

**FACULDADES DOCTUM DE CARATINGA
CURSO SUPERIOR DE ENGENHARIA ELÉTRICA**

**AUTOMAÇÃO DE CHUVEIRO RESIDENCIAL ABASTECIDO POR SISTEMA DE
AQUECIMENTO SOLAR**

WÊMERTON CLÁUDIO SILVA

Trabalho de Conclusão de Curso

Caratinga/MG

2017

WÊMERSON CLÁUDIO SILVA

**AUTOMAÇÃO DE CHUVEIRO RESIDENCIAL ABASTECIDO POR SISTEMA DE
AQUECIMENTO SOLAR**

Trabalho de Conclusão de Curso apresentado à Banca Examinadora do Curso Superior de Engenharia Elétrica das Faculdades DOCTUM de Caratinga como requisito parcial para obtenção do Grau de Bacharel em Engenharia Elétrica.
Professor Orientador: José Eugênio de Oliveira e Silva.

Caratinga/MG

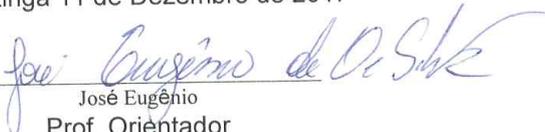
2017

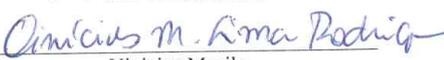
TERMO DE APROVAÇÃO

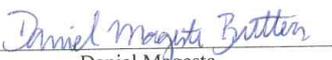
O Trabalho de Conclusão de Curso intitulado: AUTOMAÇÃO DE CHUVEIRO RESIDENCIAL ABASTECIDO POR SISTEMA DE AQUECIMENTO SOLAR, elaborado pelo(s) aluno(s) WEMERSON CLAUDIO SILVA foi aprovado por todos os membros da Banca Examinadora e aceito pelo curso de ENGENHARIA ELÉTRICA das FACULDADES DOCTUM CARATINGA, como requisito parcial da obtenção do título de

BACHAREL EM ENGENHARIA ELÉTRICA.

Caratinga 11 de Dezembro de 2017


José Eugênio
Prof. Orientador


Vinicius Murilo
Prof. Avaliador 1


Daniel Mageste
Prof. Examinador 2

A Deus, por tudo que me concedeu até aqui. Por me permitir transpor os obstáculos e colher os frutos do que semeei ao longo do caminho!

"Há tempo de plantar e tempo de colher". Eclesiastes, 3.2.

O tempo da colheita chegou.

AGRADECIMENTOS

A Deus, por me conceder saúde e força para chegar até aqui;

A toda minha família, minha esposa Eliana e meus filhos Luiz Augusto e Letícia, pelo amor, incentivo e compreensão;

Aos mestres, coordenadores e demais funcionários dessa instituição, pela oportunidade, dedicação e conhecimentos compartilhados;

Ao meu orientador José Eugênio de Oliveira e Silva, pela paciência, suporte e disponibilidade constantes na condução das atividades;

Aos colegas Gilmar Correa Moraes, Klaiver Vinicius da Silva Costa e professor Elias de Souza Gonçalves, pela presteza e suporte técnico na realização desse trabalho;

Aos meus colegas de classe que estiveram presentes em todos os momentos compartilhando angústias, alegrias, dificuldades, perdas e vitórias;

E a todos que direta ou indiretamente contribuíram para a minha formação e construção desse trabalho;

Meus sinceros agradecimentos!

*“Não existe vitória sem luta, não há recompensa sem esforço e para aqueles que perseveram
Deus tem reservado lindas surpresas.”*
(Yla Fernandes)

SILVA, Wêmersom Cláudio. **AUTOMAÇÃO DE CHUVEIRO RESIDENCIAL ABASTECIDO POR SISTEMA DE AQUECIMENTO SOLAR**. Caratinga, 2017. Trabalho de Conclusão de Curso Superior de Engenharia Elétrica. Faculdades DOCTUM de Caratinga, Caratinga, 2017.

RESUMO

Este trabalho aborda um estudo de caso de uma residência que utiliza um sistema de aquecimento solar. O chuveiro dessa residência está instalado a uma distância considerável do reservatório de água quente. Devido a essa característica o volume de água armazenado entre o chuveiro e o reservatório é muito grande. Para que seja possível utilizar a água quente do sistema solar é necessário descartar a água fria da tubulação para que a água aquecida chegue ao chuveiro. O intuito desse trabalho é eliminar o desperdício de água por meio da automação do chuveiro mediante a construção de um controle eletrônico, com o objetivo de aquecer a água fria presente na tubulação com uso da energia elétrica, evitando assim seu descarte. A pesquisa realizada é do tipo experimental e envolve a construção de um protótipo com o uso de um microcontrolador. Após a construção e realização dos testes pode-se concluir que o objetivo foi alcançado tendo como principal ganho a economia de água, além de outros benefícios agregados com a implantação do sistema.

Palavras-chave: Automação. Controle. Microcontrolador. Protótipo.

SILVA, Wêmersom Cláudio. **AUTOMAÇÃO DE CHUVEIRO RESIDENCIAL ABASTECIDO POR SISTEMA DE AQUECIMENTO SOLAR.** Caratinga, 2017. Trabalho de Conclusão de Curso Superior de Engenharia Elétrica. Faculdades DOCTUM de Caratinga, Caratinga, 2017.

ABSTRACT

This paper addresses a case study of a residence using a solar heating system. The shower in this residence is installed a considerable distance from the hot water tank. Due to this feature the volume of water stored between the shower and the reservoir is very large. In order to be able to use hot water from the solar system it is necessary to discard the cold water from the tubing so that the heated water reaches the shower. The purpose of this work is to eliminate the waste of water by means of the automation of the shower by means of the construction of an electronic control, with the objective of heating the cold water present in the pipe with the use of the electric energy, thus avoiding its disposal. The research carried out is of the experimental type and involves the construction of a prototype with the use of a microcontroller. After the construction and realization of the tests, it can be concluded that the objective was achieved with the main gain of water savings, as well as other added benefits with the implementation of the system.

Key-words: Automation. Control. Microcontroller. Prototype.

LISTA DE ILUSTRAÇÕES

Figura 1	– Sistema de malha aberta	18
Figura 2	– Sistema de malha fechada	19
Figura 3	– Microcontroladores da Atmel e Microchip	20
Figura 4	– Microcontrolador ATmega328	20
Figura 5	– Placa de Prototipagem Arduino UNO	22
Figura 6	– Shields para Arduino	22
Figura 7	– Display LCD 16x2	26
Figura 8	– Sensor de temperatura digital DS18B20	28
Figura 9	– Eletroválvula para Água	29
Figura 10	– IDE do Arduino	33
Figura 11	– Fluxograma de Operação	35
Figura 12	– Esquema de Montagem do Sistema	37
Figura 13	– Circuito de Controle e Acionamento do Protótipo	37
Figura 14	– Interface do Sistema	38
Figura 15	– Instalação do Protótipo	39
Figura 16	– Protótipo Etiquetado com Instruções de Uso	40
Figura 17	– Resposta do Sistema	41

LISTA DE TABELAS

Tabela 1	– Especificações da placa Arduino UNO R3	23
Tabela 2	– Especificações do Display LCD16x2	26
Tabela 3	– Tipos de Sensores	27
Tabela 4	– Especificações do Sensor de Temperatura DS18B20	28
Tabela 5	– Características e Especificações da Eletroválvula	30
Tabela 6	– Relação dos Componentes	32
Tabela 7	– Medições da Temperatura Máxima da Água Aquecida pelo Sistema Solar .	32

LISTA DE ABREVIATURAS E SIGLAS

A/D	Analógico/Digital
CLP	Controlador Lógico Programável
CPU	<i>Central Processing Unit</i>
DC	<i>Direct Current</i>
EEPROM	<i>Electrically Erasable Programmable Read-Only Memory</i>
EV1	Eletroválvula Água Quente
EV2	Eletroválvula Água Fria
GND	<i>Ground</i>
I/O	<i>Input/Output</i>
ID	Identidade
IDE	<i>Integrated Development Environment</i>
IoT	<i>Internet of Things</i>
LCD	<i>Liquid Cristal Display</i>
LED	<i>Ligth Emitting Diode</i>
<i>Open-source</i>	Código Aberto
PC	<i>Personal Computer</i>
PLC	<i>Power Line Carrier</i>
PWM	<i>Pulse Width Modulation</i>
RAM	<i>Random Access Memory</i>
RE	Resistência Elétrica
RISC	<i>Reduced Instruction Set Computer</i>
RS	<i>Register Select</i>
RW	<i>Read/Write</i>
SRAM	<i>Static Random Access Memory</i>

ST	Sensor de Temperatura
USB	<i>Universal Serial Bus</i>
Vca	Tensão em Corrente Alternada
VDD	Tensão Positiva de Alimentação
VEE/VO	Ajuste de Contraste
VSS	Tensão Negativa de Alimentação ou Terra

LISTA DE SÍMBOLOS

A	Ampère
cm	Centímetro
g	Gramma
°C	Grau Celsius
l/min	Litro/minuto
MHz	Megahertz
m^3	Metro cúbico
μs	Microsegundo
mA	Miliampère
mm	Milímetro
ms	Milissegundo
mV	Milivolt
kB	Quilobyte
kgf/cm^2	Quilograma-força/centímetro quadrado
V	Volt

SUMÁRIO

1 INTRODUÇÃO	15
2 REFERENCIAL TEÓRICO	17
2.1 AUTOMAÇÃO	17
2.2 SISTEMAS DE CONTROLE	17
2.2.1 SINAIS DE ENTRADA E SAÍDA	18
2.2.2 SISTEMA DE CONTROLE EM MALHA ABERTA	18
2.2.3 SISTEMA DE CONTROLE EM MALHA FECHADA	19
2.3 MICROCONTROLADOR	19
2.4 ARDUINO	21
2.4.1 LINGUAGEM DE PROGRAMAÇÃO C	23
2.4.1.1 Funções-Base	23
2.4.1.2 Entrada e Saída de Dados	23
2.4.1.3 Tempo	24
2.4.1.4 Bibliotecas	24
2.4.1.5 Variáveis	24
2.4.1.6 Classes de Variáveis	24
2.4.1.7 Tipos de Dados e Modificadores	25
2.4.1.8 Estruturas de Controle de Fluxo	25
2.5 DISPLAY LCD	26
2.6 SENSORES	26
2.6.1 Sensor de Temperatura	27
2.7 ELETROVÁLVULAS	28
3 PROCEDIMENTOS METODOLÓGICOS	31
3.1 ESTUDO DE CASO	31
3.2 ANÁLISE DA VIABILIDADE	31
3.3 COLETA DE DADOS	32
3.4 PROGRAMAÇÃO DO SISTEMA	33
3.5 FUNÇÕES IMPLEMENTADAS NO SISTEMA	34
3.5.1 FLUXOGRAMA	34
3.6 FUNCIONAMENTO DO SISTEMA	35
3.6.1 INTERFACE DO SISTEMA	38
3.7 INSTALAÇÃO DO PROTÓTIPO	38
4 RESULTADOS E DISCUSSÕES	40
4.1 PROBLEMAS/DIFICULDADES ENCONTRADAS	41
4.2 SUGESTÕES PARA TRABALHOS FUTUROS	42

5 CONCLUSÃO	43
6 REFERÊNCIAS BIBLIOGRÁFICAS	44
APÊNDICE A PROGRAMAÇÃO DO SISTEMA	46

1 INTRODUÇÃO

A escassez cada vez maior dos recursos hídricos primordiais à sobrevivência da componente principal de nossa matriz energética, aliada ao custo mais elevado da tarifa de energia, tem nos impulsionado cada vez mais a utilizar fontes alternativas de energia de forma renovável e principalmente reduzir os gastos com o consumo, que está na lista das principais despesas no orçamento familiar.

