicia ESP8266

    //Mensagem inicial
    texto_lcd4_4L ((char*)"Teste de conexao",0,0);
    texto_lcd4_4L ((char*)"ESP8266 com PIC",1,0);
    __delay_ms(1500);
    Limpa_LCD();

    //Verifica comunicação entre o PIC e o ESP
    do
    {
        //Enquanto não encontra o ESP
        texto_lcd4_4L ((char*)"ESP OFF",0,0);
    }

```

```

}
while (!esp8266_isStarted()); //ESP conectado
texto_lcd4_4L ((char*)"ESP conectado",0,0);
__delay_ms(1500);
Limpa_LCD();

//coloca o ESP no modo AP (modo=2) //Utilizado para testar uma rede Teste
//esp8266_mode(2); // Modo AP
//texto_lcd4_4L ((char*)"ESP modo 2",0,0);
//__delay_ms(1500);
//Limpa_LCD();

//coloca o ESP no modo Station (modo=1)
esp8266_mode(1); // Modo Station
texto_lcd4_4L ((char*)"ESP modo 1",0,0);
__delay_ms(1500);
Limpa_LCD();

//Conecta o ESP a rede existente
esp8266_connect("AUTOMACAO K13", "*****");
texto_lcd4_4L ((char*)"conexao OK",0,0);
__delay_ms(1500);
Limpa_LCD();

//Conecta ao IP fixo, Gateway e Mascara da sub-rede
esp8266_enderecoIP("192.168.1.150","192.168.1.1","255.255.255.0");
texto_lcd4_4L ((char*)"Endereco OK",0,0);
__delay_ms(1500);
Limpa_LCD();

//Configura um nome e uma senha para a rede
//esp8266_config_softAP("Rede Teste", "*****");
//texto_lcd4_4L ((char*)"Rede OK",0,0);
//__delay_ms(1500);
//Limpa_LCD();

//Permite múltiplas conexões TCP
_esp8266_enale_MUX();

//Cria um servidor TCP na porta 80 (HTML)
_esp8266_create_server();
texto_lcd4_4L ((char*)"WEB criada",0,0);
__delay_ms(1500);
Limpa_LCD();

//Carrega o IP no vetor de entrada e fornece do Display LCD
esp8266_ip(meuip);
sprintf(data,"IP:%d.%d.%d.%d",meuip[0],meuip[1],meuip[2],meuip[3]);
texto_lcd4_4L (data,0,0);

MENU=5; // INICIA EM MODO LOGIN
//LOOP principal para funcionamento do modo WEB
while(1)
{
  while (MENU==5)
  {
    //Teste visual para verificar o menu na placa através do LED
    teste_menu();
    //Envia pagina de login
    envia_pagina4();
  }
}

```

```

    //Recebe a resposta da página de login
    recebe_pagina4();
    Limpa_LCD();
    texto_lcd4_4L ((char*)"MODO WEB",0,0);
}

//Teste visual para verificar o menu na placa através do LED
teste_menu();
//Envia menu principal
envia_pagina3();
//Recebe a resposta do menu principal
recebe_pagina3();

//Se Alarme for selecionado...
if (MENU==1)
{
    while(MENU==1)
    {
        teste_alarme();
        envia_pagina2(AL);
        recebe_pagina2();
    }
}
//Se Lâmpadas for selecionado...
if (MENU==2)
{
    while(MENU==2)
    {
        teste_lampadas();
        envia_pagina();
        recebe_pagina();
    }
}
}
return;
}
}
}

//----- INTERRUPTÕES DE ALARME E MODO MANUAL-----
void interrupt_isr()
{
    //Se ocorreu mudança de estado RB4 até RB7
    if(RBIF==1)
    {
        //Se o sensor é acionado, faz o teste do alarme
        if(SENSOR01==1)
        {
            verifica_alarme(AL);
            envia_pagina2(AL);
            recebe_pagina2();
            __delay_ms(200);
        }
    }
    //-----MODO MANUAL-----
    if(MODOMANUAL==1)
    {
        CONT=0;
        MENU=0;
        MANUAL=0;
        TELA_INICIAL(CONT);
        //Enquanto o modo manual não é desabilitado
        while(MANUAL==0)
        {

```

