

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ  
DEPARTAMENTO DE ELETRÔNICA  
ENGENHARIA ELETRÔNICA**

**EDUARDO VIEIRA**

**DESENVOLVIMENTO DE UM PROTÓTIPO PARA O CONTROLE DO  
PROCESSO DE BRASSAGEM NA FABRICAÇÃO DE CERVEJA  
ARTESANAL**

**TRABALHO DE CONCLUSÃO DE CURSO**

**PONTA GROSSA**

**2017**

**EDUARDO VIEIRA**

**DESENVOLVIMENTO DE UM PROTÓTIPO PARA O CONTROLE DO  
PROCESSO DE BRASSAGEM NA FABRICAÇÃO DE CERVEJA  
ARTESANAL**

Trabalho de Conclusão de Curso apresentado como requisito parcial à obtenção do título de Bacharel em Engenharia Eletrônica, do Departamento de Eletrônica, da Universidade Tecnológica Federal do Paraná.

Orientador: Prof. Dr. Frederic Conrad Janzen

**PONTA GROSSA**

**2017**



Ministério da Educação  
**Universidade Tecnológica Federal do Paraná**  
Câmpus Ponta Grossa  
DAELE – Departamento de Eletrônica



---

## FOLHA DE APROVAÇÃO

### DESENVOLVIMENTO DE UM PROTÓTIPO PARA O CONTROLE DO PROCESSO DE BRASSAGEM NA FABRICAÇÃO DE CERVEJA ARTESANAL

Desenvolvido por:

EDUARDO VIEIRA

Este trabalho de conclusão de curso foi apresentado em 24 de novembro de 2017, como requisito parcial para obtenção do título de Engenheiro Eletrônico. O candidato foi arguido pela banca examinadora composta pelos professores abaixo assinado. Após deliberação, a Banca Examinadora considerou o trabalho aprovado.

---

Dr. Frederic Conrad Janzen  
Professor Orientador

---

Dr. Josmar Ivanqui  
Membro titular

---

Ms. Jeferson José Gomes  
Membro titular

- A Folha de Aprovação assinada encontra-se arquivada na Secretaria Acadêmica -

"A maior recompensa para o trabalho do  
homem não é o que ele ganha com isto,  
mas o que ele se torna com isto."  
(John Ruskin)

## **AGRADECIMENTOS**

Agradeço ao professor Dr. Frederic Conrad Janzen por proporcionar o desenvolvimento e conclusão deste trabalho, por dar todo o apoio e incentivo necessário sendo paciente e compreensível nos momentos de dificuldade.

A todos os meus familiares em especial a minha mãe Maria Cenir Guedes Vieira, minhas irmãs Joiceni Moreira Giaretta, Patricia Moreira Weis e Isabella Brandão de Queiroz pela educação, perseverança e aprendizado que me proporcionaram ao longo de toda a vida.

Ao meu pai Clarettes Vieira (in memorian) que apesar da ausência deixou um legado que contribuiu diretamente para minha educação e honestidade.

Aos meus amigos e companheiros que sempre estiveram me desejando o melhor e aos que de alguma forma contribuíram para minha formação pessoal, intelectual e humana.

Aos professores e funcionários da UTFPR-PG que contribuíram para a minha formação em toda a jornada acadêmica.

E por fim, a Deus por me proporcionar a luz que me permitiu a utilização das ferramentas necessárias para minha autoconstrução.

## RESUMO

VIEIRA, EDUARDO. **Desenvolvimento de um protótipo para o controle do processo de brassagem na fabricação de cerveja artesanal.** 2017. 124 f. Trabalho de Conclusão de Curso (Bacharel em Engenharia Eletrônica) - Universidade Tecnológica Federal do Paraná. Ponta Grossa, 2017

O presente trabalho de conclusão de curso trata-se do desenvolvimento de um protótipo para o controle do processo de brassagem na fabricação de cerveja artesanal. Em função do crescimento do mercado cervejeiro por cervejas especiais, é o que motivou o desenvolvimento deste trabalho que se compõe primeiramente do estudo detalhado das etapas individuais do processo de fabricação de cerveja caseira, e após isso o estudo dos elementos necessários ao desenvolvimento de um protótipo para o controle de uma planta cervejeira. Concebido tais conceitos o projeto foi desenvolvido com base na plataforma microcontrolada Arduino Mega 2560, juntamente com periféricos para fazer o controle e monitoramento das variáveis do processo cervejeiro. Dessa forma o projeto foi executado e seus resultados apresentados.

**Palavras-chave:** Cerveja artesanal. Arduino Mega 2560. Controle de processo.

## **ABSTRACT**

VIEIRA, EDUARDO. **Development of a prototype for the control of the craft brewing process.** 2017. 124 f. Work of Conclusion Course (Bachelor of Electronic Engineering) - Federal Technology University - Paraná. Ponta Grossa, 2017.