Uma alternativa muito utilizada atualmente é o aquecimento da água através da energia solar. O princípio de funcionamento de um aquecedor solar é o sistema de termossifão, que consiste na circulação natural da água devido a diferença de densidade entre a água fria e água quente. A primeira, mais densa, pressiona a segunda fazendo com que essa saia dos coletores para o reservatório, mantendo um ciclo constante (SOLETROL, 2017).

O aquecedor solar tem a grande vantagem de utilizar uma fonte de energia limpa e renovável e seu uso numa residência para aquecer a água proporciona uma queda considerável no consumo de energia elétrica com a redução ou eliminação do uso do chuveiro elétrico, item responsável pela maior fatia de consumo de energia numa residência (INVESPAK, 2017). Porém a utilização do aquecedor solar ocasiona também algumas desvantagens como a sua limitação em períodos sem incidência de raios solares e principalmente o aumento no consumo de água, uma vez que ao ser aberto o registro da água quente torna-se necessário o descarte da água fria que fica armazenada na tubulação para que a água quente chegue ao chuveiro e possibilite a regulagem da temperatura ideal para o banho. Segundo Dias (2009) numa família de quatro pessoas que utiliza chuveiro abastecido com um sistema de aquecimento solar, o descarte da água fria no início de cada banho representa um desperdício de 600 litros a cada mês.

Considerando as desvantagens citadas acima, esse trabalho propõe um estudo de caso para eliminar o desperdício de água em um sistema de aquecimento solar instalado em uma residência em que o reservatório térmico instalado no terraço de um edifício de 4 andares fica a uma distância considerável do ponto de uso, localizado no primeiro andar.

O objetivo principal é propor a automatização do uso da água do chuveiro utilizando uma segunda fonte de energia, a elétrica, para aquecer a água fria atualmente descartada no início de cada banho. Essa alternativa seria utilizada também para compensar a limitação do sistema solar em dias nublados, onde a água não atinge a temperatura ideal para o banho.

Diante do exposto a proposta desse trabalho é a construção de um controle eletrônico para o chuveiro. Para tanto serão seguidas as etapas de análise de viabilidade técnica, escolha e aquisição dos materiais a serem utilizados, definição das funções que farão parte do controle eletrônico e da linguagem de programação a ser utilizada e finalmente a construção e instalação do controle eletrônico no local escolhido para estudo de caso.

O trabalho está estruturado em cinco capítulos, sendo esta introdução o primeiro.

No segundo capítulo serão abordados alguns conceitos relacionados ao tema trabalhado, sendo apresentado ao longo do capítulo todo o referencial teórico que sustenta o tema em questão e o conteúdo que embasa as componentes necessárias para construção do protótipo.

No terceiro capítulo será abordado o estudo de caso, os métodos e materiais empregados na confecção do controle eletrônico, funções requeridas, instalação e testes.

No quarto capítulo serão apresentados os resultados obtidos e sugestões para trabalhos futuros e por fim o quinto capítulo trará as considerações finais.

2 REFERENCIAL TEÓRICO

2.1 AUTOMAÇÃO

Podemos definir automação como sendo um sistema de equipamentos eletrônicos e/ou mecanismos que controlam seu próprio funcionamento, com pouca ou nenhuma ação humana. Com ela é possível realizar diversas ações com máquinas e equipamentos a partir de comandos automáticos, com a capacidade de se auto regularem (BORGES et al., 2017).

Um sistema de automação é composto de cinco elementos: acionamento, sensoriamento, controle, comparador ou elemento de decisão e programas (*softwares*), podendo ser aplicado nas mais diversas áreas como industrial, bancária, comercial, agrícola, transportes e residencial.

A automação surgiu a partir da segunda metade do século XVIII, na Inglaterra, com a transformação dos sistemas de produção agrário e artesanal em industrial (BORGES et al., 2017).

A automação residencial nasceu a partir da industrial, com o surgimento dos CLPs (Controladores Lógicos Programáveis) na década de 60, como consequência do avanço da microeletrônica. Também conhecida como domótica, junção das palavras “domus” que quer dizer “casa” e “robótica”, que é a automatização e controle de um processo, a automação residencial tem seu marco inicial nos anos 70, com o lançamento dos chamados X-10, primeiros módulos inteligentes que possibilitava a comunicação entre vários dispositivos de comunicação com a tecnologia PLC (*Power Line Carrier*). Com o avanço da tecnologia começou a ser utilizado o PC (*Personal Computer*) como central de automação, mas com algumas restrições, que contribuíram para o surgimento dos dispositivos dedicados, os chamados embarcados, com o uso de microprocessadores e microcontroladores, em substituição aos PCs (BORTOLUZZI, 2013).

2.2 SISTEMAS DE CONTROLE

Sistema pode ser definido como disposição, conjunto ou coleção de partes relacionadas ou conectadas entre si, formando um todo. O controle é obtido a partir da análise e atuação sobre um determinado sistema com o objetivo de se atingir um determinado resultado. O sistema de controle pode então ser considerado como o conjunto formado pelo sistema a ser controlado e o dispositivo utilizado para esse fim, chamado de controlador (OGATA, 1982).

TIPOS DE SISTEMAS

- **SISTEMAS LINEARES:** São os sistemas onde se aplica o princípio da superposição, cuja resposta produzida pela aplicação simultânea de duas funções diversas é a soma de suas respostas individuais;
- **SISTEMAS DE PRIMEIRA ORDEM:** São os sistemas com equações de primeiro grau, sendo sua variável com exponencial unitário. Sistemas de primeira ordem podem ser controlados e estabilizados com maior facilidade e rapidez;
- **SISTEMAS DE SEGUNDA ORDEM:** São os sistemas cujas funções são comparadas à equações de segundo grau, possuindo variável com exponencial igual a 2. Esse tipo de sistema, em função de um sobressinal, possui tempo de acomodação e estabilização mais elevado (OGATA, 1982).

2.2.1 SINAIS DE ENTRADA E SAÍDA

Os sinais de entrada são todos aqueles que fornecem informações ao sistema para seu processamento. O processamento desses sinais controlados pela programação do sistema gerará os sinais de saída, que serão responsáveis por ações como acionamento, interrupção, controle do nível de potência de algum equipamento etc (OGATA, 1982).

2.2.2 SISTEMA DE CONTROLE EM MALHA ABERTA

Segundo Ogata (1982) são sistemas que não possuem realimentação, ou seja, não há medição na saída. Esse tipo de sistema não faz comparação entre os valores de entrada e saída.

No controle em malha aberta é aplicado um sinal de controle pré-determinado, esperando-se após um determinado tempo a variável controlada atinja um determinado resultado. Neste tipo de sistema de controle não são utilizadas informações sobre evolução do processo para a determinar o sinal de controle a ser utilizado.

A figura 1 mostra um sistema de malha aberta.

Figura 1: Sistema de malha aberta



Fonte: (OGATA, 1982)

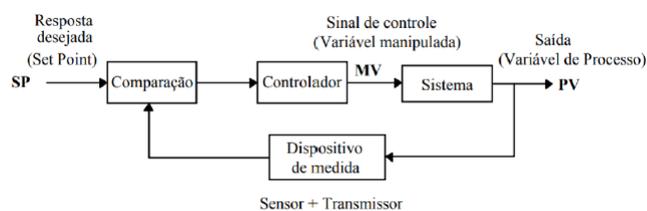
2.2.3 SISTEMA DE CONTROLE EM MALHA FECHADA

É aquele no qual o sinal de saída é usado na ação de controle. Sistemas de controle em malha fechada são sistemas realimentados. A realimentação pode ser usada com o objetivo de reduzir o erro do sistema (OGATA, 1982).

No controle em malha fechada, informações sobre como a saída de controle está evoluindo são utilizadas para determinar o sinal de controle que deve ser aplicado ao processo em um determinado instante.

A figura 2 mostra um sistema de malha fechada.

Figura 2: Sistema de malha fechada



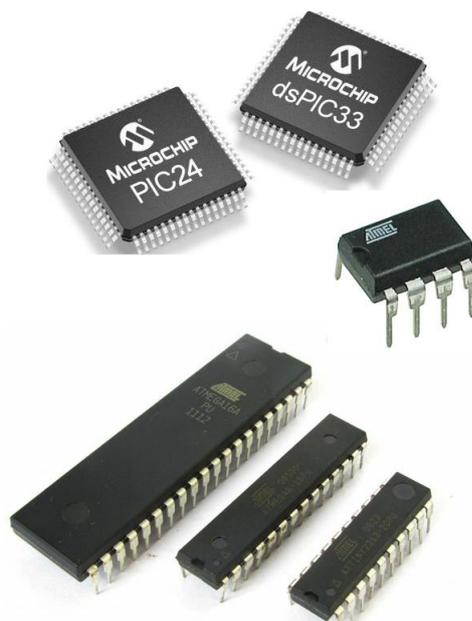
Fonte: (OGATA, 1982)

2.3 MICROCONTROLADOR

O microcontrolador pode ser definido como sendo um modelo avançado de circuito integrado, composto pelos mesmos componentes de um computador (CPU - *Central Processing Unit*, memória, portas de entrada e saída, conversores A/D - Analógico/Digital, além de outros possíveis periféricos), com a característica de ser programável para executar as mais variadas funções, da mais simples à mais complexa, desde LEDs (*Ligth Emitting Diode*) e sensores a acionamento de motores. Através de programação mais comumente nas linguagens Assembly ou C pode-se controlar o *hardware* para executar com perfeição tarefas determinadas (PALMIERE, 2016).

Há atualmente no mercado várias marcas de microcontroladores, entre elas as mais conhecidas são Microchip e Atmel. A figura 3 mostra alguns modelos de microcontroladores dos fabricantes citados. Outra vantagem no uso do microcontrolador é o acesso facilitado devido ao seu baixo custo.

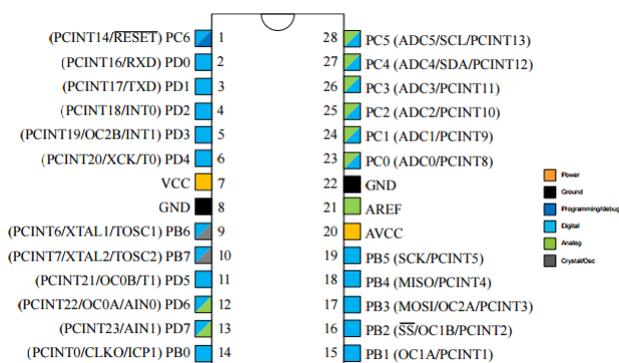
Figura 3: Microcontroladores da Atmel e Microchip



Fonte: http://www.newtoncbraga.com.br/images/stories/artigos2017/mic0139_0002.jpg

Os microcontroladores podem ser de 8, 16 ou 32 bits, e variam também em quantidade disponível de memória RAM (*Random Access Memory*), FLASH e EEPROM (*Electrically Erasable Programmable Read-Only Memory*), quantidade de pinos e frequência máxima de clock. Segundo Braga (2014) a velocidade de processamento desses dispositivos é surpreendente, atingindo milhares ou milhões de operações por segundo. Grande parte dos modelos existentes utiliza dois tipos de arquitetura: RISC (*Reduced Instruction Set Computer*) e Harvard. A figura 4 mostra a estrutura dos pinos do microcontrolador ATmega328, da Atmel, que será utilizado nesse trabalho.