```

//Menu Principal
if (ATUALIZAR==1)
{
  if (CONT<2)
  {
    TELA_INICIAL(CONT);
    ATUALIZAR=0;
  }
  else
  {
    TELA_INICIAL2(CONT);
    ATUALIZAR=0;
  }
}
if (CIMA==1 || BAIXO==1)
{
  POSICAO(CONT);
  ATUALIZAR=1;
}
//Menu da configuração de alarme
if (SELECAO==1 && CONT==0)
{
  MENU=1;
}
//Menu da configuração de Lâmpadas
if (SELECAO==1 && CONT==1)
{
  MENU=2;
  CONT=0;
  TELA_LAMP(CONT);
  __delay_ms(200);
}
//Menu para voltar a modo web
if (SELECAO==1 && CONT==2)
{
  MENU=3;
  __delay_ms(200);
}
//Verifica se os sensores foram ativados
if (SENSOR01==1)
{
  verifica_alarme_manual(AL);
}
//Menu alarme
while (MENU==1)
{
  Limpa_LCD();
  if (AL==0)
  {
    texto_lcd4_4L ((char*)"ALARME ACIONADO",0,0);
    AL=1;
    __delay_ms(1000);
    ALARME=1;
    ATUALIZAR=1;
  }
  else
  {
    texto_lcd4_4L ((char*)"ALARME DESLIGADO",0,0);
    AL=0;
    __delay_ms(1000);
  }
}

```

```

    ALARME=0;
    ATUALIZAR=1;
}
MENU=0;
ATUALIZAR=1;
}
//Menu do acionamento das lâmpadas
while (MENU==2)
{
    if (ATUALIZAR==1)
    {
        if (CONT<2)
        {
            TELA_LAMP(CONT);
            ATUALIZAR=0;
        }
        else
        {
            TELA_LAMP2(CONT);
            ATUALIZAR=0;
        }
    }
    if (CIMA==1 || BAIXO==1)
    {
        POSICAO(CONT);
        ATUALIZAR=1;
        __delay_ms(100);
    }
    POSICAO(CONT);
    if (VOLTAR==1)
    {
        MENU=0;
        ATUALIZAR=1;
        CONT=0;
    }
    if (SELECAO==1 && CONT==0)
    {
        if (lamp[0]==0)
        {
            RC0=1;
            lamp[0]=1;
            __delay_ms(200);
        }
        else
        {
            RC0=0;
            lamp[0]=0;
            __delay_ms(200);
        }
    }
    if (SELECAO==1 && CONT==1)
    {
        if (lamp[1]==0)
        {
            RC1=1;
            lamp[1]=1;
            __delay_ms(200);
        }
        else
        {

```



```

    RBIF=0; // Zerar a flag do RB
}
}
//-----FUNCOES-----
//TESTE DA PAGINA MENU
void teste_menu()
{
    for (i=0;i<=10;i++)
    {
        RA0=1; //LED01 ON
        __delay_ms(50);
        RA0=0; //LED01 OFF
        __delay_ms(50);
    }
}
//TESTE DA PAGINA LAMPADAS
void teste_lampadas()
{
    for (i=0;i<=5;i++)
    {
        RA0=1; //LED01 ON
        __delay_ms(100);
        RA0=0; //LED01 OFF
        __delay_ms(100);
    }
}
//TESTE DA PAGINA DO ALARME
void teste_alarme()
{
    for (i=0;i<=3;i++)
    {
        RA0=1; //LED01 ON
        __delay_ms(200);
        RA0=0; //LED01 OFF
        __delay_ms(200);
    }
}
//VERIFICACAO DO ALARME QUANDO O SENSOR É ACIONADO
void verifica_alarme(AL)
{
    if (AL==1)
    {
        while (DISPARO==0)
        {
            RC3=1;
            TELA_AL_DISP();
            //RC4=1;
            envia_pagina2(AL);
            AL=recebe_pagina2();
            //__delay_ms(500);
            if (AL==0 || RB3==1)
            {
                DISPARO=1;
                RC3=0;
                //RC4=0;
                Limpa_LCD();
                texto_lcd4_4L ((char*)"MODO WEB",0,0);
            }
        }
        DISPARO=0;
    }
}

```

```

    }
}
//VERIFICACAO DO ALARME QUANDO O SENSOR É ACIONADO
void verifica_alarme_manual(AL)
{
    if (AL==1)
    {
        while (DISPARO==0)
        {
            RC3=1;
            //RC4=1;
            TELA_AL_DISP();
            if (AL==0 || RB3==1)
            {
                DISPARO=1;
                RC3=0;
                //RC4=0;
                ATUALIZAR=1;
            }
        }
        DISPARO=0;
    }
}
//POSICAO DO CURSOR NO DISPLAY LCD
int POSICAO(int POS)
{
    CONT=POS;
    if (CIMA==1)
    {
        CONT=CONT-1;
        __delay_ms(200);
    }
    if (BAIXO==1)
    {
        CONT=CONT+1;
        __delay_ms(200);
    }
    return CONT;
}
//-----FIM-----

```

APÊNDICE B – Código de programação (Biblioteca ESP-8266)

