DEMONSTRAÇÃO DE AUTOMAÇÃO RESIDENCIAL COM ARDUINO

José Domingos Da Silva Oliveira[1]
Bruno Ramon De Almeida[2]
Junior Marcos Bandeira[3]

Resumo: O artigo em questão, demonstra automação residencial em uma plataforma alternativa chamada Arduino. A automação residencial, considerada uma tendência, já é uma realidade na vida das pessoas, e esse artigo visa apresentar um sistema mais acessível, capaz de satisfazer necessidades peculiares de uma casa. Uma maquete foi criada para simular um ambiente residencial e demonstrar a possível aplicação do Arduino em um sistema centralizado, relatando um conjunto de ações e informações que podem ser executadas e extraídas no ambiente que está sendo aplicada. Ao finalizar o desenvolvimento do projeto, o sistema de automação foi submetido a testes demonstrando estabilidade na disponibilização e integridade dos dados coletados e segurança na execução de ações solicitadas pelo usuário.

Palavras chave: Automação residencial, Arduino, Sistema centralizado.

Abstract: The article in question demonstrates residential automation on an alternative platform called Arduino. Residential automation, considered a trend, is already a reality in people's lives, and this article aims to present a more accessible system, able to meet the peculiar needs of a home. A model was created to simulate a residential environment and demonstrate the possible application of the Arduino in a centralized system, reporting a set of actions and information that can be executed and extracted in the environment being applied. At the end of the project development, the automation system was submitted to tests demonstrating stability in the availability and integrity of the data collected and security in the execution of actions requested by the user.

Key words: Residential automation, Arduino, Centralized system.

1. INTRODUÇÃO

Automação, segundo o dicionário (2015), trata-se de uma ou grupo de máquinas sob controle de um único sistema centralizado, que permite efetuar operações contábeis, estatísticas ou industriais sem intervenção humana.

A automação aplica-se a diversas situações conforme a cada ambiente, assim como é aplicada no setor industrial também aplica-se a residências com intenção de auxiliar ou até mesmo dispensar o esforço humano (ALVES, MOTA, 2003). Na década de 1920 quando surgiram os primeiros eletrodomésticos a grande promessa seria que eles poupariam o tempo das pessoas em tarefas do cotidiano. Os fabricantes já usavam o termo casa do futuro, mas ainda hoje há dúvidas por parte

[1] Autor – Sistemas de Informação, Unibalsas Faculdade de Balsas, jose.tecnoinf@gmail.com.
[2] Orientador – Professor de Sistemas de Informação Unibalsas Faculdade de Balsas
[3] Orientador – Professor de Sistemas de Informação Unibalsas Faculdade de Balsas

de alguns quanto aos benefícios de interconectar cada eletrodoméstico, permitindo o monitoramento e comando remoto de todos. Porém, a aceitação tem sido crescente no ambiente residencial ampliando benefícios como: Conforto, economia, prevenção de acidentes e falhas de equipamentos, gerenciamento técnico e segurança aos usuários. (ALVES, MOTA, 2003)

Acredita-se que a Automação residencial não é mais um item de luxo e sim uma necessidade (FRANCISCO, TREVISANI, 2013). Pensando desta forma é possível analisar que a melhor maneira de tornar acessível a execução dessas "tarefas comuns" seria apenas aproximando as pessoas dessas ações.

Francisco e Trevisani (2013) afirmam que, na atualidade existem diversas plataformas utilizadas na automação residencial, uma das primeiras tecnologias que surgiram foi o X10, que funciona através de mensagens simplificadas tornando o protocolo leve, assim como os demais, *Insteon* e *RedEye*, que levam uma desvantagem por terem a arquitetura fechada dificultando a comunicação de dispositivos fabricados por terceiros e limitando possíveis funcionalidades que possam somar significativamente em um projeto. Com a proposta de utilizar tecnologia de código livre o Arduíno destaca-se por sua compatibilidade e suporte a inúmeras plataformas e seu baixo custo, com ele pode-se programar leds, displays, sensores e motores, tornando ilimitado as possibilidades de criar soluções para diversos fins.

Contudo, o objetivo dessa pesquisa é demonstrar que é possível criar soluções de automação residencial com Arduino, e partindo desse objetivo pode-se obter conhecimento de forma global sobre automação e o Arduino no decorrer do artigo. Para demonstrar como o Arduino se porta ao gerir uma casa, esse projeto utiliza uma maquete em MDF [4] de 3mm para simular um ambiente residencial e aplicar ações controladas pelo usuário e executadas pelo Arduino.

2. AUTOMAÇÃO RESIDENCIAL

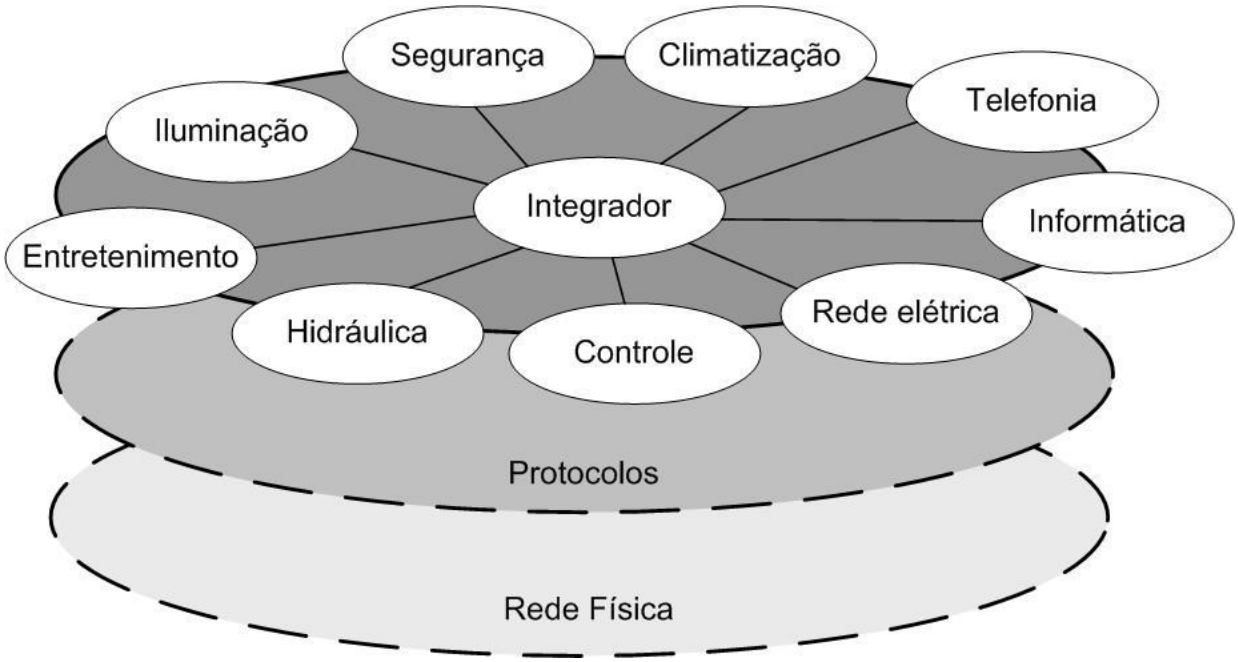
A automação residencial também chamada de Domotica é uma tendência do século 21, (BOLZANI, 2004) afirma que conferir E-mails, acessar dados de um ERP [5] e imprimir documentos são tarefas automatizadas, criadas pensando em poupar as pessoas de problemas simples para um mundo moderno e imediato. O

[4] (Medium Density Fiberboard) traduzido para português significa Painel de fibra de densidade média
[5] Sistema de Planejamento de Recursos Empresariais

autor acredita que a vida será muito complicada se existir mais senhas, chaves e botõezinhos, e a demótica da possibilidade de unificar o controle de equipamentos elétricos conectados entre si, permitindo controlar a residência remotamente, poupa tempo, energia, dinheiro e principalmente, dar conforto.