Figura 4: Microcontrolador ATmega328



Fonte: http://www.atmel.com/pt/br/Images/Atmel-42735-8-bit-AVR-Microcontroller-ATmega328-328P_Summary.pdf

Os microcontroladores possuem saídas PWM (*Pulse Width Modulation*) que podem ser programadas para controle dos motores. Por necessitar de elevadas correntes para esse tipo de acionamento, a ligação é feita de forma indireta, onde o dispositivo pode controlar shields

(dispositivos intermediários ou drivers), que são capazes de trabalhar com correntes mais altas (BRAGA, 2014).

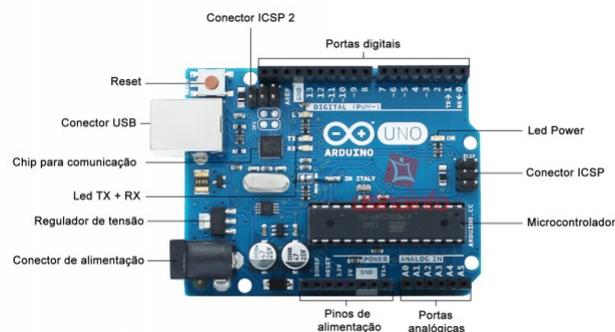
Para serem usados os microcontroladores precisam ser montados em placas específicas e acoplados a um computador para serem programados. Uma maneira mais fácil é adquirir a placa já montada com o microcontrolador e *hardwares* prontos com botões, display, luzes, sensores, acionadores e muito mais. Um exemplo de sistemas já montados é o Arduino, que ao contrário do que muitos pensam não é um microcontrolador, mas sim uma placa de prototipagem eletrônica que traz acoplado o microcontrolador AVR da Atmel (BRAGA, 2014).

2.4 ARDUINO

O Arduino é uma plataforma de prototipagem eletrônica de livre acesso, de origem italiana, com *hardware* e *software* de fácil utilização. Por ser uma tecnologia *open-source* (as placas e *software* são de código aberto) e com *hardware* facilmente adaptável a sistemas computacionais e principalmente de baixo custo, o Arduino é muito utilizado nas mais variadas áreas para automação de infinitas atividades, desde tarefas mais simples a projetos mais complexos. Os projetos com Arduino estão disponíveis gratuitamente conforme licença da *Creative Commons* (MONK, 2017).

O Arduino foi criado inicialmente como uma ferramenta de prototipagem rápida, para uso didático, mas seu sucesso rapidamente se espalhou e para corresponder às novas demandas modificou sua oferta de placas simples de 8 bits para produtos para aplicativos IoT (*Internet of Things*), *wearable*, impressão em 3D e ambientes incorporados, passando a ser comercializado a partir de 2005. Seu *software* é de fácil entendimento para iniciantes, mas flexível o suficiente para profissionais, com a vantagem de poder ser executado no Mac, Windows e Linux, ao passo que a maioria dos outros sistemas de microcontroladores são compatíveis apenas com o Windows (ARDUINO, 2017).

A figura 5 mostra uma placa do Arduino UNO, um dos modelos mais simples disponível no mercado e escolhido para ser utilizado nesse trabalho.

Figura 5: Placa de Prototipagem Arduino UNO

Fonte: <https://static.usinainfo.com.br/1003853-infografico/arduino-uno-r3-cabo-usb.jpg>

O microcontrolador que comanda o Arduino é da família ATmega, fabricado pela empresa Atmel. A ligação da placa ao computador é feita através da conexão USB (*Universal Serial Bus*). Essa conexão é necessária para programar o Arduino, e para isso é utilizada a linguagem de programação C. A possibilidade de ligação de dispositivos eletrônicos externos como relés, sensores, motores, microfones e outros, se dá através de um conjunto de pinos de conexão. A placa pode ser alimentada pela conexão USB ligada ao computador, por uma bateria de 9V (*Volt*) ou por fonte de alimentação. Existe uma quantidade considerável de placas acessórias denominadas *shields*, que podem ser acopladas ao Arduino para acesso a alguns recursos como o Ethernet, que o permite funcionar como servidor Web, o *shield* Motor que permite o acionamento de motores elétricos, o USB Hoste que controla dispositivos USB e o Relays que auxilia no comando de relés. A figura 6 mostra alguns exemplos de *shields* (ARDUINO, 2017).

Figura 6: Shields para Arduino

Fonte: https://store.open-electronics.org/image/cache/catalog/data/Sistemidisviluppo-Softwareedidattica-Libri-Documentazionetecnica/8190-BATTERYSHIELD_ard-500x500.jpg

A tabela 1 apresenta as especificações da placa Arduino UNO R3.

Tabela 1: Especificações da placa Arduino UNO R3

Microcontrolador:	ATmega328P
Tensão de operação:	5V
Tensão de entrada (recomendada):	7-12V
Tensão de entrada (limites):	6-20V
Pinos de entrada/saída digitais:	14 (6 pinos podem fornecer saída PWM)
Pinos de entrada analógica:	6
Corrente DC por pino de E/S:	20 mA
Corrente DC por pino de 3,3V:	50 mA
Memória Flash:	32 kB
SRAM:	2 kB
EEPROM:	1 Kb
Frequência de clock:	16 MHz

Fonte: <https://www.arduino.cc/en/Main/ArduinoBoardUnoSMD>

2.4.1 LINGUAGEM DE PROGRAMAÇÃO C

O Arduino tem sua programação baseada na Linguagem C. Trata-se de uma linguagem considerada de nível médio devido à combinação de caracteres de baixo e alto nível, o que possibilita a criação de programas simples e robustos. Serão apresentados a seguir alguns conceitos básicos sobre essa linguagem (GAIER, 2011).

Na Linguagem C letras maiúsculas e letras minúsculas fazem muita diferença. No final de cada linha do programa é necessário o uso do “;”.

2.4.1.1 Funções-Base

É obrigatório o uso da função `void setup()`, considerada função de inicialização, e a função `void loop()`, que deverá conter dentro dela tudo que se deseja que o programa execute.

A função `pinMode(pino,MODO)` é responsável por configurar um pino específico da placa, como entrada ou saída. Nesta função a informação pino é um número inteiro e o MODO refere-se à informação de INPUT ou OUTPUT, ou seja, entrada ou saída de dados.

Existem dois tipos de funções, as que retornam algum valor para a função onde está inserida e as que não retornam nenhum valor (GAIER, 2011).

2.4.1.2 Entrada e Saída de Dados

São comandos para introduzir ou ler dados recebidos de uma porta. Podem ser: Portas digitais: o comando `digitalRead(pino)` é responsável pela leitura de dados nas portas digitais.

Portas analógicas: o comando `analogRead(pino)` é responsável pela leitura de alguma entrada analógica. O uso deste comando retorna um valor inteiro entre 0 e 1023, que equivale a aproximadamente 5mV (Milivolt) por unidade.

Em *software* o comando `analogReference(TIPO)` define qual a tensão de referência será usada em *hardware*. O TIPO será DEFAULT, INTERNAL ou EXTERNAL, ou seja, padrão, interna ou externa, respectivamente.

O comando `analogWrite(pino, valor)` controla os pinos PWM. No Arduino Uno os pinos correspondentes são o 3, 5, 6, 9, 10 e 11. O valor alterna entre 0 e 255 (GAIER, 2011).

2.4.1.3 Tempo

Segundo Gaier (2011) o comando `delay(tempo em ms-milissegundo)` estabelece uma pausa do programa que está sendo executado correspondente à quantidade de milissegundos informada. Para o comando `delayMicroseconds(tempo em μ s-microsegundo)` será determinada uma pausa do programa correspondente à quantidade de microssegundos informada. O comando `millis()` retorna a quantidade de tempo que passou, em milissegundos, desde que o programa começou a ser executado. Esse comando deve ser usado em conjunto com a variável `unsigned long`.

2.4.1.4 Bibliotecas

Alguns equipamentos ou acessórios que são adquiridos para uso com o Arduino já vem com uma biblioteca pronta para facilitar a programação. Nesses casos é só adicioná-la ao programa, verificando se suas instruções estão corretas, em condições de serem utilizadas na programação (GAIER, 2011).

2.4.1.5 Variáveis

Monk (2017) afirma que uma variável é o processo de identificação de um número ou dado. Os nomes das variáveis não podem começar com letras ou sublinhado (`_`) e não podem ser iguais a qualquer palavra reservada pelo programa ou nome de alguma biblioteca. Elas devem ser declaradas antes de serem usadas.

2.4.1.6 Classes de Variáveis

As variáveis se classificam em:

Locais: são variáveis declaradas dentro de uma função específica. Elas só existirão enquanto o bloco em que elas estiverem declaradas estiver em execução.

Globais: são variáveis conhecidas pelo programa e funcionarão normalmente independente da

função que estiver sendo executada. São declaradas antes da função void setup().

Estáticas: apesar de conservarem seu valor durante a execução de outras funções, só são reconhecidas dentro da função onde são declaradas (GAIER, 2011).

2.4.1.7 Tipos de Dados e Modificadores

Conforme afirma Gaier (2011) a linguagem de programação do Arduino existe cinco tipos de dados: char, int, float, void e double.

Char: é um tipo de dado que reserva 1 byte de memória para guardar o valor de 1 caractere.

Int: é a variável padrão do programa. Ela memoriza dados de -32.768 a 32.767. Através da propriedade matemática complemento de dois é possível que essa variável memorize números negativos.

Float: é um tipo de dado real de precisão simples.

Void: é um tipo de dado usado apenas na declaração de funções, indicando que a função requerida não retornará nenhum valor para a função que a chamou.

Double: é um tipo de dado real de precisão dupla.

Modificadores: para cada tipo de variável existem os modificadores de tipos: signed, unsigned, short e long. Contudo, no dado float não é possível se aplicar nenhum modificador, e no dado double o único modificador aceito é o long.

2.4.1.8 Estruturas de Controle de Fluxo

Segundo Gaier (2011) as estruturas de controle de fluxo são a parte mais importante da programação em Arduino, uma vez que toda a programação é executada em torno delas:

IF: verifica se determinada condição foi obtida. Se o resultado da condição for 0 (falsa), ela não será executada, se for 1 (verdadeira), será executada. O formato desta estrutura de controle de fluxo é: IF...ELSE. O uso do if/else permite um controle maior sobre a estrutura. Considera-se o else como um complemento do comando if.

FOR: é uma das estruturas que trabalha com loops de repetição. Empregada para repetir um bloco de informações definidas na função. Enquanto a condição for verdadeira, as informações existentes na função serão executadas.

SWITCH CASE: empregado para programar blocos de instruções distintos para condições distintas.

WHILE: esta estrutura funciona como um loop de repetição. Enquanto a condição declarada entre parênteses for verdadeira, a estrutura será executada. A partir do instante que a condição for falsa, o programa seguirá.

DO-WHILE: esta estrutura é a última de controle de loop repetitivo. Seu funcionamento é parecido com as estruturas for e while, porém, ao contrário.

2.5 DISPLAY LCD

Para que o display de cristal líquido funcione corretamente no Arduino sua base deve ser similar ao CHIPSET HITACHI. Tomaremos como referência o modelo alfanumérico 16x2 (16 colunas, 2 linhas) mostrado na figura 7. O referido modelo será utilizado nesse trabalho.