```

//-----BIBLIOTECAS-----
#include <stdbool.h>
#include <stdint.h>
#include <stdio.h>
#include <string.h>

//-----DEFINICOES-----
#define ESP8266_STATION 0x01
#define ESP8266_SOFTAP 0x02
#define ESP8266_TCP 1
#define ESP8266_UDP 0
//PARA RESPOSTAS DO ESP
#define ESP8266_OK 1
#define ESP8266_READY 2
#define ESP8266_FAIL 3
#define ESP8266_NOCHANGE 4
#define ESP8266_LINKED 5
#define ESP8266_UNLINK 6
//PARA LAMPADAS
#define LED1ON 1
#define LED1OFF 2
#define LED2ON 3
#define LED2OFF 4
#define LED3ON 5
#define LED3OFF 6
#define LED4ON 7
#define LED4OFF 8
#define LAMPADASVOLTAR 9
#define SENHAINC 1
#define SENHACOR 2
//PARA ALARME
#define ALARMEON 1
#define ALARMEOFF 2
#define ALARMEVOLTAR 3
//PARA MENU
#define MENU01 1
#define MENU02 2
#define MENU03 3
#define MENU04 4

//-----VARIAVEIS GLOBAIS-----
int len;

//-----DECLARACAO DE FUNCOES-----
void Initialize_ESP8266(void); //FUNCAO PARA INICIALIZAR ESP-8266
bit esp8266_isStarted(void); // VERIFICA SE FOI INICIADO O ESP (AT)
bit esp8266_restart(void); // REINICIA O MODULO ESP (AT+RST)
void esp8266_mode(unsigned char); // CONFIGURA O MODO WIFI(AT+CWMODE)
unsigned char esp8266_config_softAP(unsigned char* softssid, unsigned char* softpass); //NOME E
SENHA REDE
void _esp8266_enale_MUX(); // PERMITE MULTIPLAS CONEXOES
void _esp8266_create_server(); //CRIA SERVIDOR NA PORTA 80
void esp8266_connect(unsigned char*, unsigned char*); //CONECTA O ESP A REDE WIFI
void esp8266_enderecoIP(unsigned char*, unsigned char*, unsigned char*); //CONECTA AO IP
void esp8266_ip(char*); //INFORMA O IP LOCAL (AT+CIFSR)
void _esp8266_putch(unsigned char); //ENVIA UM BYTE PARA UART (ESCREVER)
unsigned char _esp8266_getch(void); //RECEBE UM BYTE DA UART

```

```

void _esp8266_print(unsigned const char *);           //PRINT PARA A WEB
inline uint16_t _esp8266_waitFor(unsigned char *);  //AGUARDA POR AQUELA STRING
bit esp8266_start(unsigned char protocol, char* ip, unsigned char port);
void envia_pagina();                                //ENVIA PAGINA LAMPADAS
void envia_pagina2(int AL);                         //ENVIA PAGINA ALARME
void envia_pagina3();                              //ENVIA PAGINA MENU PRINCIPAL
void envia_pagina4();                              //ENVIA PAGINA DE LOGIN
int recebe_pagina();                               //RECEBE PAGINA LAMPADAS
int recebe_pagina2();                             //RECEBE PAGINA ALARME
int recebe_pagina3();                             //RECEBE PAGINA MENU PRINCIPAL
int recebe_pagina4();                             //RECEBE PAGINA DE LOGIN
inline unsigned char _esp8266_aguarda_resp(void);  //RESPOSTAS DO ESP
inline unsigned char _esp8266_aguarda_resp2(void); //RESPOSTAS LAMPADAS
inline unsigned char _esp8266_aguarda_resp3(void); //RESPOSTAS ALARME
inline unsigned char _esp8266_aguarda_resp4(void); //RESPOSTAS MENU PRINCIPAL
inline unsigned char _esp8266_aguarda_resp5(void); //RESPOSTAS DA PAGINA DE LOGIN

```

//-----FUNCOES-----

//FUNCAO QUE INICIALIZA O ESP-8266

```

void Initialize_ESP8266(void)
{
    //****Setting I/O pins for UART****//
    TRISC6 = 0; // TX Pin set as output
    TRISC7 = 1; // RX Pin set as input
    //_____I/O pins set _____//

    /**Initialize SPBRG register for required
    baud rate and set BRGH for fast baud_rate**/
    SPBRG = 10;
    BRGH = 1; // for high baud_rate
    //_____End of baud_rate setting_____//