No geral cada aparelho elétrico exerce apenas sua função isolada. Bolzani, (2004) acredita-se que a automação residencial quer quebrar esse conceito, e conectar esses dispositivos a partir de um controlador centralizado, denominado Integrador conforme ilustra a Figura 1.

Figura 1 - Conceito de automação residencial



Fonte: Bolzani (2004)

A infraestrutura doméstica padrão é composta por funções independes, redes não compatíveis, falta de uniformidade e equipamentos limitados, responsáveis por problemas como: multiplicidade de rede e cabos, manutenção cara e complicada, impossibilidade de automação global e dificuldade de interligar novos serviços e redes, Bolzani, (2004).

A proposta de infraestrutura automatizada centralizada relaciona alguns pontos positivos citados também por Bolzani, (2004), entre eles está o maior conforto e automatização de serviço, barateamento de equipamentos e processos, simplificação da rede, acesso a informação de qualquer ponto da casa, economia

de energia e auxilio na administração da residência com constante supervisão do conjunto de equipamentos.

Banzi, (2011) deixou bem claro que o Arduino foi criado no intuído de motivar o conhecimento, mas sua capacidade de aplicação vai muito além, e ao discorrer deste capitulo veremos o quanto a plataforma se adapta perfeitamente ao sistema de automação residência proposto. Pois o Arduino é uma plataforma de computação física de código fonte aberto, tendo como base uma placa simples de entrada/saída, possibilitando acionamento de atuadores e sensores necessários para colher informações do ambiente residencial.

3. PLATAFORMA ARDUINO E COMPONENTES

O Arduino surgiu em 2005 na Itália, segundo Mcroberts, (2011) especificamente na *Interaction Design Institute* (Instituto Designer de interação) na cidade de Ivrea. Desenvolvido pelo professor Massimo Banzi que procurava uma forma mais fácil e barata para seus alunos estudarem tecnologia, e após discutir sobre o problema com o David Cuartielles (pesquisador visitante da Universidade de *Malmö*) o Arduino nasceu.

A primeira versão do Arduino foi construído baseado no microprocessador ATMEGA8 com memória flash de 8Kb segundo Mcroberts, (2011), com constante evolução para o mais popular e mais vendido, Arduino UNO.

O Arduino tem uma dualidade segundo Medeiros; Guimaraes; Placco, (2013) é uma combinação de hardware e software capaz de controlar as entradas e saídas. Medeiros; Guimaraes; Placco, (2013) continuam afirmando que o software é gerado em uma IDE programada em Java. Mas não existe apenas uma versão do Arduino, existem alguns modelos oficiais e inúmeros criados por terceiros, já que se trata de um projeto com código fonte aberto. Nesta pesquisa é abordada apenas uma versão oficial, o Arduino Mega 2560.

3.1. Arduino Mega

O Arduino Mega segundo Mcroberts, (2011), lançado anteriormente com microprocessador ATmega1280 foi atualizado para o ATmega2560 juntamente como Arduino Uno. O ATmega2560 é capacitado com memória flash de 256KB.

Mcroberts, (2011) faz uma descrição dos componentes da versão Mega, que foi criada afim de servir grandes projetos que necessitam de muitas entradas

e saídas, capaz de comportar uma grande quantidade de leds. A versão conta com quatro portas serial de hardware, 54 pinos digitais de in/out, 14 com capacidade de fornecer saída analógica PWM, e 16 pinos de entrada analógica. A comunicação de suporte para dispositivos I2C/TWI e SPI estão disponíveis, além de possuir um conector ICSP e um botão reset, para limpar a programação gravada no Arduino. Mcroberts, (2011) não deixar esquecer que o Arduino Mega possuir o chiset ATmega8U2, possibilitando cominação USB.

Na Figura 02 demonstra o esquema de pinos e demais componentes do Arduino Mega.

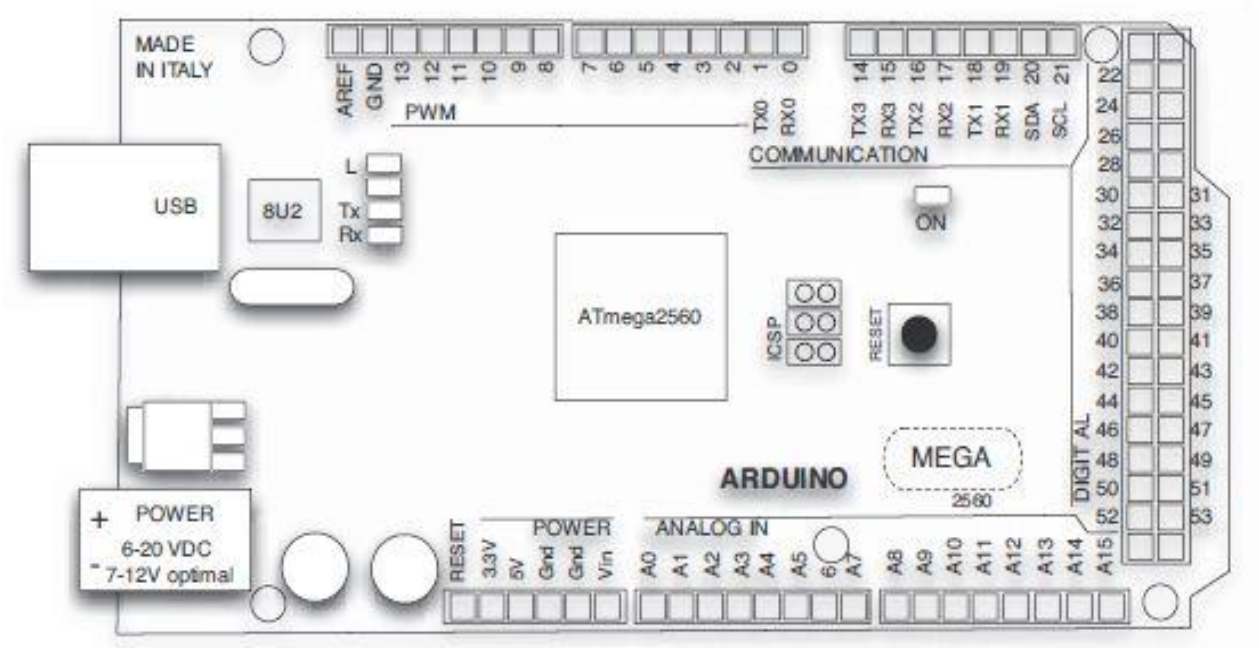


Figura 02 - Layout da placa Arduino Mega

Fonte: Mcroberts (2015)