Figura 7: Display LCD 16x2



Fonte: https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcQV4C47wNM1qgyAw1wkPcQu25LzJBL2iPOLs_sS_78h3Oe1JF3F

Modelos de display compatíveis com o Chipset Hitachi podem operar nos modos de 4 bits ou 8 bits, sendo que no modo 4 bits são necessários apenas 7 pinos I/O (Entrada/Saída) do Arduino e no modo 8 bits são necessários 11 pinos I/O. Para controle do contraste do monitor, é utilizado um potenciômetro para regulagem da tensão aplicada ao pino VEE/VO do display (GAIER, 2011). A tabela 2 apresenta as especificações do display de LCD mencionado.

Tabela 2: Especificações do Display LCD16x2

VSS:	alimentação do display (<i>Ground</i> , Terra)
VDD:	alimentação do display (5V)
VEE/VO:	ajuste de contraste
RS (<i>Register Select</i>):	controla em qual parte do LCD os dados são enviados
RW (<i>Read/Write</i>):	seleciona o modo (leitura/envio de dados)
E (<i>Enable</i>):	habilita o envio de dados aos registros
D0 a D7:	controla o estado dos pinos (<i>HIGH</i> ou <i>LOW</i>)

Fonte: (GAIER, 2011)

2.6 SENSORES

Segundo Gaier (2011) sensor é um dispositivo que responde por meio de sinal elétrico a entradas específicas provenientes do ambiente físico. Existe uma infinidade de sensores disponíveis no mercado, que são classificados quanto à forma de energia que eles captam, conforme mostra a tabela 3.

Tabela 3: Tipos de Sensores

De luz:	células solares, fototransistores, fotodiodos
De som:	microfones, sensores sísmicos
De temperatura:	termômetros, termistores, termostato
De calor:	calorímetro
De radiação:	dosímetro, contator Gaiger
De resistência elétrica:	ohmímetro
De corrente elétrica:	galvanômetro, amperímetro
De tensão elétrica:	voltímetro
De potência elétrica:	wattímetro
Magnéticos:	magnetômetro, compasso magnético
De pressão:	barômetro
De movimento:	velocímetro, tacômetro
Mecânico:	sensor de posição
De proximidade	
De distância	

Fonte: (GAIER, 2011)

Os sensores transformam entradas físicas em tensão elétrica, enviando em seguida ao Arduino. Para programação do sistema deve-se levar em conta as características de cada sensor e o tipo de entrada, que pode ser analógica ou digital. A maioria dos fornecedores disponibilizam informações para a utilização de seus sensores com o Arduino, além de bibliotecas já prontas, o que facilita consideravelmente a vida dos usuários, permitindo uma programação mais rápida e segura. Para os sensores analógicos são utilizadas as funções `analogRead()` para receber os sinais e `analogWrite()` para enviar informações aos sensores. Esses sensores recebem sinais de 0 a 1023, cabendo ao usuário/programador identificar a grandeza medida. Já os sensores digitais trabalham simplesmente com sinal alto (*HIGH*) ou baixo (*LOW*). A leitura desses sensores é feita através da função `digitalRead()` com valores 0 para sinal baixo e 1 para sinal alto (GAIER, 2011).

2.6.1 Sensor de Temperatura

O sensor de temperatura DS18B20 pode ser usado nos mais variados ambientes, inclusive submerso, por ser à prova d'água. Por essa razão é confeccionado em aço inoxidável e cabo protegido por material isolante. Uma característica desse dispositivo é ser capaz de captar sinais com resolução de 9 a 12-bit, e possuir uma interface de comunicação simples por meio de um fio único, função conhecida como *1-Wire*, que permite ligar vários sensores de temperatura em uma única saída digital do Arduino (USINAINFO, 2017). Pelas características apresentadas e a relação custo-benefício esse sensor, mostrado adiante na figura 8, foi escolhido para ser utilizado nesse trabalho.

Figura 8: Sensor de temperatura digital DS18B20

Fonte: http://www.webtronico.com/image/cache/data/produtos/DS18b20_sensor_de_temperatura_prova_dagua-500x500.jpg

Na tabela 4 são apresentadas as especificações do sensor de temperatura DS18B20.

Tabela 4: Especificações do Sensor de Temperatura DS18B20

Tensão de alimentação:	3.0 - 5.5V DC
Leitura de temperatura entre:	-55°C a +125°C
Fios:	Vermelho (VCC), Branco (DATA) e Preto(GND)
Tempo de atualização:	<750ms
Precisão:	0,5°C
Resolução:	9 ou 12 bits
Interface:	1 fio (1-Wire)
ID (Identidade):	único de 64 bits
Diâmetro:	6mm (Milímetro)
Comprimento do fio:	90cm (Centímetro)
Peso:	21g

Fonte: (USINAINFO, 2017)

2.7 ELETROVÁLVULAS

Como o próprio nome sugere, esse dispositivo nada mais é que uma válvula comandada eletricamente. É um dispositivo que tem por finalidade abrir ou fechar um fluxo de água, ar ou gás. A válvula é composta mecanicamente de um registro com um êmbolo, que ao movimentar abre ou fecha o fluxo. A parte de comando é constituída por uma bobina solenoide (um ímã

artificial) que submetido a uma corrente elétrica gera um campo magnético que atrai o êmbolo da parte mecânica, abrindo o registro. Quando a corrente elétrica é interrompida o campo magnético se extingue deixando de atrair o êmbolo, que empurrado por uma mola retorna para a posição inicial, interrompendo o fluxo (BRAZ, 2013).

Existem tipos diferentes de eletroválvulas variando o material empregado na sua fabricação de acordo com o fluxo a ser controlado por ela. Os modelos com mecanismo construído em metal são destinados ao controle do fluxo de gases, óleo diesel e líquidos com temperatura elevada; os feitos em plástico controlam líquidos como água e para líquidos corrosivos são usadas as de mecanismo construído com plásticos especiais.

A eletroválvula é considerada um dispositivo de suma importância na área da automação, pois seus recursos possibilitam seu uso em várias aplicações como controle de combustível de um veículo, controle de fluxo de água em sistemas de aquecimento solar, controle de nível de reservatórios dentre outras (BRAZ, 2013).

A figura 9 mostra uma eletroválvula para controle de fluxo de água, no modelo escolhido para realização desse trabalho, devido às suas características e por ser compatível para uso com o Arduino.

Figura 9: Eletroválvula para Água



Fonte: http://www.msseletronica.com/imagens/1015_641.jpg

A tabela 5 apresenta as características e especificações da eletroválvula empregada nesse trabalho:

Tabela 5: Características e Especificações da Eletroválvula

MATERIAIS	Corpo:	termoplástico
	Filtro:	aço inox
	Partes metálicas:	aço zincado
	Membrana:	borracha (padrão)
	Terminais:	latão
PRESSÃO DE OPERAÇÃO	De 0,2 à 8 kgf/cm ²	
	À 0,2kgf/cm ² , vazão mínima:	7 l/min (Litro/minuto)
	À 8 kgf/cm ² ; vazão máxima:	40 l/min
DEMAIS CARACTERÍSTICAS	Temperatura Máxima do Líquido:	60°C
	Rigidez Dielétrica:	1.500 Vca - 1 min
	Diâmetro de conexão:	3/4"
	Saída de Água:	180°
	Número de Entradas:	1
	Número de Saídas:	1
	Vida Útil:	50.000 operações
	Tipo de Terminal (alimentação):	Faston 6,3mm x 0,8mm
	Tensão Bobina:	127Vca
	Consumo de corrente:	74mA

Fonte: (MSSELETRÔNICA, 2017)

3 PROCEDIMENTOS METODOLÓGICOS

O estudo de caso proposto nesse trabalho tem como problema fundamental o desperdício de água ao ser utilizado um chuveiro abastecido com água aquecida por um sistema de energia solar. Para resolver esse problema foi proposta a automatização do chuveiro com a confecção de um protótipo.

3.1 ESTUDO DE CASO

- Localização da instalação: Centro de Caratinga-MG
- Tipo de imóvel: residencial
- Número de usuários: 4
- Características: apartamento no primeiro andar de um edifício de 4 pavimentos e sistema de aquecimento solar instalado acima do terraço. Ponto de uso (chuveiro) distante do reservatório de água quente do sistema solar, sendo apontado como a principal causa do problema em estudo.

Para realização desse trabalho foi utilizada pesquisa do tipo experimental, sendo nótoria a gama de conhecimentos adquiridos com a confecção de um protótipo e a realização de testes. Para cumprimento da proposta foram realizadas as etapas previstas no capítulo 1, que serão elencadas na sequência.

3.2 ANÁLISE DA VIABILIDADE

Foram estudadas alternativas que fossem viáveis tecnicamente. Nesse sentido optou-se pela utilização do Arduino devido às suas características já mencionadas, destacando-se o custo benefício e a facilidade de acesso por se tratar de uma tecnologia *open-source*. Na sequência foi realizada uma pesquisa de preços para aquisição dos materiais escolhidos. A aquisição dos componentes eletroeletrônicos selecionados foi realizada com base nas especificações apresentadas como facilidade no acesso à programação e manuseio, funcionalidades e custo, com respaldo nas informações dos fabricantes e nas literaturas consultadas, conforme descrito no capítulo 2.

A compra de todo material foi realizada com base no menor preço preservando-se as características de qualidade e aplicabilidade dos produtos. A tabela 6 exhibe a relação dos componentes adquiridos para a construção do protótipo.

Tabela 6: Relação dos Componentes

DESCRIÇÃO	QUANTIDADE
Arduino Uno R3	1
Buzzer 5V Ativo	1
Case para Arduino UNO	1
Chave Botoeira	1
Display LCD 16x2	1
Eletroválvula para Água 1/2"	2
Fonte de Alimentação 9V	1
Jumpers 20cm - 20 unidades	1
LED 3mm Alto Brilho	1
Módulo Relés 4 Canais	1
Módulo Serial I2C LCD	1
Quadro Distribuição PVC	1
Relé de Estado Sólido 50A	1
Sensor de Temperatura Digital DS18B20	2
Teclado Matricial 4x4 Alfanumérico	1

Fonte: Autor

3.3 COLETA DE DADOS

Para definição dos parâmetros do programa foram feitas 12 medições entre os dias 23 de setembro e 23 de outubro de 2017 em diferentes horários, conforme tabela 7. A temperatura da água indicada na tabela foi medida na saída do chuveiro, com utilização de um multímetro digital modelo ET-2082C da MINIPA.

Tabela 7: Medições da Temperatura Máxima da Água Aquecida pelo Sistema Solar

DATA	HORÁRIO	TEMPERATURA °C	CONDIÇÃO DO TEMPO
23/set/17	20:00	47	Bom/Sol
24/set/17	11:00	47	Bom/Sol
25/set/17	07:00	40	Bom/Sol
26/set/17	12:30	54	Quente/Sol
26/set/17	19:00	46	Bom/Sol
27/set/17	19:00	36	Nublado
27/set/17	12:15	48	Bom/Sol
28/set/17	18:30	46	Bom/Sol
01/out/17	17:00	48	Bom/Sol
16/out/17	18:26	47	Bom/Sol
17/out/17	18:20	37	Nublado
23/out/17	18:15	57	Quente/Sol

Fonte: Autor

No mesmo período de coleta de dados citado anteriormente foram feitas medições da quantidade de água fria que era necessário descartar para que a água quente chegasse até o chuveiro, sendo constatado que até então havia um desperdício de 10 litros de água a cada banho. Considerando-se que cada um dos quatro moradores dessa residência tome um banho por dia em horários diferentes o desperdício de água pode chegar a 1.200 litros por mês.