    //****Enable Asynchronous serial port*****//
    SYNC = 0; // Asynchronous
    SPEN = 1; // Enable serial port pins
    //_____Asynchronous serial port enabled_____//
    /**Lets prepare for transmission & reception**//
    TXEN = 1; // enable transmission
    CREN = 1; // enable reception
    //__UART module up and ready for transmission and reception__//

    /**Select 8-bit mode**//
    TX9 = 0; // 8-bit reception selected
    RX9 = 0; // 8-bit reception mode selected
    //__8-bit mode selected__//
}
// VERIFICA SE FOI INICIADO O ESP (AT)
bit esp8266_isStarted(void) {
    _esp8266_print("AT\r\n");
    return (_esp8266_aguarda_resp() == ESP8266_OK);
}
//*****
// REINICIA O ESP
bit esp8266_restart(void) {
    _esp8266_print("AT+RST\r\n");
    if (_esp8266_aguarda_resp() != ESP8266_OK) {
        return false;
    }
    return (_esp8266_aguarda_resp() == ESP8266_READY);
}

```

```

//CONFIGURA O MODO WIFI (station/softAP/station+softAP) (AT+CWMODE)
void esp8266_mode(unsigned char mode) {
    _esp8266_print("AT+CWMODE=");
    _esp8266_putchar(mode + '0');
    _esp8266_print("\r\n");
    _esp8266_aguarda_resp();
}
//NOME E SENHA REDE
unsigned char esp8266_config_softAP(unsigned char* softssid, unsigned char* softpass)
{
    _esp8266_print("AT+CWSAP=");
    _esp8266_print(softssid);
    _esp8266_print("\",\"");
    _esp8266_print(softpass);
    _esp8266_print("\",5,3\r\n");
    return _esp8266_aguarda_resp();
}
//PERMITE MULTIPLAS CONEXOES
void _esp8266_enale_MUX()
{
    _esp8266_print("AT+CIPMUX=1\r\n"); //Enable Multiple Connections
    _esp8266_aguarda_resp();
}
//CRIA SERVIDOR NA PORTA 80
void _esp8266_create_server()
{
    _esp8266_print("AT+CIPSERVER=1,80\r\n"); //Enable Server on port 80
    _esp8266_aguarda_resp();
}
//CONECTA O ESP8266 A UMA REDE WI-FI
void esp8266_connect(unsigned char* ssid, unsigned char* pass)
{
    _esp8266_print("AT+CWJAP=");
    _esp8266_print(ssid);
    _esp8266_print("\",\"");
    _esp8266_print(pass);
    _esp8266_print("\",\r\n");
    _esp8266_aguarda_resp();
}
//CONECTA O ESP8266 A UMA REDE WI-FI
void esp8266_enderecoIP(unsigned char* IP, unsigned char* gateway, unsigned char* mascara)
{
    _esp8266_print("AT+CIPSTA_CUR=");
    _esp8266_print(IP);
    _esp8266_print("\",\"");
    _esp8266_print(gateway);
    _esp8266_print("\",\"");
    _esp8266_print(mascara);
    _esp8266_print("\",\r\n");
    _esp8266_aguarda_resp();
}
//INFORMA O IP LOCAL
void esp8266_ip(unsigned char* store_in) {
    _esp8266_print("AT+CIFSR\r\n");
    unsigned char received;
    do {
        received = _esp8266_getch();
    } while (received < '0' || received > '9');
    for (unsigned char i = 0; i < 4; i++) {
        store_in[i] = 0;
    }
}

```

```

do {
    store_in[i] = 10 * store_in[i] + received - '0';
    received = _esp8266_getch();
} while (received >= '0' && received <= '9');
received = _esp8266_getch();
}
_esp8266_waitFor("OK");
}
//ENVIA UM BYTE PARA UART (USADA PARA ESCREVER NA WEB)
void _esp8266_putch(char bt)
{
    while(!TXIF); // hold the program till TX buffer is free
    TXREG = bt; //Load the transmitter buffer with the received value
}
//RECEBE UM BYTE DA UART
char _esp8266_getch()
{
    if(OERR) // check for Error
    {
        CREN = 0; //If error -> Reset
        CREN = 1; //If error -> Reset
    }

    while(!RCIF); // hold the program till RX buffer is free

    return RCREG; //receive the value and send it to main function
}
//PRINT PARA A WEB
void _esp8266_print(unsigned const char *ptr) {
    while (*ptr != 0) {
        _esp8266_putch(*ptr++);
    }
}
//AGUARDA POR AQUELA STRING
inline uint16_t _esp8266_waitFor(unsigned char *string) {
    unsigned char so_far = 0;
    unsigned char received;
    uint16_t counter = 0;
    do {
        received = _esp8266_getch();
        counter++;
        if (received == string[so_far]) {
            so_far++;
        } else {
            so_far = 0;
        }
    } while (string[so_far] != 0);
    return counter;
}
//ABRE CONEXAO TCP OU UDP
bit esp8266_start(unsigned char protocol, char* ip, unsigned char port) {
    _esp8266_print("AT+CIPSTART=");
    if (protocol == ESP8266_TCP) {
        _esp8266_print("TCP");
    } else {
        _esp8266_print("UDP");
    }
    _esp8266_print("\",");
    _esp8266_print(ip);
    _esp8266_print("\",");
}

```