A aplicação do projeto cria a necessidade de utilização de um Ethernet Shield de acordo com Camargo (2012 apud LIMA, 2013), *Shield* no inglês significa escudo, e trata-se de placas de extensão para o Arduino. O Ethernet Shield da capacidade ao Arduino de realizar conexão com a Internet ou simplesmente uma rede interna, através de um conector RJ45. O extensor ainda conta com um conector para cartão de memória micros SD, permitindo armazenamento de dados. Essa extensão tem aparência similia com a do Arduino com demonstra Figura 03, pois se encaixa perfeitamente sobre o hardware,

protegendo a capacidade de pinos, aproveitando todas entradas, além de agregar função ao hardware.

Figura 03 - Placa de expansão de rede Ethernet Shield

Fonte: Luthortronics (2016)

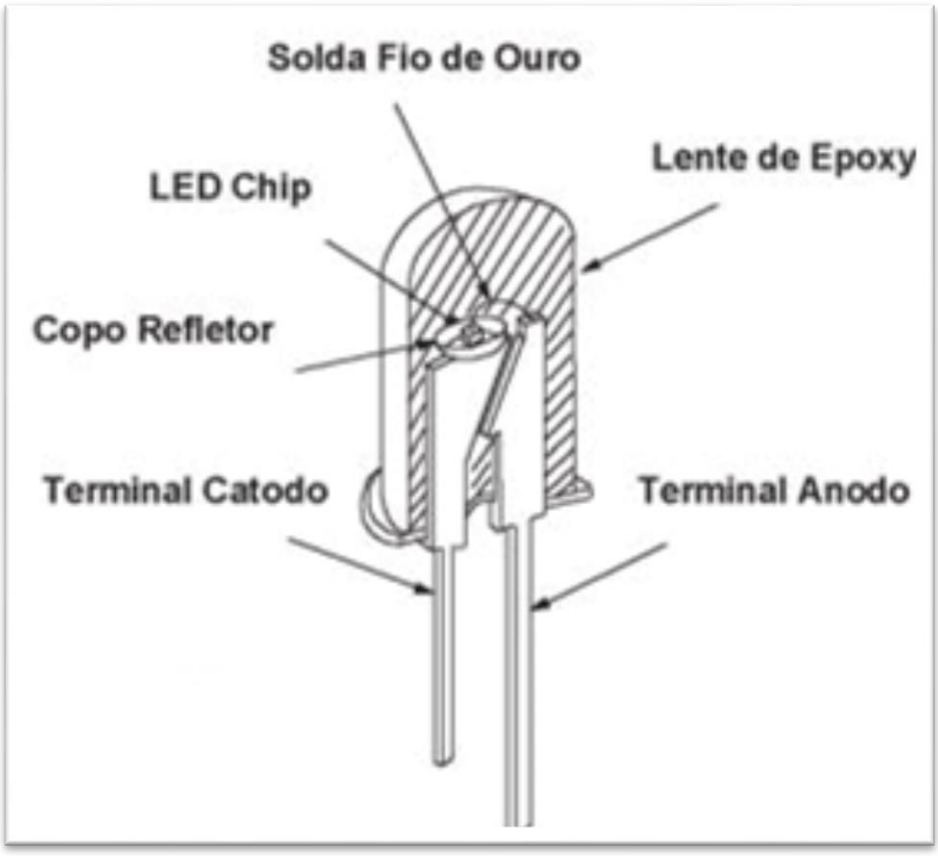
O Arduino dispõe de uma enorme quantidade de tipos de sensores e atuadores, por ser uma plataforma de código aberto, isso facilita o surgimento de inúmeros dispositivos. Serão demonstrados alguns utilizados nesse projeto.

O Sensor de Temperatura segundo Nakamura (1996 apud MEDEIROS, 2013) trata-se te um sensor de entrada externa capaz de converter a temperatura em um sinal elétrico proporcional. O Sensor de vibração é um dispositivo de entrada externa capaz de captar a vibração de uma turbina instalada em seu interior, e retornar a intensidade de vibração em dados (MEDEIROS, 2013). O Sensor de chuva é um dispositivo de entrada externa capaz de captar sinais de chuva quando molhado, fechando curto entre os condutores expostos em sua superfície, gerando então um sinal analógico, mas convertido para sinal digital antes de chegar ao

Arduino (ARDUINOLANDIA, 2016). O Sensor de presença captura dados externos, sendo capaz de detectar movimentos com sensor infravermelho através do calor emitido pelo objeto. Seu método está em sua sensibilidade de mudança de radiação infravermelha, se o objeto se mexer o movimento será captado. Esse modelo gera apenas dois sinais, detecção de movimento ou sem movimento (MARCHESAN, 2012).

Tratando agora de componentes eletrônicos não pertencentes originalmente ao Arduino, seguem os Leds que são semicondutores com propriedade de transformar energia elétrica em luz. A mudança de energia elétrica para luz é realizada na matéria, chamada de Estado Solido (*Solid State*) (SCOPACASA, 2008). Na Figura 04 podemos compreender a estrutura de um Led.

Figura 04 - ilustração dos componentes de um led



Fonte: Scopacasa, (2016)

E os resistores que são utilizados para criar uma resistência de corrente para ligar os Leds, são componentes eletrônicos capazes de criar uma resistência elétrica. Essa resistência é medida pela unidade ohms(Ω), criando uma oposição a

passagem de uma determinada corrente. Os mais comuns tem o corpo feito em carbono e o fio de nicromo (BRAGA, 2005).

No capítulo seguinte serão classificadas as etapas do projeto desenvolvido em duas partes, sistema elétrico e comunicação, que explicarão os métodos utilizados para demonstrar a automação residencial com Arduino.

4. PROTÓTIPO AUTOMAÇÃO RESIDENCIAL

Para demonstrar o funcionamento de um sistema de automação, a pesquisa utiliza um modelo para aplicação. A pesquisa é desenvolvida em uma maquete conforme mostra a Figura 05;

Figura 05 - Maquete com sistema de automação residencial

Fonte: Próprio autor (2016)

Para construção desse protótipo foram utilizados os componentes elétricos e materiais listados da tabela 01.