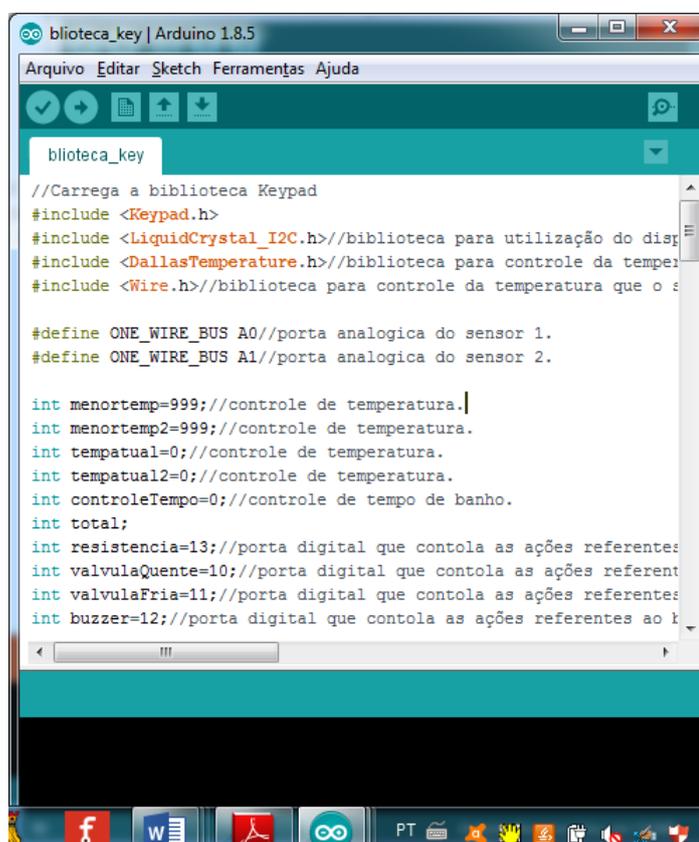
3.4 PROGRAMAÇÃO DO SISTEMA

Após aquisição de todo material e coletados os dados necessários, foi feita a programação do sistema com todas as funções desejadas, através de uma IDE (*Integrated Development Environment*), *software* gratuito da Arduino, onde são criados os *sketches* (esboços) da programação, compilados e transmitidos para o microcontrolador.

Segundo Souza (2013) o ciclo de programação do Arduino se divide em conectar a placa Arduino ao computador através da porta USB, criar os *sketches* com os comandos para o Arduino, *Upload* dos *sketches* criados para o microcontrolador e reinicialização do sistema.

Após a reinicialização os *sketches* criados poderão ser executados. Com o *upload* concluído o Arduino poderá ser desconectado do computador, mas para ser utilizado precisará ser alimentado por uma fonte de energia. A figura 10 mostra uma tela da IDE do Arduino.

Figura 10: IDE do Arduino



```
bliblioteca_key | Arduino 1.8.5
Arquivo Editar Sketch Ferramentas Ajuda
bliblioteca_key
//Carrega a biblioteca Keypad
#include <Keypad.h>
#include <LiquidCrystal_I2C.h>//biblioteca para utilização do display
#include <DallasTemperature.h>//biblioteca para controle da temperatura
#include <Wire.h>//biblioteca para controle da temperatura que o sensor

#define ONE_WIRE_BUS A0//porta analogica do sensor 1.
#define ONE_WIRE_BUS A1//porta analogica do sensor 2.

int menortemp=999;//controle de temperatura.
int menortemp2=999;//controle de temperatura.
int tempatual=0;//controle de temperatura.
int tempatual2=0;//controle de temperatura.
int controleTempo=0;//controle de tempo de banho.
int total;
int resistencia=13;//porta digital que contola as ações referentes
int valvulaQuente=10;//porta digital que contola as ações referentes
int valvulaFria=11;//porta digital que contola as ações referentes
int buzzer=12;//porta digital que contola as ações referentes ao k
```

Fonte: Autor

3.5 FUNÇÕES IMPLEMENTADAS NO SISTEMA

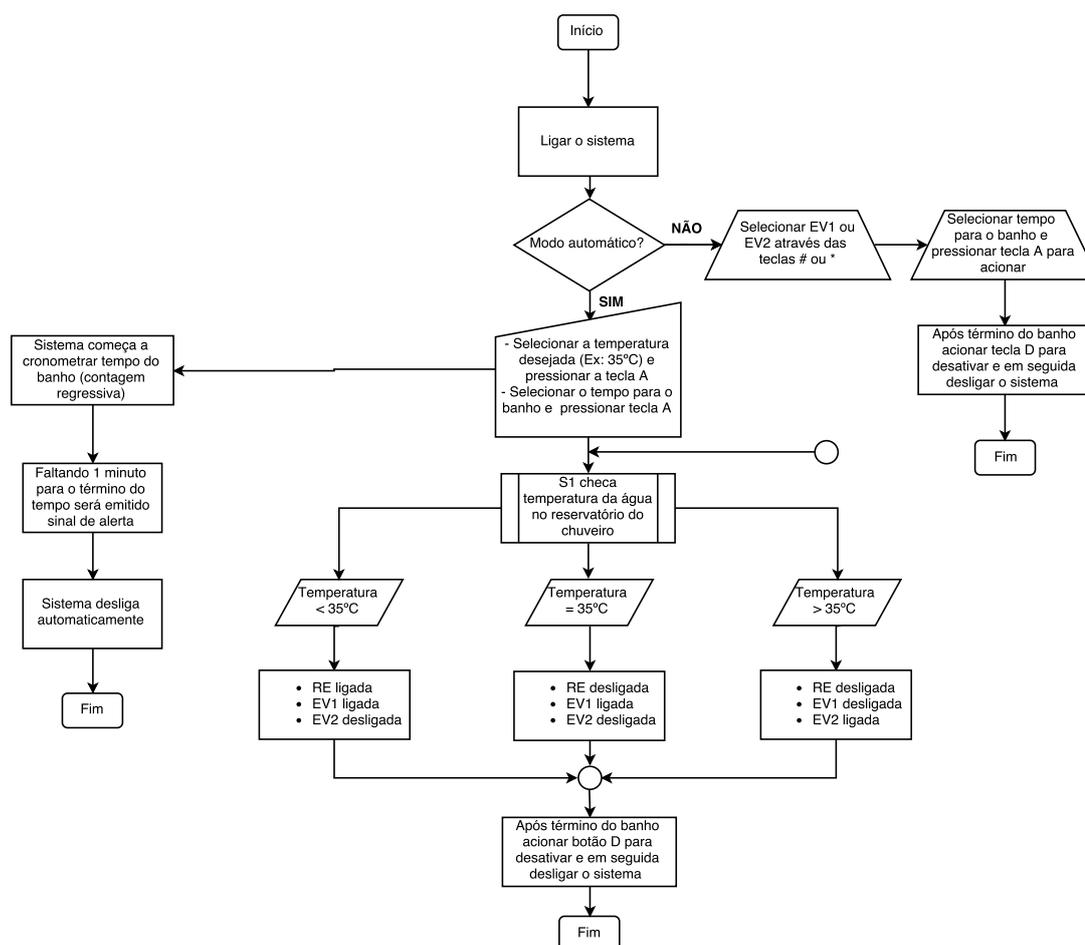
Foram implementadas na programação do controle as seguintes funções:

- Apresentação da temperatura da água na tela do display;
- Apresentação do tempo de duração do banho na tela (contagem regressiva);
- Escolha pelo usuário da temperatura para o banho. O sistema aceitará um valor de temperatura entre 25°C e 50°C. Caso não queira selecionar uma temperatura, o usuário poderá confirmar a temperatura padrão do sistema, que é de 35°C;
- Configuração do tempo máximo para o banho. O sistema foi configurado para um tempo máximo de 10 minutos, podendo ser alterado pelo usuário para um tempo de até 15 minutos. Caso não queira alterar o tempo de banho, o usuário poderá confirmar o tempo padrão do sistema. Se for atingido o tempo escolhido para o banho o sistema será desligado automaticamente.
- Opção de utilização do chuveiro no modo manual, com acionamento direto das válvulas de água quente ou fria. Nessa opção não há escolha de temperatura e acionamento da resistência elétrica.

3.5.1 FLUXOGRAMA

Para auxiliar na configuração do sistema com implementação das funções estabelecidas foi feito um fluxograma de operação do controle eletrônico, exibido na figura 11.

Figura 11: Fluxograma de Operação



Fonte: Autor

3.6 FUNCIONAMENTO DO SISTEMA

A configuração do sistema está descrita passo a passo nos itens relacionados a seguir:

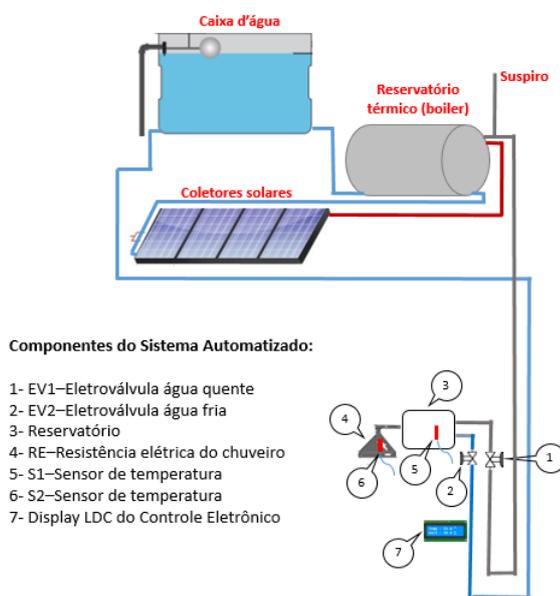
- No início do banho será acionado pelo usuário o botão para ligar o sistema;
- Em seguida o usuário poderá optar pelo acionamento direto das válvulas de água quente ou fria, bastando para isso pressionar a tecla "#" para abertura da válvula de água quente ou a tecla "*" para acesso a água fria, escolhendo em seguida o tempo de banho. Ao pressionar a tecla "A" será acionada a válvula selecionada. A outra opção é o modo automático, que se desdobra no cumprimento das etapas seguintes;
- Deve ser indicada no controle eletrônico a temperatura desejada da água para o banho, sendo pressionada em seguida a tecla "A". Caso não queira selecionar outra temperatura, o usuário poderá confirmar a temperatura padrão do sistema;
- Na tela seguinte o usuário deverá selecionar o tempo de banho ou confirmar o tempo padrão do sistema que é de 10 minutos, pressionando em seguida a tecla "A". A contagem

regressiva do tempo de banho será iniciada simultaneamente ao acionamento do sistema, e será exibida na tela do controle;

- Ao acionar o sistema, estando a temperatura da água abaixo da selecionada pelo usuário, serão acionadas RE (Resistência Elétrica) e EV1 (Eletroválvula da Água Quente). A resistência aquecerá a água fria presente na tubulação do sistema de aquecimento solar até que a água aquecida chegue ao chuveiro;
- Quando S1 (sensor de temperatura 1) indicar que a água vinda do boiler atingiu a temperatura desejada, RE será desligada;
- Quando S1 indicar que a água atingiu a temperatura indicada + 1 grau, EV1 será desligada sendo acionada EV2 (Eletroválvula da Água Fria). Havendo queda da temperatura até o valor predeterminado o processo se inverterá, sendo acionada EV1 e desligada EV2, e assim sucessivamente até que o sistema seja desligado. Se a temperatura atingir valor menor que o escolhido, RE será acionada novamente. O controle do sistema é feito pelo sensor S1 a partir da leitura da temperatura da água que está armazenada no reservatório do chuveiro, com capacidade para 2,1 litros de água, onde é feita a mistura de água quente e fria;
- Se o cronômetro atingir a marca de 1 minuto para o término do banho será emitido um alerta sonoro e o display piscará até o término desse tempo. Se não for desligado pelo usuário, o sistema se desligará automaticamente;
- Se o usuário terminar o banho antes do tempo limite bastará apenas pressionar o botão "D" para colocar o sistema em stand by e em seguida desligá-lo;
- Se for informado algum valor de temperatura ou tempo incorretos basta ao usuário pressionar a tecla "B" para limpar a tela e informar novamente os dados.