```

unsigned char port_str[5] = "\0\0\0\0";
sprintf(port_str, "%u", port);
_esp8266_print(port_str);
_esp8266_print("\r\n");
if (_esp8266_aguarda_resp() != ESP8266_OK) {
    return 0;
}
if (_esp8266_aguarda_resp() != ESP8266_LINKED) {
    return 0;
}
return 1;
}
//RESPOSTAS DO ESP
inline unsigned char _esp8266_aguarda_resp(void) {
    unsigned char so_far[6] = {0,0,0,0,0,0};
    unsigned const char lengths[6] = {2,5,4,9,6,6};
    unsigned const char strings[6] = {"OK", "ready", "FAIL", "no change", "Linked", "Unlink"};
    unsigned const char responses[6] = {ESP8266_OK, ESP8266_READY, ESP8266_FAIL,
    ESP8266_NOCHANGE, ESP8266_LINKED, ESP8266_UNLINK};
    unsigned char received;
    unsigned char response;
    bool continue_loop = true;
    while (continue_loop) {
        received = _esp8266_getch();
        for (unsigned char i = 0; i < 6; i++) {
            if (strings[i][so_far[i]] == received) {
                so_far[i]++;
                if (so_far[i] == lengths[i]) {
                    response = responses[i];
                    continue_loop = false;
                }
            } else {
                so_far[i] = 0;
            }
        }
    }
    return response;
}
//-----LAMPADAS-----
//PAGINA DAS LAMPADAS
void envia_pagina()
{
    _esp8266_print("AT+CIPSEND=0,832\r\n"); //esta sendo subtraido -9
    __delay_ms(300);
    _esp8266_print("<html><head><title>lampadas</title><meta                http-equiv='refresh'
content='8' ></head><h1>Menu Lampadas</h1>"); //104
    _esp8266_print("<body bgcolor='#FFF8DC'><p><form method='GET'></p>"); //52
    _esp8266_print("<p><font color='green'><input type='submit' name='B' value='1' > LED1
ON</p>"); //77
    _esp8266_print("<p><font color='red'><input type='submit' name='B' value='2' > LED1
OFF</p>"); //76
    _esp8266_print("<p><font color='green'><input type='submit' name='B' value='3' > LED2
ON</p>"); //77
    _esp8266_print("<p><font color='red'><input type='submit' name='B' value='4' > LED2
OFF</p>"); //76
    _esp8266_print("<p><font color='green'><input type='submit' name='B' value='5' > LED3
ON</p>"); //77
    _esp8266_print("<p><font color='red'><input type='submit' name='B' value='6' > LED3
OFF</p>"); //76
    _esp8266_print("<p><font color='green'><input type='submit' name='B' value='7' > LED4

```

```

ON</p>");//77
_esp8266_print("<p><font color=\"red\"><input type=\"submit\" name=\"B\" value=\"8\" > LED4
OFF</p>");//76
_esp8266_print("<p><font color=\"blue\"><input type=\"submit\" name=\"B\" value=\"9\" >
VOLTAR</p>");//77
_esp8266_print("AT+CIPCLOSE=0\r\n");
__delay_ms(250);
_esp8266_print("AT+CIPCLOSE=0\r\n");
__delay_ms(250);
}
//RECEBE INFORMACAO SOBRE ACIONAMENTO DE LAMPADAS
int recebe_pagina()
{
  switch (_esp8266_aguarda_resp2())
  {
    case (LED1ON):
    {
      RC0=1;
    }
    break;

    case (LED1OFF):
    {
      RC0=0;
    }
    break;

    case (LED2ON):
    {
      RC1=1;
    }
    break;

    case (LED2OFF):
    {
      RC1=0;
    }
    break;

    case (LED3ON):
    {
      RC2=1;
    }
    break;

    case (LED3OFF):
    {
      RC2=0;
    }
    break;

    case (LED4ON):
    {
      RC3=1;
    }
    break;

    case (LED4OFF):
    {
      RC3=0;
    }
  }
}

```

```

    }
    break;