Tabela 01 - Lista de materiais utilizados para o desenvolvimento do projeto

Quant	Descrição
01	Sensor de temperatura
01	Sensor de Chuva
01	Sensor de Vibração
01	Sensor de Presença
01	Arduino Mega
01	Ethernet Shields
13	Resistores 31Ω
13	Leds 3,6V
-	Fios para conexão dos componentes
01	Maquete em MDF 3mm
01	Placa compensado 15mm - 40x60cm
01	Buzzer

Fonte: Próprio autor (2016)

Com todo material citado na Tabela 01, é possível criar o ambiente para o desenvolvimento desta pesquisa, e de forma geral esses itens compõem todo sistema elétrico para demonstrar de forma funcional um sistema de automação residencial, como pode ser acompanhado a seguir.

4.1. Sistema elétrico

Para alimentação do sistema o Arduino é capaz de fornecer energia suficiente a partir das entradas/saídas com uma corrente de 5V, os Leds que representam as lâmpadas devem ser alimentados com uma corrente de 3,4V, como o Arduino tem voltagem maior, utiliza-se resistores para controlar a corrente. E com a fórmula demonstrada na Figura 06, pode-se determinar que o resistor necessário para reduzir a tensão é de 31Ω. O resistores são ligados entre o terminal Anodo do led e o pino digital do Arduino.

Figura 06 - Formula para calcular resistor para um led

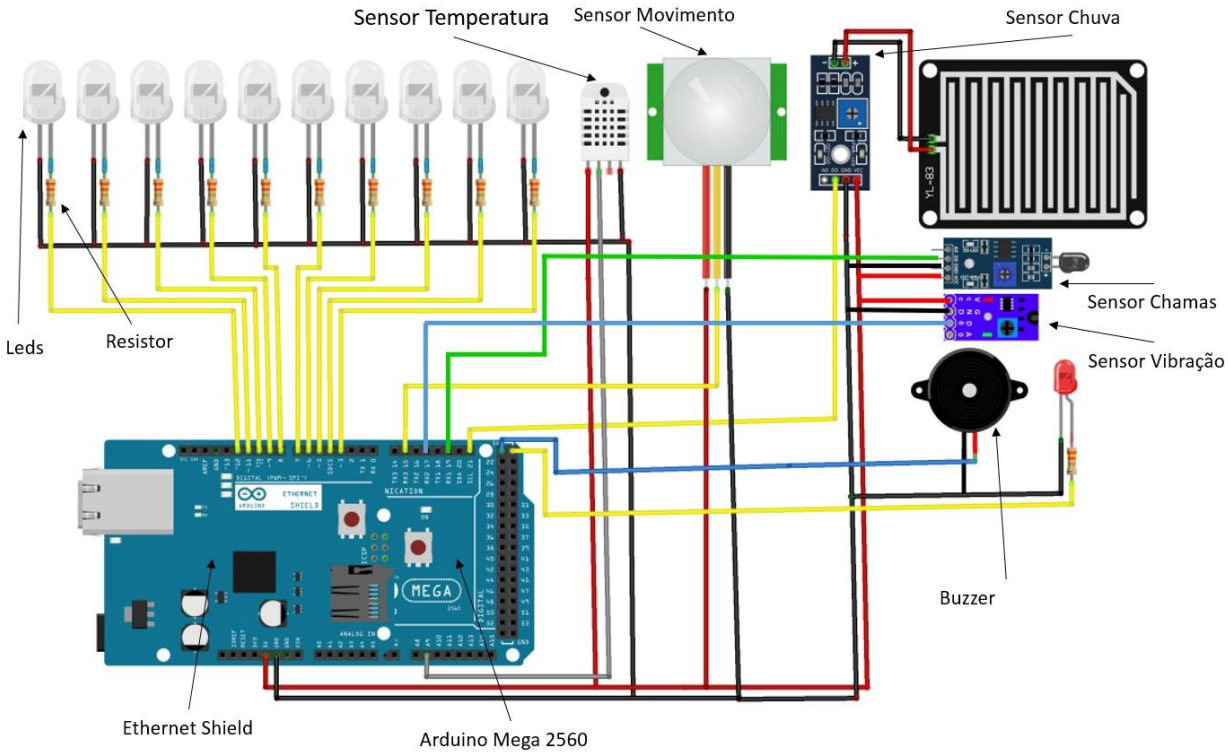
$$R_{LED} = \frac{V_{fonte} - V_{LED}}{\mathrm{I_{LED}}}$$

R_{LED} - Resistência em série com o LED
V_{fonte} - Tensão da fonte que alimenta o circuito
V_{LED} - Queda de tensão no LED
I_{LED} - Corrente máxima suportada pelo LED

Fonte: Eletrinuite (2016)

Os leds são conectados nos pinos digitais (entradas/saídas) do Arduino, configurados para enviar dados da placa para os leds resultando no acionamento ou desligamento de cada led individualmente. A conexão entre o Arduino e todos os componentes elétricos com leds, sensores e resistores é feita com fios de cobre revestido para evitar qualquer curto circuito. Os sensores tem alimentação compatível com a da placa descartando o uso de resistores. Confira o esquema elétrico na figura 07.