This course conclusion work it is the development of a prototype to control the brewing process in the manufacture of craft beer. Due to the growth of the beer market by special beers, this motivated the development of this work, which consists first of the detailed study of the individual steps of the homemade brewing process, and after that the study of the elements necessary to the development of a prototype to control a brewing plant. Conceived of such concepts the project was developed based on the microcontrolled platform Arduino Mega 2560, along with peripherals to make the control and monitoring of the variables of the brewing process. In this way the project was executed and its results presented.

**Keywords:** Craft beer. Arduino Mega 2560. Process control.

## LISTA DE ILUSTRAÇÕES

Figura 1 – Saco de Grãos .....	17
Figura 2 – Cadeira de mostura (à direita) e fundo falso (à esquerda).....	18
Figura 3 – Rótulo da cerveja Eisenbahn estilo Pale Ale .....	21
Figura 4 – Processo Cervejeiro.....	22
Figura 5 – Moinho de Disco.....	23
Figura 6 – Escala das enzimas presentes na brassagem .....	24
Figura 7 – Filtragem e clarificação.....	26
Figura 8 – Chiller de imersão.....	28
Figura 9 – Chiller de placas .....	29
Figura 10 – Sistema de recirculação contínua .....	32
Figura 11 – Plataforma Arduino Mega 2560 .....	33
Figura 12 – Módulo LCD .....	36
Figura 13 – Esquemático Rele .....	39
Figura 14 – Módulo rele 8 canais.....	40
Figura 15 – Esquemático eletrônico de 1 canal do modulo rele .....	40
Figura 16 – Sensor Ultrassônico HC-SR04.....	41
Figura 17 – Sistema de medição de nível .....	42
Figura 18 – Válvula solenoide de água 2 vias normalmente fechada .....	43
Figura 19 – Válvula solenoide de gás .....	45
Figura 20 – Sensor de chama Arduino .....	46
Figura 21 – Módulo sensor de chama Arduino.....	46
Figura 22 – Módulo relógio em tempo real.....	48
Figura 23 – Software de programação da plataforma Arduino .....	49
Figura 24 – Válvula tipo esfera em aço inox.....	51
Figura 25 – Esboço do fogão cervejeiro .....	55
Figura 26 – Vista frontal da disposição física dos caldeirões.....	56
Figura 27 – Fogão cervejeiro.....	56
Figura 28 - Sistema de aquecimento com chama piloto .....	57
Figura 29 - Planta cervejeira.....	58
Figura 30 - Simbologia da planta cervejeira.....	59
Figura 31 - Circuito eletrônico da IHM .....	62
Figura 32 - Verso da IHM .....	63
Figura 33 - Frente da IHM.....	63
Figura 34 - Circuito do Painel .....	65
Figura 35 - Montagem do painel com gabarito .....	68
Figura 36 - Cortes, furos e fixação dos componentes do painel.....	68
Figura 37 - Cabeamento do painel .....	69
Figura 38 - Painel final.....	69
Figura 39 - Seleção das variáveis do processo.....	70
Figura 40 - Painel de controle em funcionamento.....	75



## LISTA DE TABELAS

Tabela 1 – Características principais do Arduino Mega 2560.....	34
Tabela 2 – Pinagem do modulo display LCD.....	37
Tabela 3 – Custos totais do sistema .....	76

## LISTA DE SIGLAS

BIAB	Brew in a bag
PIB	Produto Interno Bruto
USB	Universal Serial Bus
PWM	Pulse Width Modulation
IHM	Interface Homem Máquina
LCD	Liquid Cristal Display
LED	Light Emissor Diode
RTC	Real Time Clock
I <sup>2</sup> C	Inter Integrated Circuit
SAE	Society of Automotive Engineers
INOX	Aço inoxidável
CI	Circuito Integrado

## LISTA DE SÍMBOLOS

V	Volts
mA	miliamperes
A	Amper
KB	Kilobyte
Ph	Potencial Hidrogênico
Khz	Kilohertz
Mhz	Megahertz
°C	Graus Celcius
Kgf	Quilograma força
cm <sup>2</sup>	Centímetro quadrado