    case (LAMPADASVOLTAR):
    {
        MENU=0;
        return MENU;
    }
    break;
}
}
//RESPOSTAS UTILIZADAS PARA AS LAMPADAS
inline unsigned char _esp8266_aguarda_resp2(void)
{
    unsigned char so_far[7] = {0,0,0,0,0,0,0}; //quantidade de respostas
    unsigned const char lengths[7] = {3,3,3,3,3,3,3}; //tamanho das respostas
    unsigned const char* strings[7] = {"B=9", "B=6", "B=5", "B=4", "B=3", "B=2", "B=1"}; //respostas
    unsigned const char responses[7] = {LAMPADASVOLTAR, LED3OFF, LED3ON, LED2OFF,
    LED2ON, LED1OFF, LED1ON};
    unsigned char received;
    unsigned char response;
    bool continue_loop = true;
    while (continue_loop) {
        received = _esp8266_getch();
        for (unsigned char i = 0; i < 7; i++) {
            if (strings[i][so_far[i]] == received) {
                so_far[i]++;
                if (so_far[i] == lengths[i]) {
                    response = responses[i];
                    continue_loop = false;
                }
            }
            else {
                so_far[i] = 0;
            }
        }
    }
    return response;
}
//-----ALARME-----
//PAGINA DO ALARME
void envia_pagina2(int AL)
{
    _esp8266_print("AT+CIPSEND=0,433\r\n"); //esta sendo subtraido -9
    __delay_ms(300);
    _esp8266_print("<html><head><title>Alarme</title><meta http-equiv=\"refresh\"
content=\"8\" ></head><h1>Menu Alarme</h1>");//100
    _esp8266_print("<body bgcolor=\"#FFF8DC\"><p><form method=\"GET\"></p>"); //53
    _esp8266_print("<p><font color=\"green\"><input type=\"submit\" name=\"B\" value=\"1\" > ATIVAR
ALARME</p>");//83
    _esp8266_print("<p><font color=\"red\"><input type=\"submit\" name=\"B\" value=\"2\" > DESATIVAR
ALARME</p>");//86
    _esp8266_print("<p><font color=\"blue\"><input type=\"submit\" name=\"B\" value=\"9\" >
VOLTAR</p>");//77
    //AL=1; - ACIONA A VERIFICAÇÃO DO ALARME
    if (RC3==1 && AL==1) //QUANDO RC4=1, ALARME DISPARA
    {
        _esp8266_print( " <p>O ALARME DISPAROU! </p></body></html>\r\n"); //27
    }
    if (RC3==0 && AL==1) //QUANDO RC4=1, ALARME DISPARA

```

```

    {
        _esp8266_print( "<p>MONITORANDO ALARME ON</p></body></html>\r\n"); //27
    }
    else _esp8266_print("<p>ALARME DESATIVADO OFF</p></body></html>\r\n"); //28

    _esp8266_print("AT+CIPCLOSE=0\r\n");
    __delay_ms(250);
    _esp8266_print("AT+CIPCLOSE=0\r\n");
    __delay_ms(250);
}
//RECEBE INFORMACAO SE ALARME ESTA ACIONADO OU NAO
int recebe_pagina2()
{
    switch (_esp8266_aguarda_resp3())
    {
        case (ALARMEVOLTAR):
        {
            //AL=2;
            MENU=0;
            //return AL;
        }
        break;

        case (ALARMEON):
        {
            AL=1;
            return AL;
        }
        break;

        case (ALARMEOFF):
        {
            AL=0;
            return AL;
        }
        break;
    }
}
//RESPOSTAS UTILIZADAS PARA O MENU ALARME
inline unsigned char _esp8266_aguarda_resp3(void)
{
    unsigned char so_far[3] = {0,0,0}; //quantidade de respostas
    unsigned const char lengths[3] = {3,3,3}; //tamanho das respostas
    unsigned const char* strings[3] = {"B=9", "B=2", "B=1"}; //respostas
    unsigned const char responses[3] = {ALARMEVOLTAR, ALARMEOFF, ALARMEON};
    unsigned char received;
    unsigned char response;
    bool continue_loop = true;
    while (continue_loop) {
        received = _esp8266_getch();
        for (unsigned char i = 0; i < 3; i++) {
            if (strings[i][so_far[i]] == received) {
                so_far[i]++;
                if (so_far[i] == lengths[i]) {
                    response = responses[i];
                    continue_loop = false;
                }
            }
        }
        else {
            so_far[i] = 0;
        }
    }
}

```