Figura 07 - Esquema elétrico da construção do projeto

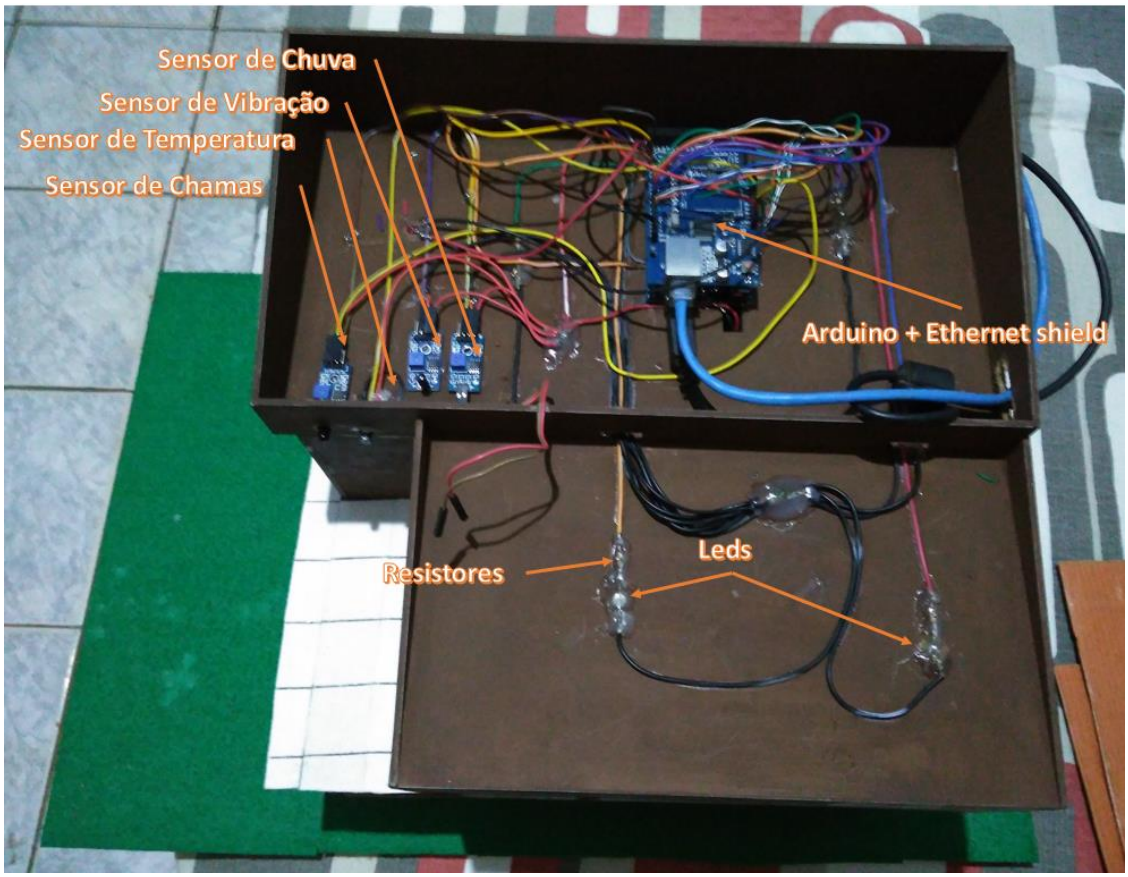


Fonte: Próprio autor (2016)

A Figura 08 exibe o sistema elétrico já desenvolvido, seguindo o esquema conforme a Figura 07. Toda elétrica foi centralizada na parte superior da maquete, abaixo do telhado, nela é possível ver os sensores de chuva, temperatura e chamas instalados na parte externa, já que a informação a serem coletadas por eles veem da parte exterior da maquete. Já os sensores de vibração

e de movimento ficam na parte interior da maquete, para capturar movimento de dentro do ambiente. Os leds, apenas de pouca visibilidade esta instalados no interior da maquete junto aos resistores que criam a resistência de corrente. O Arduino e o Ethernet Shield são o centro de todos os componentes ligados por fios de cobre revestido.

Figura 08 – Sistema elétrico instalado na maquete



Fonte: Próprio autor (2016)

Em termo básicos toda instalação elétrica resume-se em ligar os dispositivos a central de controle representada pela Arduino Mega2560, pois ele é o responsável para executar a ações enviadas pelo usuário e coletar dados do ambiente. O esquema elétrico é representado na Figura 06, nela é possível observar todos sensores e outros componente conectados ao Arduino através dos fios de cobre revestido.

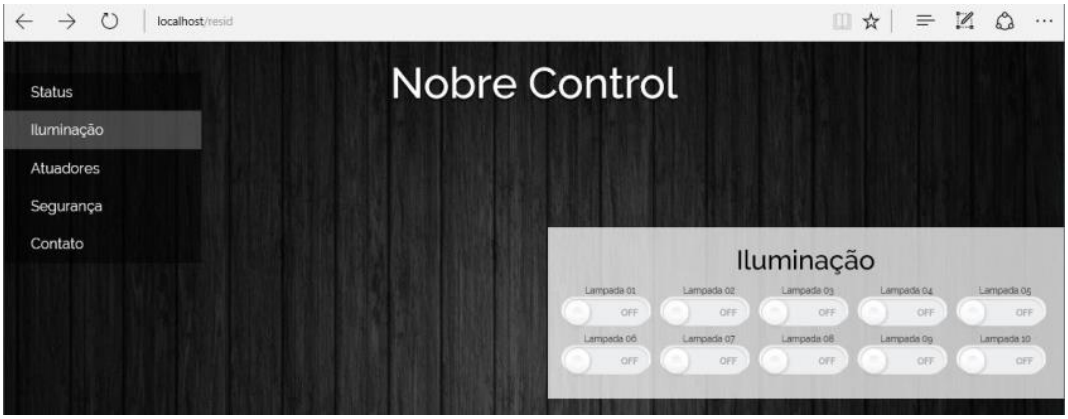
4.2. Comunicação

A interação do usuário com a central de controle é executada através de uma interface. O usuário seleciona a ação desejada e a interface é responsável

por traduzir isso para o Arduino, que por sua vez executa a ação. Um aplicativo WEB responsivo é a melhor proposta para fornecer acessibilidade das ações, podendo ser acessado a partir de um celular ou computador.

A Figura 09 demonstra a interface de interação do usuário com o sistema, desenvolvida com as linguagens de programação, HTML5, Java Script e PHP. Essa versão é exibida em ambiente desktop, nela o usuário interage com os leds através de botões representando um interruptor de energia.

Figura 09 - Aplicativo Nobre Control – versão Desktop



Fonte: Próprio autor (2016)

O aplicativo precisa de um hospedeiro para ser exibido, isso coloca a necessidade de um computador servidor entre o usuário e o Arduino, que hospeda também um banco de dados Mysql. Como é possível observar na Figura 10, o banco de dados criado com o nome “automação” tem uma única tabela, por nome “sensores”, composta pelas colunas temperatura, chuva, fogo, movimento e vibração, para armazenamento dos dados colhidos pelo Arduino dos sensores, através de rotinas criadas em PHP e executadas pelo Arduino a cada segundo, como demonstra nas linha 288 a 306 no código em anexo.

Figura 10 - Banco de dados na tabela sensores

Fonte: Próprio autor (2016)

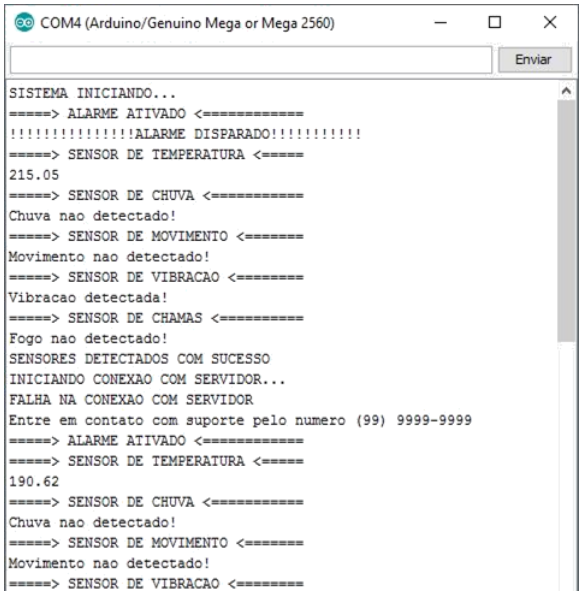
O servidor precisa estar conectado ao Arduino continuamente para não perder informações do ambiente na ausência do usuário. O Arduino Mega2560 não possui conexão com rede de computadores, e por essa razão é necessário o uso do Ethernet Shield para fornecer conexão com o roteador, e permitir comunicação via HTTP com o servidor.

5. COLETA DE DADOS

Após o desenvolvimento das rotinas do bando de dados, e programação do Arduino para receber comandos e executar ações, alguns testes foram executados, como solicitações repentinas e constantes ao Arduino, que se portou de forma positiva, podendo ser conferido na Figura 09, que apresenta o monitor serial utilizado para fazer depuração dos processos em execução na placa, que em caso de falhas, o administrador possa conferir em qual etapa está ocorrendo o erro. Essa depuração é elaborada através de comentários no código separando cada etapa como podemos conferir nas linhas 60, 115 ou 118 do código em anexo.