A figura 12 mostra o esquema de montagem do sistema com a identificação dos principais componentes.

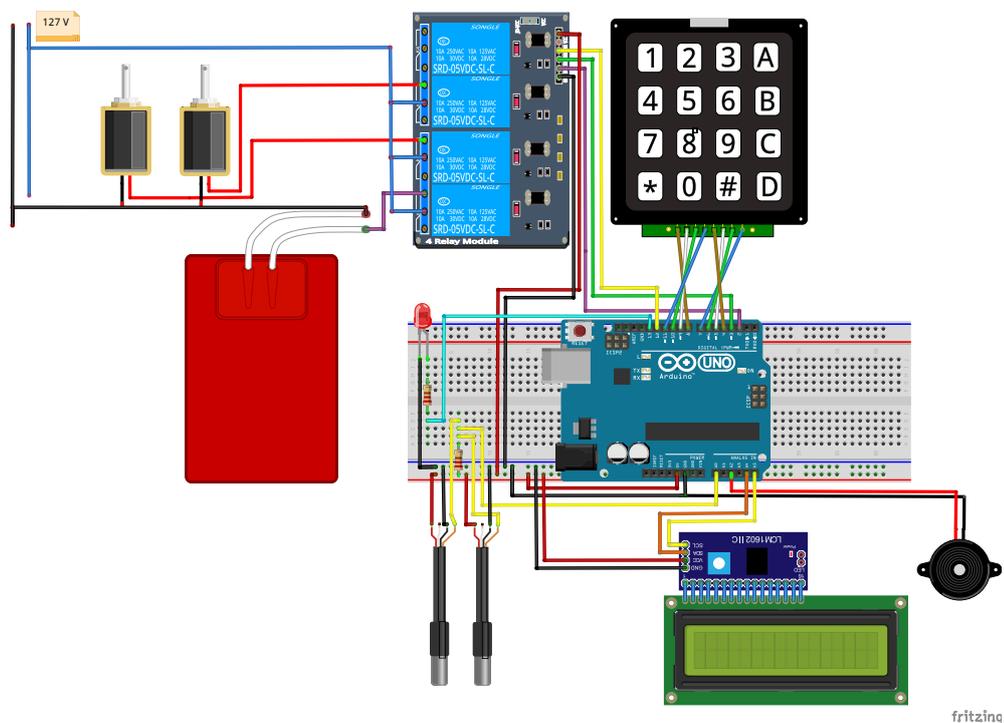
Figura 12: Esquema de Montagem do Sistema



Fonte: Autor

Para melhor compreensão sobre o funcionamento do sistema e auxílio na montagem do circuito, foi feita no software *Fritzing* a montagem do circuito elétrico de controle e acionamento do sistema, conforme mostra a figura 13.

Figura 13: Circuito de Controle e Acionamento do Protótipo



Fonte: Autor

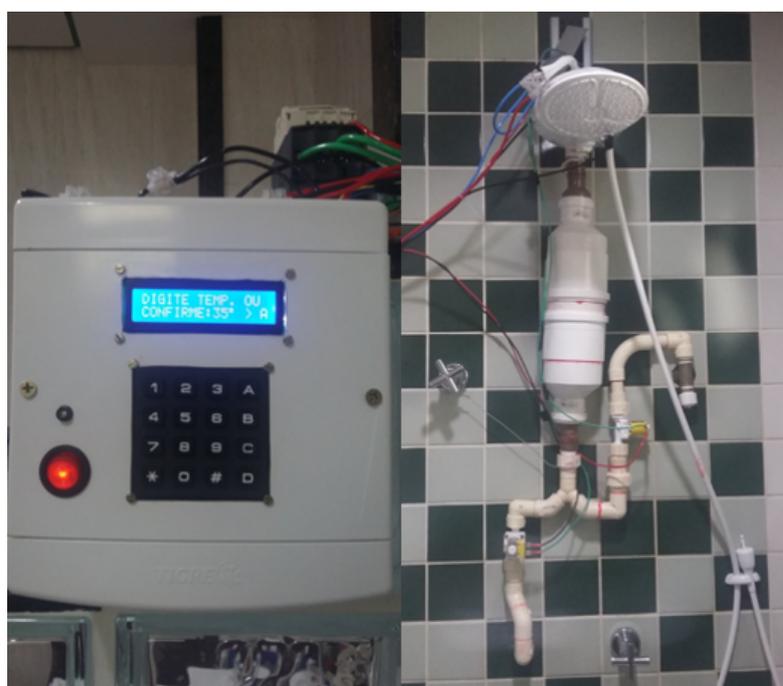
Conforme o Manual de Características Técnicas do fabricante o modelo do chuveiro instalado tem uma potência de 5.500W e tensão de 127V (LORENZETTI, 2015). Com essas informações foi possível determinar a corrente máxima do chuveiro com aplicação da seguinte equação:

$$I = \frac{P}{V} \quad (3.1)$$

Os cálculos realizados indicaram uma corrente de 43 A, mas medições realizadas com o circuito em funcionamento através do Alicata Amperímetro modelo ET-3201 do fabricante MINIPA indicaram uma corrente de 38 A.

- Instalação hidráulica, sendo necessário algumas adaptações na rede existente no banheiro. Para conexão do circuito hidráulico do protótipo com à rede já existente foi necessária a confecção em torno mecânico de dois adaptadores de cobre com conexões de roscas específicas que não estão disponíveis para aquisição no mercado. Isso permitiu que o protótipo fosse instalado sem a necessidade de danificar a estrutura de alvenaria e acabamento do banheiro. A construção do reservatório para o chuveiro com capacidade para 2,1 litros de água, onde é feita a mistura da água fria com a água quente, também exigiu algumas adaptações, mas sua montagem foi satisfatória para funcionamento do sistema. A figura 15 mostra o sistema já instalado.

Figura 15: Instalação do Protótipo



Fonte: Autor

4 RESULTADOS E DISCUSSÕES

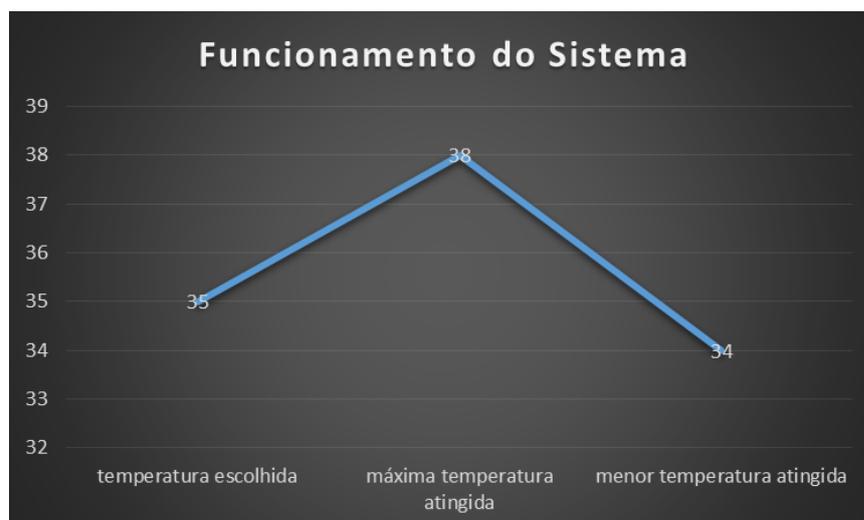
Visando facilitar o acesso do usuário ao sistema foi confeccionada uma etiqueta para a parte frontal do controle eletrônico, com menu informativo e indicação das teclas com suas funções correspondentes. A figura 16 mostra o protótipo instalado, etiquetado com as instruções e pronto para testes.

Figura 16: Protótipo Etiquetado com Instruções de Uso



Fonte: Autor

Após a instalação do protótipo foram realizados vários testes cujos resultados são apresentados na figura 17, onde comprovou-se o funcionamento de todas as funções implementadas no controle eletrônico, que respondeu satisfatoriamente ao objetivo proposto. O controle se mostrou eficaz no acionamento da resistência elétrica, cumprindo seu papel de aquecer a água da tubulação vinda do aquecedor, evitando seu descarte. O acionamento das eletroválvulas foi executado com eficiência mantendo a água do reservatório do chuveiro dentro dos limites aceitáveis de temperatura de acordo com a escolha do usuário.

Figura 17: Resposta do Sistema

Fonte: Autor

Os testes mostraram que ao ser selecionada a temperatura de 35 graus o sistema apresenta uma oscilação entre 38 e 34 graus, o que representa uma taxa aproximada de 8,5% acima do tempo escolhido, com um tempo de resposta de 10 segundos entre a temperatura escolhida e a temperatura máxima.

Com a implantação do controle automatizado para o chuveiro deixarão de ser descartados em média 1.200 litros de água por mês, o que representa uma economia de 14,4 m^3 de água em um ano. Ressaltando-se que neste caso trata-se de água limpa, potável e tratada.

4.1 PROBLEMAS/DIFICULDADES ENCONTRADAS

Para a construção e instalação do protótipo algumas dificuldades foram encontradas tendo que ser contornadas para que o projeto fosse concluído.

Na montagem do protótipo pretendia-se utilizar um relé de estado sólido para acionamento da resistência elétrica, por ser um componente utilizado para acionar cargas com elevadas correntes e por ter algumas vantagens sobre os contatores, como vida útil maior por não ter partes mecânicas e não sofrer desgaste com suas operações, possui preço mais acessível que os contatores e menor tempo de resposta além de ser completamente silencioso (DO BIT AO BYTE, 2017). No entanto o referido relé foi adquirido junto a um fornecedor da região norte do país, foi utilizado na montagem do protótipo porém não funcionou. Como não havia tempo hábil para solicitar a substituição da peça ou comprar outra por ser um tipo de componente não disponível em nossa região, o projeto teve que ser modificado sendo utilizada para a função de acionamento da carga resistiva um contator usado em boas condições, sem custos para o projeto.

Na instalação do protótipo algumas dificuldades também foram enfrentadas para adaptar o circuito hidráulico à instalação já existente no local. Para evitar que fosse necessário danificar

a alvenaria e acabamento do banheiro foram confeccionadas algumas peças para interligar as tubulações.

4.2 SUGESTÕES PARA TRABALHOS FUTUROS

O protótipo construído pode ser aperfeiçoado com a construção de um controle com melhorias na montagem dos componentes e construção de um circuito mais compacto e eficiente a um custo mais baixo que o do protótipo.

Como propostas para trabalhos futuros destacam-se:

- Substituir o atual contator por um relé de estado sólido pelas vantagens que este possui sobre o contator, citadas anteriormente.
- Implementar um sensor de presença para que seja possível uma pausa durante o banho se for desejo do usuário. Se for percebida uma determinada distância do usuário em relação ao chuveiro, o sistema desliga pausando o tempo do banho. Ao detectar a aproximação do usuário o sistema é religado dando continuidade ao tempo de banho.
- Implementar a função para informar o consumo de água no final do banho.
- Substituir as eletroválvulas por outras com vazão proporcional e aplicar métodos de controle visando reduzir ou eliminar as oscilações do sistema.