```

    }
}
return response;
}

//-----MENU PRINCIPAL-----
//PAGINA MENU PRINCIPAL
void envia_pagina3()
{
    _esp8266_print("AT+CIPSEND=0,378\r\n"); //esta sendo subtraido 8
    __delay_ms(300);
    _esp8266_print("<html><head><title>Principal</title><meta                http-equiv=\"refresh\"
content=\"8\" ></head><h1>Menu Principal</h1>"); //104
    _esp8266_print("<body bgcolor=\"#FFF8DC\"><p><form method=\"GET\"></p>"); //52
    _esp8266_print("<p><font color=\"green\"><input type=\"submit\" name=\"B\" value=\"1\" >
ALARME</p>"); //80
    _esp8266_print("<p><font color=\"red\"><input type=\"submit\" name=\"B\" value=\"2\" >
LAMPADAS</p>"); //76
    _esp8266_print("<p><font color=\"blue\"><input type=\"submit\" name=\"B\" value=\"3\" >
SAIR</p>"); //76
    _esp8266_print("AT+CIPCLOSE=0\r\n");
    __delay_ms(250);
    _esp8266_print("AT+CIPCLOSE=0\r\n");
    __delay_ms(250);
}
//RECEBE INFORMACAO SOBRE O MENU QUE FOI SELECIONADO
int recebe_pagina3()
{
    switch (_esp8266_aguarda_resp4())
    {
        case (MENU01):
        {
            MENU=1;
            return MENU;
        }
        break;

        case (MENU02):
        {
            MENU=2;
            return MENU;
        }
        break;

        case (MENU03):
        {
            MENU=5;
            return MENU;
        }
        break;

        case (MENU04):
        {
            MENU=0;
            return MENU;
        }
        break;
    }
}
}

```

```
//RESPOSTAS UTILIZADAS PARA O MENU PRINCIPAL
```

```
inline unsigned char _esp8266_aguarda_resp4(void)
{
    unsigned char so_far[4] = {0,0,0,0}; //quantidade de respostas
    unsigned const char lengths[4] = {3,3,3,3}; //tamanho das respostas
    unsigned const char* strings[4] = {"B=9", "B=3", "B=2", "B=1"}; //respostas
    unsigned const char responses[4] = {MENU04, MENU03, MENU02, MENU01};
    unsigned char received;
    unsigned char response;
    bool continue_loop = true;
    while (continue_loop) {
        received = _esp8266_getch();
        for (unsigned char i = 0; i < 4; i++) {
            if (strings[i][so_far[i]] == received) {
                so_far[i]++;
                if (so_far[i] == lengths[i]) {
                    response = responses[i];
                    continue_loop = false;
                }
            } else {
                so_far[i] = 0;
            }
        }
    }
    return response;
}
```