Na aplicação uma solicitação de acionamento foi enviada para cada led, afim de constatar falhas no envio de comandos, e o resultado pode ser constatado na Figura 11, onde a maquete se apresenta com todos os leds acessos. O sistema não falhou em nenhum dos testes de solicitações de ação.

Figura 11 - Teste com sequência de solicitações



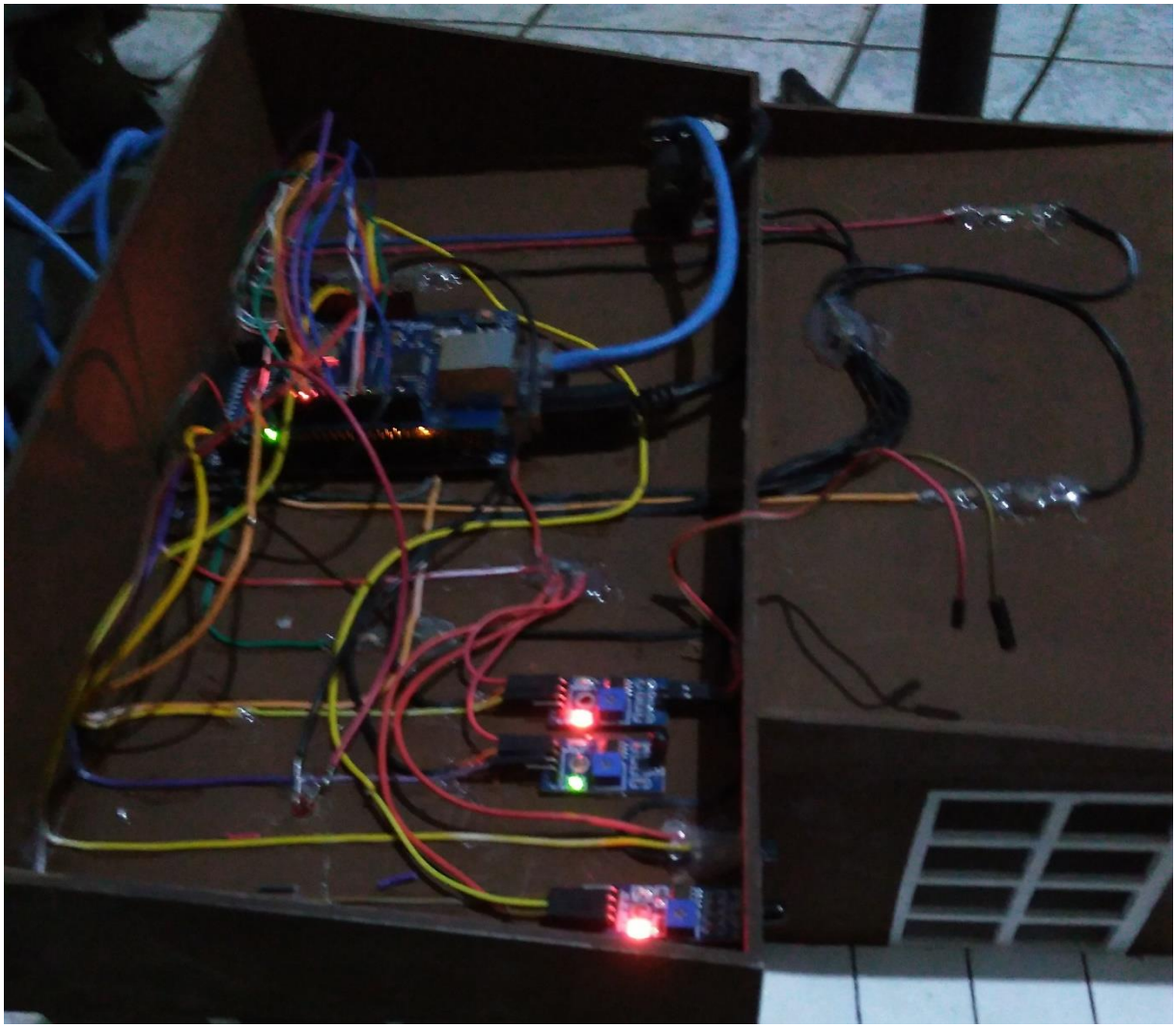
Fonte: Próprio autor (2016)

Figura 12 - Maquete com led acesso

Fonte: Próprio autor (2016)

O próximo passo foi testar a informação colhida dos sensores e os parâmetros são os seguintes: Sensor de temperatura: Uma vez acionado deve retornar a temperatura ambiente e salvar na tabela “sensores” coluna “temperatura”. O acionamento é feito através de uma rotina configurada no Arduino. Foram feitos leituras do sensor por três dias, o mesmo retornou corretamente a temperatura. O sensor de chuva que testado com uma quantidade de agua, alterou seu estado de “sem chuva” para “com chuva”. O sensor de vibração que em outros países com maior índice de terremotos pode ter grande necessidade foi inserido a testes de movimentação do sensor, alterando o status de “sem vibração” para “vibração detectada”. O sensor de presença foi testado com movimentos aleatórios em seu alcance de captação, e retornando a informação de “movimento detectado”.

Figura 13 – Sistema elétrico em funcionamento



Fonte: Próprio autor (2016)

O funcionamento é conferido através de um led em cada sensor conforma na Figura 13. O status de todos os sensores testados são atualizados a cada 1 segundo, no banco de dados na tabela “sensores”, a conexão com o banco pode ser observada na Figura 11. A disponibilidade e integridade dos dados foi testada com consultas diretas ao banco de dados a cada segundo, e mesmo com algum tempo de funcionamento o sistema não demonstrou perca de informação ou falha na atualização de dados, como é demonstrado na Figura 08.

A interface da aplicação é composta por um menu suspenso facilitando a navegação para dispositivos moveis por entre a funções do sistema, contendo botões personalizado para acionamento de cada led como demonstra a Figura 11, além de permitir acompanhamento em tempo real do que está acontecendo na casa. O acesso a aplicação é realizado pela rede local, e de acordo o ambiente configurado pode ser acessado sem fio para comodidade de seus usuários que interagem com os botões enviando determinada ação para o Arduino executar.

Figura 11 - Aplicativo Nobre Control – Versão mobile

Fonte: Próprio autor (2016)

O sistema se portou de modo esperado, tanto ao executar ações solicitadas pelo usuário como na coleta de informações dos sensores e armazenamento no

banco de dados, mostrando-se um sistema estável para um projeto de automação residencial.

6. CONCLUSÃO

O desenvolvimento do presente estudo, permitiu compreender como um sistema de automação aplicado em residências funciona, demonstrando através de um ambiente residencial simulado em uma maquete controlada por um Arduino, além disso possibilitou avaliar a importância dessa tendência já tão presente em indústrias e prédios, e está conquistando espaço na categoria residencial.