5 CONCLUSÃO

Em tempos de crise hídrica, em que a escassez de água se torna cada vez mais ameaçadora, ações sustentáveis e atitudes preservacionistas precisam ser exercidas assiduamente em todas as frentes, para garantia de nossa sobrevivência. A água é um recurso natural finito e quanto mais necessitamos dele, menos disponível ele se torna.

Nesse sentido pode-se afirmar que esse trabalho cumpriu seu objetivo principal, o de evitar o desperdício de água. Pode-se concluir que a elaboração do estudo de caso com a construção e utilização do protótipo teve um resultado positivo, com um produto simples e prático. As respostas dadas pelo sistema corresponderam às expectativas iniciais do projeto.

O trabalho comprovou a eficiência do Arduino para aplicação em protótipos e sistemas de comando deixando evidentes os benefícios, citados anteriormente, que essa plataforma oferece.

Além de resolver o problema do desperdício de água a implantação do controle eletrônico agregou outros benefícios como o conforto e segurança para o usuário com ganho de tempo devido à facilidade e rapidez no acesso ao sistema. Outro benefício foi a implementação do tempo predeterminado para o banho, que permite ao usuário monitorar indiretamente o consumo de água, evitando desperdícios.

6 REFERÊNCIAS BIBLIOGRÁFICAS

- ARDUINO. *O que é Arduino?* 2017. Disponível em: <<https://www.arduino.cc/en/Guide/Introduction>>. Acesso em: 08 out. 2017.
- BORGES, F. C. D. et al. *Automação*. Escola SENAI. Módulos Especiais Mecânica Telecurso 2000 Profissionalizante. São Paulo: [s.n.], [2017]. Não paginado. Disponível em: <<http://bmalbert.yolasite.com/resources/automa>
- BORTOLUZZI, M. *Histórico da Automação Residencial*. SRA Engenharia Blog. 2013. Disponível em: <http://sraengenharia.blogspot.com.br/2013/01/historico-da-automacao-residencial_10.html#!/2013/01/historico-da-automacao-residencial_10.html>. Acesso em: 04 out. 2017.
- BRAGA, N. C. *O básico sobre os Microcontroladores*. 2014. Disponível em: <<http://www.newtoncbraga.com.br/index.php/electronica/52-artigos-diversos/13263-o-basico-sobre-os-microcontroladores-parte-1-mic139>>. Acesso em: 04 out 2017.
- BRAZ, J. *Eletroválvula*. Eletro Física Blog. 04 abr. 2013. Disponível em: <<http://eletrofisica1.blogspot.com.br/2013/04/eletrovalvula.html>>. Acesso em: 09 out. 2017.
- COPASA. *Nota Fiscal/Fatura de Serviços*. 2017. Disponível em: <<http://www2.copasa.com.br/servicos/FaturaPorEmail/SegViaFatura.aspx?matr=00136603696/>>. Acesso em: 25 nov. 2017.
- DIAS, H. *Chuveiro híbrido de elétrico e solar é a opção mais barata e ecológica*. 2009. Disponível em: <<http://casa.abril.com.br/materiais-construcao/chuveiro-hibrido-de-eletrico-e-solar-e-a-opcao-mais-barata-e-ecologica/>>. Acesso em: 10 mai. 2017.
- DO BIT AO BYTE. *Relé de estado sólido FOTEK SSR-40 DA com Arduino, ESP8266, Raspberry...* 2017. Disponível em: <<http://www.dobitaobyte.com.br/rele-de-estado-solido-fotek-ssr-40-da-com-arduino-esp8266-raspberry/>>. Acesso em: 19 nov. 2017.
- GAIER, M. B. *Aprendendo a Programar em Arduino*. Cuiabá: IFMT/DAI, 2011. 49 p. Apostila. Disponível em: <<https://pt.slideshare.net/Miojex360/apostila-para-programar-arduino>>. Acesso em: 08 out. 2017.
- INVESPARK. *Maiores vilões de consumo de energia dentro de casa*. 2015. Disponível em: <<http://www.invespark.com.br/blog/maiores-viloes-de-consumo-de-energia-dentro-de-casa/>>. Acesso em: 05 nov. 2017.
- LORENZETTI. *Ducha Top Jet Eletrônica*. 2015. Disponível em: <http://www.lorenzetti.com.br/pt/Detalhes_Produto.aspx?id=1066>. Acesso em: 25 nov. 2017.
- MONK, S. *Programação com Arduino: começando com sketches*. 2. ed. Porto Alegre: Bookman, 2017. Disponível em: <<http://pensecomigo.com.br/livro-programacao-com-arduino-comecando-com-sketches-pdf-simon-monk/>>. Acesso em: 08 out. 2017.

- MUNDO DA ELÉTRICA. *Contatores, o que são?* 2017. Disponível em: <<https://www.mundodaeletrica.com.br/contatores-o-que-sao/>>. Acesso em: 25 nov. 2017.
- OGATA, K. *Engenharia de Controle Moderno*. 5. ed. Rio de Janeiro: Editora Pearson, 2011.
- PALMIERE, S. E. *CLP versus Microcontrolador*. 2016. Disponível em: <<https://www.embarcados.com.br/clp-versus-microcontrolador/>>. Acesso em: 04 out. 2017.
- SOLETROL. *Como Funciona o Aquecedor Solar de Água Soletrol*. 2017. Disponível em: <<http://www.soletrol.com.br/extras/como-funciona-o-aquecedor-solar-soletrol/>>. Acesso em: 26 mar. 2017.
- SOUZA, F. *Arduino - Primeiros Passos*. 2013. Disponível em: <https://www.embarcados.com.br/arduino-primeiros-passos/>. Acesso em: 25 out. 2017.
- USINAINFO. *Sensor de Temperatura DS18B20 à Prova D'Água*. 2017. Disponível em: <<https://www.usinainfo.com.br/sensor-de-temperatura-arduino/sensor-de-temperatura-ds18b20-a-prova-d-agua-2645.html>>. Acesso em: 17 jul. 2017.

APÊNDICE A PROGRAMAÇÃO DO SISTEMA

Desenvolvimento da programação de autoria própria

```

include <DallasTemperature.h>//biblioteca para controle da temperatura
#include <Wire.h>//biblioteca para controle da temperatura que o sensor esta medindo.
#include <LiquidCrystal_I2C.h>//biblioteca para utilizacao display 16x2 com modulo I2C.
#include <Keypad.h>//biblioteca para controle do teclado de membrana.,,0

// Sensores de entrada
float SensorTemp01 = 0; // variavel global que recebe o valor do sensor 1
float SensorTemp02 = 0; // variavel global que recebe o valor do sensor 2

// Dispositivos de saida
int Chuveiro=2; // porta digital que controla as acoes referentes a resistencia.
int ValvulaQuente=3; // porta digital que controla as acoes referentes a agua quente.
int ValvulaFria=12; // porta digital que controla as acoes referentes a agua fria.
int Buzzer=A2; // porta digital que controla as acoes referentes ao buzzer.

// Variaveis de controle de tempo
int TemperaturaBanho = 35; // variavel de controle dos minutos de banho.
int TempoBanho = 10; // variavel de exibicao de tempo.
int seg = 0, min = 0, hor = 0;

// millis no lugar delay
unsigned long tBase = 0; // recebe valor inicial
int tInt = 1000; // tempo a ser subtraido

// Variaveis de controle
int AtivaSistema = 0;
int ConRegBSair = 5;

// exibicao das telas
char state = 1; // variavel que guarda posicao atual do menu
int OrigemMenu = 1;

// Configuracao dos sensores DS18B20
OneWire oneWire(14); //Definindo a porta para os sensores
DallasTemperature sensors(&oneWire); //endereço de memoria que auxiliara o sensor.
DeviceAddress address2 = {0x28, 0xFF, 0x2C, 0xA0, 0x94, 0x16, 0x5, 0xF7}; //End. S1
DeviceAddress address1 = {0x28, 0xFF, 0xED, 0x93, 0x94, 0x16, 0x5, 0xD8}; //End. S2

// Configuracao do teclado
const byte LINHAS = 4; //Linhas do teclado definido pelos contatos 1,2,3 e 4.
const byte COLUNAS = 4; //coluna do teclado definido pelos contatos 5,6,7 e 8.

//Matriz de caracteres , define as cordenadas linhas x Colunas.
char matriz_teclas[LINHAS][COLUNAS] =
{
  { '1', '2', '3', 'A' },
  { '4', '5', '6', 'B' },
  { '7', '8', '9', 'C' },

```

```

    {'*', '0', '#', 'D'},
};

byte PinosLinhas[LINHAS] = {4, 5, 6, 7}; //Definicao dos pinos das linhas
byte PinosColunas[COLUNAS] = {8, 9, 10,11}; //Definicao dos pinos das colunas

Keypad teclado = Keypad( makeKeymap(matriz_teclas), PinosLinhas,
                        PinosColunas, LINHAS, COLUNAS);

LiquidCrystal_I2C lcd(0x27,2,1,0,4,5,6,7,3, POSITIVE);//caminho trilhado p/ativar display.

void setup() {

    lcd.begin(16,2);                // inicia o tipo de display
    lcd.setBacklight(HIGH);

    // splash
    lcd.setCursor(0,0);            // posiciona a primeira mensagem
    lcd.print("CHUVEIRO AUTOMAT"); // mensagem exibida na primeira coluna
    lcd.setCursor(0,1);           // posiciona a segunda mensagem
    lcd.print("TCC 2 – WEMERSON"); // mensagem exibida na segunda coluna
    delay(2000);                  // mostra a mensagem por 2 segundos

    // Configura o status das portas digitais
    pinMode( Chuveiro ,OUTPUT);    // definine esta porta como saida
    pinMode( ValvulaQuente ,OUTPUT); // definine esta porta como saida
    pinMode( ValvulaFria ,OUTPUT); // definine esta porta como saida
    pinMode( Buzzer ,OUTPUT);     // definine esta porta como saida
    digitalWrite( Chuveiro ,HIGH); // define o tipo de porta e o estado de saida
    digitalWrite( ValvulaQuente ,HIGH); // defini o tipo de porta e o estado de saida
    digitalWrite( ValvulaFria ,HIGH); // defini o tipo de porta e o estado de saida
    digitalWrite( Buzzer ,LOW);   // defini o tipo de porta e o estado de saida
}

void loop() {

    sensors.requestTemperatures(); // Envia o comando para obter temperaturas
    SensorTemp01= sensors.getTempC(address1); // Obter a temperatura em graus do sensor 1
    SensorTemp02= sensors.getTempC(address2); // Obter a temperatura em graus do sensor 2

    char teclaPressionada = teclado.getKey(); // Leitura do teclado

    switch (state) {                // Switch que define a posicao das telas
    case 1:                          // Na variavel global "state = 1"
        switch (teclaPressionada) { // Switch que recebe valores do teclado
            case 'A':                // Se for pressionado a tecla "A" faca.
                if (TemperaturaBanho<25 || TemperaturaBanho>50) // Faz a comparacao da temp.
                {
                    // Exibe mensagem de erro quando a temp. nao estiver dentro das especific.
                    tone(Buzzer, 1500, 400); // Faz beep
                    lcd.setCursor(0,0);
                    lcd.print(" > TEMPERATURA <");
                    lcd.setCursor(0,1);
                    lcd.print(" > INVALIDA <");
                    delay(2000); // Trava a mensagem por 2s para que usuario veja que errou
                }
            }
        }
    }
}