```
//-----TELA DE LOGIN-----
```

```
//PAGINA MENU PRINCIPAL
```

```
void envia_pagina4()
{
    _esp8266_print("AT+CIPSEND=0,247\r\n"); //esta sendo subtraido 8
    __delay_ms(300);
    _esp8266_print("<html><head><title>Login</title><meta http-equiv='refresh' "
content="15" ></head><h1>Login</h1>"); //104
    _esp8266_print("<body bgcolor='#FFF8DC'><p><form method='GET'><p>"); //52
    _esp8266_print("<p><font color='green'><input type='Password' name='B' "
id="Pass" ><p>"); //80
    _esp8266_print("<p>DIGITE A SUA SENHA </p></body></html>\r\n"); //27
    _esp8266_print("AT+CIPCLOSE=0\r\n");
    __delay_ms(250);
    _esp8266_print("AT+CIPCLOSE=0\r\n");
    __delay_ms(250);
}
```

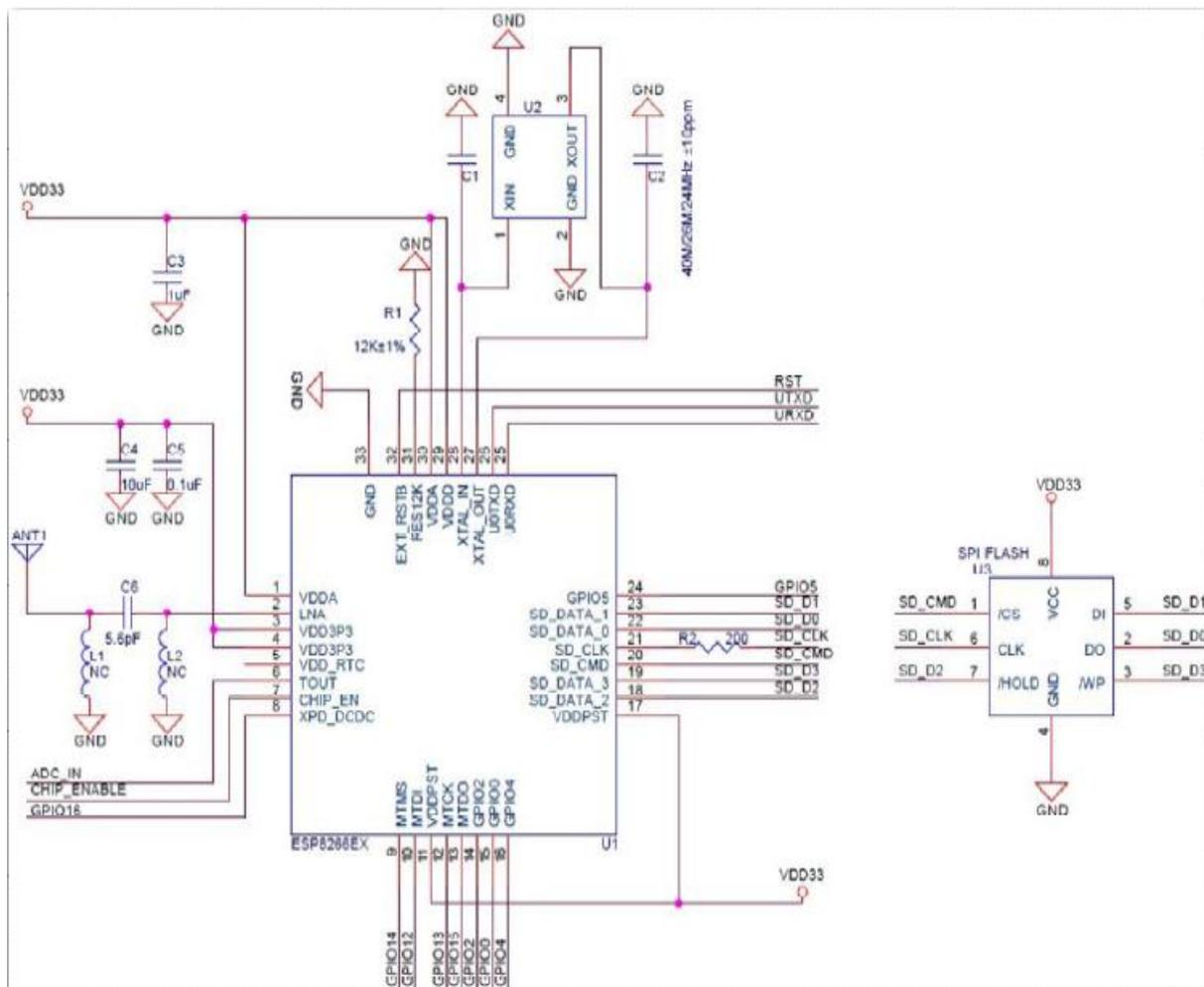
```
//RECEBE INFORMACAO SOBRE O MENU DE LOGIN
```

```
int recebe_pagina4()
{
    switch (_esp8266_aguarda_resp5())
    {
        case (SENHACOR):
        {
            MENU=7;
            return MENU;
        }
        break;

        case (SENHAINC):
        {
            MENU=5;
            return MENU;
        }
    }
}
```

```
        break;
    }
}
//RESPOSTAS UTILIZADAS PARA O LOGIN
inline unsigned char _esp8266_aguarda_resp5(void)
{
    unsigned char so_far[2] = {0,0}; //quantidade de respostas
    unsigned const char lengths[2] = {5,3}; //tamanho das respostas
    unsigned const char* strings[2] = {"B=987","B=1"}; //respostas
    unsigned const char responses[2] = {SENHACOR,SENHAINC};
    unsigned char received;
    unsigned char response;
    bool continue_loop = true;
    while (continue_loop) {
        received = _esp8266_getch();
        for (unsigned char i = 0; i < 2; i++) {
            if (strings[i][so_far[i]] == received) {
                so_far[i]++;
                if (so_far[i] == lengths[i]) {
                    response = responses[i];
                    continue_loop = false;
                }
            } else {
                so_far[i] = 0;
            }
        }
    }
    return response;
}
//FIM
```


ANEXO A – Esquemático do ESP-01





TERMO DE ACEITE PARA DIVULGAÇÃO

Eu, Kleison Antonio Possenti, aluno(a) do curso de Bacharelado em Engenharia Elétrica do IFSC, Câmpus Joinville, aceito e autorizo a divulgação e/ou publicação do meu TCC – Trabalho de Conclusão de Curso, que contém os seguintes dados:

Título:

Projeto de Controle e Automação Residencial Através do Monitoramento de uma Página WEB.

Professor Orientador: Nivaldo Theodoro Schiefler Junior, Dr.

Data: 22 / 02 / 2022.

Kleison A. Possenti

Assinatura do(a) Aluno(a)