O estudo ainda fala das vantagens da automação residencial para o usuário, tratando da realidade atual, essa em que pessoas trocam informação de forma instantânea, gerando comodidade, segurança e economia. O Arduino destaca-se pela sua flexibilidade para sanar necessidades bem peculiares de uma residência, ao contrário de uma plataforma pré-programada com arquitetura fechada impedindo uma compatibilidade com equipamentos fabricados por terceiros.

Contudo que foi estudo, e aplicado, o objetivo de construir um projeto que demonstrasse um sistema de automação residencial com Arduino foi alcançado ao implementar um conjunto de conhecimentos intermediário em elétrica e programação.

A instalação de Leds simulando lâmpadas de uma casa reforçaram o conceito residencial para a demonstração, e ao clicar no botão que envia ordem para o Arduino ligar uma determinada “lâmpada” é uma característica comum de um sistema de automação. O sensor de presença ao captar movimentos em conjunto com um buzzer, simularam um controle de alarme, dado ênfase ao que o sistema de automação residencial pode fornecer para segurança de uma casa.

Existem ações, que podem somar ainda mais para controle de uma casa, não necessitando sempre de intervenção humana. Dados coletados do ambiente podem ser tratados e servir para melhor o consumo de energia e agua. É o que pode ser feito com um sensor de luminosidade, que configurado conforme desejo do usuário pode acender uma lâmpada fora de casa ao anoitecer, ou até mesmo um simples sensor de temperatura pode abrir uma janela quando a temperatura interior da casa estiver muito acima da temperatura externa.

Ao fim do desenvolvimento do projeto, o mesmo foi inserido a um conjunto de testes, portando-se de forma positiva ao esperado, podendo ser utilizado como modelo em um projeto real de automação residencial, e até mesmo ser reestruturado para aplicação em uma casa.

7. REFERENCIAS

ALVES, José Augusto; MOTA, José. **Casas inteligentes**. Centro Atlântico, 2003.

ARDUINOLANDIA. **Descrição sensor de chuva.** Disponível em: <http://www.arduinolandia.com.br/sensor-de-chuva>. Acesso em: 20 outubro. 2016.

BANZI, Massimo. **Primeiros passos com o Arduino**. São Paulo: Novatec, p. p1, 2011.

BOLZANI, Caio Augustus Morais. **Residências inteligentes**. Editora Livraria da Física, 2004.

BRAGA, Newton C. Eletrônica básica para mecatrônica. Saber, 2005.

DICIONARIO. **DEFINIÇÃO de automação.** Disponível em: <http://www.dicio.com.br/automacao>. Acesso em: 21 março. 2015.

ELETRONITE. **Formula cálculo de resistor.** Disponível em: <http://www.eletronite.com.br/ferramentas/calculadora-de-leds.html>. Acesso em: 20 outubro. 2016.

FRANCISCO, Lucas; TREVISANI, Kleber Manrique. HMS: **Uma Arquitetura para automa-ção residencial aberta independente de tecnologia de rede**. In:Colloquium Exactarum. 2013. p. 43-56.

LIMA, Gustavo Fernandes. **Controle de temperatura de um sistema de baixo custo utilizando a placa Arduino**. In: IX Congresso de Iniciação Científica do IFRN. 2013.

LUTHORTRONICS. **Figura Ethernet Shield.** Disponível em: < http://luthortronics.com.br/wp-content/uploads/2015/10/qnfgjz1339666028684.jpg >. Acesso em: 20 outubro. 2016.

MCROBERTS, Michael. **Arduino básico**. Novatec Editora, 2011.

MEDEIROS, Josenei G.; GUIMARAES, Lamartine F.; PLACCO, Guilherme M. **Control system to a Rankine cycle with a Tesla turbine using Arduino**. 2013.

MARCHESAN, Marcelo. Sistema de monitoramento residencial utilizando a plataforma arduino. Santa Maria, 2012.

SCOPACASA, Vicente A. Introdução à Tecnologia de LED. Revista LA_PRO, São Paulo, ed, v. 1, p. 5-10, 2008.