```

```

else // ou
{
    OrigemMenu = 1;           // Define de onde vai vim o menu
    tone(Buzzer, 1500, 50); // Faz beep
    Set_state(2);           // Se "A", pos. menu "Set_state(2);" e pos.tela 2
}
break;                       // Pausa no case
case '#':                   // Aciona direto eletrovalvula agua fria
    OrigemMenu = 2;         // Define de onde vai vim o menu
    tone(Buzzer, 1500, 50); // Faz beep
    Set_state(2);           // Se "#", pos. menu "Set_state(2);" e pos.tela 2

break;

case '*':                   // Aciona direto eletrovalvula agua quente
    OrigemMenu = 3;         // Define de onde vai vim o menu
    tone(Buzzer, 1500, 50); // Faz beep
    Set_state(2);           // Se "*", pos. menu "Set_state(2);" e pos.tela 2
break;
default:
    if (teclaPressionada){
        tone(Buzzer, 1500, 50); // Faz beep
        if(
            teclaPressionada == '0' ||
            teclaPressionada == '1' ||
            teclaPressionada == '2' ||
            teclaPressionada == '3' ||
            teclaPressionada == '4' || //verificacao de qual tecla foi pressionada.
            teclaPressionada == '5' ||
            teclaPressionada == '6' ||
            teclaPressionada == '7' ||
            teclaPressionada == '8' ||
            teclaPressionada == '9'
        )
        {
            if (String(TemperaturaBanho).length() > 1){
                lcd.setCursor(10,1);
                TemperaturaBanho = 0;
            }
            else
                lcd.setCursor(10,1);
            TemperaturaBanho = TemperaturaBanho * 10 + (teclaPressionada - 0x30);
            lcd.print(TemperaturaBanho);
            lcd.write(223);
        }
    }
    desligatudo();
    Set_state(1);           // Mantem a posicao 1
}
break;
case 2:
    switch (teclaPressionada) {
    case 'A':
        if (TempoBanho > 20)
        {
            tone(Buzzer, 1500, 400);
            lcd.setCursor(0,0);

```

```

        lcd.print(">TEMPO DE BANHO<");
        lcd.setCursor(0,1);
        lcd.print(">  INVALIDO  <");
        delay(2000);
    }
    else
    {
        tone(Buzzer, 1500, 100);
        Set_state(3);
        min = TempoBanho - 1;
    }
    break;
case 'B':
    OrigemMenu = 1;
    tone(Buzzer, 1500, 50);
    Set_state(1);
    break;
default:
    if (teclaPressionada){
        tone(Buzzer, 1500, 50);          // FAZ BEEP
        if(
            teclaPressionada == '0' ||
            teclaPressionada == '1' ||
            teclaPressionada == '2' ||
            teclaPressionada == '3' ||
            teclaPressionada == '4' || //verificacao de qual tecla foi pressionada.
            teclaPressionada == '5' ||
            teclaPressionada == '6' ||
            teclaPressionada == '7' ||
            teclaPressionada == '8' ||
            teclaPressionada == '9'
        )
        {
            if (String(TempoBanho).length()> 1){
                lcd.setCursor(10,1);
                TempoBanho = 0;
            }
            else
            lcd.setCursor(10,1);
            TempoBanho = TempoBanho * 10 + (teclaPressionada - 0x30);
            lcd.print(TempoBanho);
            lcd.print("m");
        }
    }
    Set_state(2);
}
break;
case 3: //MONITOR DE BANHO
switch (teclaPressionada) {
    case 'C':
        // tone(Buzzer, 1500, 50);
        break;
    case 'B':
        tone(Buzzer, 1500, 50);
        Set_state(2);
        break;
    case 'D':

```

```

        tone(Buzzer, 1500, 50);
        Set_state(5);
    break;
    default:
        Set_state(3);
    }
    break;
case 4: //FIM DE BANHO
    switch (teclaPressionada) {
        break;
        default:
            Set_state(4);
    }
    break;
case 5: //MONITOR DE BANHO
    switch (teclaPressionada) {
        case 'D':
            tone(Buzzer, 1500, 400);
            seg = 0;
            AtivaSistema = 0;
            TempoBanho = 10;
            TemperaturaBanho = 35;
            min = TempoBanho;
            Set_state(4);
        break;
        default:
            if (ConRegBSair == 0){
                Set_state(3);
                ConRegBSair = 5;
            }
            static unsigned long ult_tempo = 0;
            int tempo = millis();
            if (tempo - ult_tempo >= 1000) {
                ult_tempo = tempo;
                ConRegBSair--;
            }
            break;
            Set_state(5);
        }
        break;
    default: ;
    }
}

void Set_state(char index) {
    state = index; // Atualiza a variavel state para a nova tela
    switch (state) { // verifica qual a tela atual e exibe o conteudo correspondente
        case 1: //===== state 1
            lcd.setCursor(0,0); //MENU DO CASE 1
            lcd.print("DIGITE TEMP. OU ");
            lcd.setCursor(0,1);
            lcd.print("CONFIRME:");
            lcd.print(TemperaturaBanho);
            lcd.write(223);
            lcd.print(" > A");

            break;

```

```

case 2: //===== state 2
    lcd.setCursor(0,0);
    lcd.print("DIGITE TEMPO OU ");
    lcd.setCursor(0,1);
    lcd.print("CONFIRME:");
    lcd.print(TempoBanho);
    lcd.print("m");
    lcd.print(" > A");
    break;

case 3: //===== state 3
    if (min < 1 && seg <= 60){
        Sinalizador();
    }
    else
    {
        lcd.setBacklight(HIGH);
    }
    CalcTempDes(); // Chama funcao CalcTemDes
    if ((millis() - tBase) >= tInt) // trocando delay por millis, 1s de espera
    {
        tBase = millis(); // Atualiza tBase
        lcd.setCursor(0,0); // Posiciona o texto no LCD
        lcd.print("AQUEC.:"); // Imprime o texto no LCD
        long xpto1 = (long)(SensorTemp01); // Arredonda os valores em 2 casas decimais.
        lcd.print(xpto1); // Imprime o valor da variavel ja corrigido.
        lcd.write(223); // Imprime no LCD o caracter de graus
        lcd.setCursor(0,1); // Posiciona o texto no LCD
        lcd.print("CHUV. :"); // Imprime o texto no LCD
        long xpto2 = (long)(SensorTemp02); // Arredonda os valores em 2 casas decimais.
        lcd.print(xpto2); // Imprime o valor da variavel ja corrigido.
        lcd.write(223); // Imprime no LCD o caracter de graus
    }
    lcd.setCursor(10,0);
    lcd.print("l");
    lcd.setCursor(11,0);
    lcd.print("TEMPO");
    lcd.setCursor(10,1);
    lcd.print("l");
    if (OrigemMenu == 1){
        Opcao1();
    }
    else
    if (OrigemMenu == 2){
        Opcao2();
    }
    else
    if (OrigemMenu == 3){
        Opcao3();
    }
    break;

case 4: //===== state 4
    lcd.setCursor(0,0);
    lcd.print("> FIM DO BANHO <");
    lcd.setCursor(0,1);
    lcd.print(">>>>>>>><<<<<<<<");

```

```
        delay(5000);
        lcd.clear();
        tone(Buzzer, 1500, 50);
        OrigemMenu = 1;
        Set_state(1);
        break;

        case 5: //===== state 5
        lcd.setCursor(0,0);
        lcd.print("TERMINAR O BANHO");
        lcd.setCursor(0,1);
        lcd.print("CONFIRMAR: > D");
        break;
        default: ;
    }
}
void Opcao1()
{
    long xpto1 = (long)(SensorTemp01);
    if (xpto1 < TemperaturaBanho)
    {
        digitalWrite(Chuveiro,LOW);
        digitalWrite(ValvulaQuente,LOW);
        digitalWrite(ValvulaFria,HIGH);
    }
    else
    if (xpto1 == TemperaturaBanho)
    {
        digitalWrite(Chuveiro,HIGH);
        digitalWrite(ValvulaQuente,LOW);
        digitalWrite(ValvulaFria,HIGH);
    }
    else
    if (xpto1 >= TemperaturaBanho +1)
    {
        digitalWrite(Chuveiro,HIGH);
        digitalWrite(ValvulaQuente,HIGH);
        digitalWrite(ValvulaFria,LOW);
    }
}
void Opcao2()
{
    digitalWrite(Chuveiro,HIGH);
    digitalWrite(ValvulaQuente,LOW);
    digitalWrite(ValvulaFria,HIGH);
}
void Opcao3()
{
    digitalWrite(Chuveiro,HIGH);
    digitalWrite(ValvulaQuente,HIGH);
    digitalWrite(ValvulaFria,LOW);
}
void DelayMillis(){
    static unsigned long ult_tempo = 0;
    int tempo = millis();
    if (tempo - ult_tempo >= 1000) {
        ult_tempo = tempo;
    }
}
```

```

    }
}

void desligatudo ()
{
    digitalWrite (ValvulaQuente ,HIGH);
    digitalWrite (ValvulaFria ,HIGH);
    digitalWrite (Chuveiro ,HIGH);
}

void CalcTempDes () // define valor da variavel str1
{
    if (min >=0) // Faz comparacao da temp. digitada com valores predefinidos.
    {
        AtivaSistema = 1;
        static unsigned long ult_tempo = 0;
        int tempo = millis ();
        if (tempo - ult_tempo >= 1000) {
            ult_tempo = tempo;
            seg--;
        }
        if (seg <= 0) {
            seg = 59; min--;
        }
        if (min >= 60) {
            min = 0;
        }
    }
    else{
        seg = 0;
        AtivaSistema = 0;
        TempoBanho = 10;
        TemperaturaBanho = 35;
        min = TempoBanho;
        tone(Buzzer , 1500, 400);
        desligatudo ();
        Set_state (4);
    }
    showDisp(min , seg);
}

void Sinalizador (){
    lcd.setBacklight (HIGH);
    static unsigned long ult_tempo = 0;
    int tempo = millis ();
    if (tempo - ult_tempo >= 1000) {
        ult_tempo = tempo;
        lcd.setBacklight (LOW);
        tone(Buzzer , 1500, 50);
    }
}

void showDisp(short minutos , short segundos) //Funcao para exhibir corretamente
{
    if(minutos < 10) //minutos menor que 10?
    { //Sim...
        lcd.setCursor (11,1);
        lcd.print ('0'); //Posiciona cursor na coluna 12, linha 2 e imprime o '0'
        lcd.setCursor (12,1);
    }
}

```

```
    lcd.print(minutos);      //Posiciona cursor na coluna 13, linha 2 e imprime min.
} //end if minutos
else                          //senao ...
{                              //minutos igual ou maior que 10
    lcd.setCursor(11,1);
    lcd.print(minutos);      //Posiciona cursor na coluna 12, linha 2 e imprime min.
} //end else minutos

    lcd.setCursor(13,1);
    lcd.print(':');          //Posiciona cursor na coluna 14, linha 2 e imprime ':'

if(segundos < 10)             //segundos menor que 10?
{                              //Sim...
    lcd.setCursor(14,1);
    lcd.print('0');          //Posiciona cursor na coluna 15, linha 2 e imprime o '0'
    lcd.setCursor(15,1);
    lcd.print(segundos);     //Posiciona cursor na coluna 14, linha 2 e imprime seg.
} //end if segundos
else                          //senao ...
{                              //segundos igual ou maior que 10
    lcd.setCursor(14,1);
    lcd.print(segundos);     //Posiciona cursor na coluna 15, linha 2 e imprime seg.
} //end else segundos

} //end showDisp
```