ANEXO A – CODIGO DO ARDUINO

CODIGO DO ARDUINO

```
#include <SPI.h>
#include <Ethernet.h>



byte mac[] = {0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED};
byte ip[] = {192,168,1,101};
byte gateway[] = {192,168,1,1};
byte subnet[] = {255,255,255,0};
byte server[] = {192,168,1,100};
EthernetClient client;

//LED
EthernetServer servidor(80);
String readString;
//Pino digital onde será ligado e desligado o led no Arduino.
int led1 = 12;
int led2 = 3;
int led3 = 4;
int led4 = 5;
int led5 = 6;
int led6 = 7;
int led7 = 8;
int led8 = 9;
int led9 = 10;
int led10 = 11;

//ALARME
int pinBuzzer = 22;
int pinLed = 24;
int valorSensorPIR = 0;

//Sensor temperatura
int SENSOR = 0;
float S = 0;
float T = 0;
String stg;

//Sensor Chuva
int pino_dChuva = 21; //Pino ligado ao D0 do sensor
int val_dChuva = 0; //Armazena o valor lido do pino digital

//Sensor Fogo
int pino_dFogo = 19; //Pino ligado ao D0 do sensor
int valor_dFogo = 0; //Armazena o valor lido do pino digital

//Sensor Vibração
int porta_dVibra = 17; //Pino ligado ao D0 do sensor
int val_dVibra = 0; //Armazena o valor lido do pino digital

//Sensor Movimento
int pino_dmov = 15; //Pino ligado ao sensor PIR
int val_dmov; //Variavel para guardar valor do sensor

void setup() {
    Ethernet.begin(mac, ip);
```

```
57        Serial.begin(9600);
58        servidor.begin(); //  Inicia o servidor
59
60       Serial.println("SISTEMA INICIANDO...");
61
62        //ALARME
63        pinMode(pinBuzzer,OUTPUT);
64        pinMode(pinLed,OUTPUT);
65
66        //sensor chuva
67        pinMode(pino_dChuva, INPUT);
68
69        //sensor fogo
70        pinMode(pino_dFogo, INPUT);
71
72        //sensor vibração
73        pinMode(porta_dVibra, INPUT);
74
75
76        //sensor movimento
77        pinMode(pino_dmov, INPUT);
78
79        //LED
80        pinMode(led1, OUTPUT);
81        pinMode(led2, OUTPUT);
82        pinMode(led3, OUTPUT);
83        pinMode(led4, OUTPUT);
84        pinMode(led5, OUTPUT);
85        pinMode(led6, OUTPUT);
86        pinMode(led7, OUTPUT);
87        pinMode(led8, OUTPUT);
88        pinMode(led9, OUTPUT);
89        pinMode(led10, OUTPUT);
90
91    }
92
93    void loop() {
94
95            Serial.println("=====> ALARME ATIVADO <============");
96            valorSensorPIR = digitalRead(pino_dmov);
97             //Verificando se ocorreu detecção de movimentos
98            if (valorSensorPIR == 1) {
99              ligarAlarme();
100           } else {
101             desligarAlarme();
102           }
103           Serial.println("=====> SENSOR DE TEMPERATURA <=====");
104           sensorTemp();
105           delay(100);
106           Serial.println("=====> SENSOR DE CHUVA <===========");
107           sensorChuva();
108           delay(100);
109           Serial.println("=====> SENSOR DE MOVIMENTO <=======");
110           sensorMove();
111           delay(100);
112           Serial.println("=====> SENSOR DE VIBRACAO <========");
113           sensorVibra();
114           delay(100);
115           Serial.println("=====> SENSOR DE CHAMAS <==========");
116           sensorFogo();
117           delay(100);
```

```
118             Serial.println("SENSORES DETECTADOS COM SUCESSO");
119             salvaDados();
120             delay(100);
121
122
123         delay(100);
124
125         //LED
126         EthernetClient client = servidor.available();
127         if (client) {
128             while (client.connected()) {
129                 if (client.available()) {
130                     char c = client.read();
131
132                     if (readString.length() < 100) {
133                         readString += c;
134                     }
135
136                     if (c == '\n') {
137
138                         delay(1);
139                         client.stop();
140
141                         ///////LED 01
142                         if (readString.indexOf("?ligar1") > 0) {
143                             digitalWrite(led1, HIGH); //Liga
144                         } else {
145                             if (readString.indexOf("?desligar1") > 0) {
146                                 digitalWrite(led1, LOW); //Desliga
147                             }
148                         }
149
150                         ///////LED 02
151                         if (readString.indexOf("?ligar2") > 0) {
152                             digitalWrite(led2, HIGH); //Liga
153                         } else {
154                             if (readString.indexOf("?desligar2") > 0) {
155                                 digitalWrite(led2, LOW); //Desliga
156                             }
157                         }
158
159                         ///////LED 03
160                         if (readString.indexOf("?ligar3") > 0) {
161                             digitalWrite(led3, HIGH); //Liga
162                         } else {
163                             if (readString.indexOf("?desligar3") > 0) {
164                                 digitalWrite(led3, LOW); //Desliga
165                             }
166                         }
167
168                         ///////LED 04
169                         if (readString.indexOf("?ligar4") > 0) {
170                             digitalWrite(led4, HIGH); //Liga
171                         } else {
172                             if (readString.indexOf("?desligar4") > 0) {
173                                 digitalWrite(led4, LOW); //Desliga
174                             }
175                         }
176
177                         ///////LED 05
178                         if (readString.indexOf("?ligar5") > 0) {
```

```
                        digitalWrite(led5, HIGH); //Liga
                    } else {
                        if (readString.indexOf("?desligar5") > 0) {
                            digitalWrite(led5, LOW); //Desliga
                        }
                    }

                    ///////LED 06
                    if (readString.indexOf("?ligar6") > 0) {
                        digitalWrite(led6, HIGH); //Liga
                    } else {
                        if (readString.indexOf("?desligar6") > 0) {
                            digitalWrite(led6, LOW); //Desliga
                        }
                    }

                    ///////LED 07
                    if (readString.indexOf("?ligar7") > 0) {
                        digitalWrite(led7, HIGH); //Liga
                    } else {
                        if (readString.indexOf("?desligar7") > 0) {
                            digitalWrite(led7, LOW); //Desliga
                        }
                    }

                    ///////LED 08
                    if (readString.indexOf("?ligar8") > 0) {
                        digitalWrite(led8, HIGH); //Liga
                    } else {
                        if (readString.indexOf("?desligar8") > 0) {
                            digitalWrite(led8, LOW); //Desliga
                        }
                    }

                    ///////LED 09
                    if (readString.indexOf("?ligar9") > 0) {
                        digitalWrite(led9, HIGH); //Liga
                    } else {
                        if (readString.indexOf("?desligar9") > 0) {
                            digitalWrite(led9, LOW); //Desliga
                        }
                    }

                    ///////LED 10
                    if (readString.indexOf("?ligar10") > 0) {
                        digitalWrite(led10, HIGH); //Liga
                    } else {
                        if (readString.indexOf("?desligar10") > 0) {
                            digitalWrite(led10, LOW); //Desliga
                        }
                    }

                    readString = "";
                }
            }
        }
    }

}

void sensorTemp() {
```

```
240       //Lendo dados do Sensor de temperatura
241       S = analogRead(SENSOR); //Lê porta analógica e armazena em S
242       T = (S * 500) / 1023; //Conversão do sinal lido em Temperatura
243       stg = String(T);
244       Serial.println(stg);
245   }
246
247   void sensorChuva() {
248       //Le e arnazena o valor do pino digital
249       val_dChuva = digitalRead(pino_dChuva);
250       //Envia as informacoes para o serial monitor
251       if (val_dChuva != 1){
252         Serial.println("Chuva detectado!");
253       }else{
254           Serial.println("Chuva nao detectado!");
255         }
256   }
257
258   void sensorFogo() {
259       int valor_dFogo = digitalRead(pino_dFogo);
260       if (valor_dFogo != 1) {
261           Serial.println("Fogo detectado!");
262       }else{
263           Serial.println("Fogo nao detectado!");
264         }
265       delay(500);
266   }
267
268   void sensorVibra() {
269       val_dVibra = digitalRead(porta_dVibra);
270       if (val_dVibra != 1) {
271           Serial.println("Vibracao detectada!");
272       }else{
273           Serial.println("Vibracao nao detectada!");
274         }
275
276       delay(100);
277   }
278
279   void sensorMove() {
280       val_dmov = digitalRead(pino_dmov);
281       if (val_dmov == LOW){
282         Serial.println("Movimento nao detectado!");
283       }else {
284           Serial.println("Movimento detectado!");
285        }
286   }
287
288   void salvaDados() {
289       Serial.println("INICIANDO CONEXAO COM SERVIDOR...");
290
291       if (client.connect(server, 80)) {
292
293           Serial.println(stg);
294           client.println("GET
295   http://192.168.1.100/autom/sensor_temp.php?temp="+stg+"&chuva="+val_dChuva+"&fog
296   o="+valor_dFogo+"&vibra="+val_dVibra);
```

```
297
298           client.stop();
299           Serial.println("CONEXAO REALIZADA COM SUCESSO!");
300
301       } else {
302           Serial.println("FALHA NA CONEXAO COM SERVIDOR");
303           Serial.println("Entre em contato com suporte pelo numero (99) 9999-
304   9999");
305           client.stop();
306       }
307   }
308
309   void ligarAlarme() {
310     Serial.println("!!!!!!!!!!!!!!!ALARME DISPARADO!!!!!!!!!!!");
311     digitalWrite(pinLed, HIGH);
312     tone(pinBuzzer,1500);//Ligando o buzzer com uma frequencia de 1500 hz.
313     delay(5000); //tempo que o led fica acesso e o buzzer toca
314     desligarAlarme();
315   }
316   void desligarAlarme() {
317     digitalWrite(pinLed, LOW);
318     noTone(pinBuzzer)
```