## SUMÁRIO

<b>1 INTRODUÇÃO</b> .....	14
1.1 PROBLEMA .....	15
1.2 OBJETIVOS .....	15
1.2.1 Objetivos gerais .....	15
1.2.2 Objetivos específicos .....	16
1.3 JUSTIFICATIVA .....	16
1.3.1 “Grain Bag” ou Saco de Grão .....	16
1.3.2 Fundo falso .....	18
1.4 ESTRUTURA DO TRABALHO .....	19
<b>2. REFERENCIAL TEÓRICO</b> .....	20
2.1 HISTÓRICO DA CERVEJA .....	20
2.2 DETALHAMENTO DO PROCESSO CERVEJEIRO .....	21
2.2.1 Moagem do Grão .....	22
2.2.2 Mostura ou Brassagem .....	23
2.2.3 Filtragem e Clarificação .....	25
2.2.4 Fervura com adição de lúpulo .....	26
2.2.5 Resfriamento .....	27
2.2.6 Fermentação .....	29
2.2.7 Maturação .....	30
2.2.8 Envase .....	30
2.3 RELEVÂNCIA DO CONTROLE DE TEMPERATURA E LUPULAGEM NA BRASSAGEM .....	31
2.4 MÉTODO DE RECIRCULAÇÃO CONTÍNUA COM FUNDO FALSO .....	31
2.5 PLATAFORMA ARDUINO MEGA 2560 .....	33
2.6 SENSOR DE TEMPERATURA DS18B20 .....	35
2.7 IHM – INTERFACE HOMEM MÁQUINA .....	35
2.7.1 Módulo LCD .....	35
2.7.2 Botões de pulso .....	37
2.7.3 Sirene .....	38
2.8 MÓDULO RELE DE 8 CANAIS .....	38
2.9 SISTEMA DE MEDIÇÃO DE NÍVEL .....	41
2.10 VÁLVULAS SOLENOIDE .....	43
2.11 SENSOR DE CHAMA .....	45
2.12 MÓDULO RTC .....	47
2.13 SOFTWARE UTILIZADO NA PROGRAMAÇÃO .....	48

2.14 BOMBA CENTRÍFUGA .....	50
2.15 CONEXÕES, TUBULAÇÕES, VÁLVULAS E QUEIMADORES .....	51
2.16 ESTRUTURA FÍSICA DE SUSTENTAÇÃO .....	52
2.17 ESTRUTURA FÍSICA ELÉTRICA E ELETRÔNICA.....	52
<b>3. MATERIAIS E MÉTODOS .....</b>	<b>53</b>
<b>4. METODOLOGIA .....</b>	<b>54</b>
4.1 MONTAGEM DA ESTRUTURA FÍSICA DE SUSTENTAÇÃO .....	54
4.2 SISTEMA DE ACIONAMENTO DO QUEIMADOR .....	57
4.3 PLANTA CERVEJEIRA E PROCESSO DE FABRICAÇÃO .....	58
4.5 CIRCUITOS ELÉTRICOS DE POTÊNCIA, SUPERVISÃO E CONTROLE .....	64
4.5 PROGRAMA DESENVOLVIDO .....	70
4.5.1 Sistema de seleção dos valores do processo.....	70
4.5.2 Cálculo do nível de água.....	72
4.5.3 Cálculo dos tempos de rampa.....	72
4.5.4 Leitura de temperatura .....	73
4.5.6 Sistema de segurança da chama .....	74
4.5.7 Programação do processo em andamento .....	74
<b>5. RESULTADOS.....</b>	<b>75</b>
<b>6. CONCLUSÕES .....</b>	<b>78</b>
<b>7. SUGESTÕES PARA FUTUROS TRABALHOS.....</b>	<b>79</b>
<b>REFERÊNCIAS .....</b>	<b>80</b>
<b>APÊNDICE A – Programação.....</b>	<b>83</b>

## 1 INTRODUÇÃO

O país que mais consome cerveja no mundo proporcionalmente é a República Tcheca, tendo um consumo médio *per capita* de 147,1 litros por ano, seguido pela Namíbia, Áustria e Alemanha com consumos semelhantes na faixa dos 101 a 108 litros anuais por habitante. O Brasil está na vigésima sétima posição, com um consumo médio *per capita* de 66,9 litros, o que é significativamente grande e torna-o o terceiro maior produtor de cerveja do mundo, perdendo apenas para a China e para os Estados Unidos da América, respectivamente. (CERVBRASIL, 2015)

Entretanto, quando se trata das cervejas não industriais ou cervejas artesanais, segundo Sebrae (2014), estas representam apenas 0,15% do mercado total de cervejas, porém em 10 anos espera-se um crescimento para a faixa dos 2%. Dessa forma mostra-se que a tendência de crescimento e interesse por cervejas artesanais está aumentando.

Não só o consumo, mas a produção de cerveja caseira tende ao crescimento, como pode-se observar no surgimento de empresas específicas na venda de equipamentos e insumos para cervejeiros caseiros como a WE Consultoria®, ou até mesmo a micro cervejaria Bode Brown® que faz este tipo de trabalho em paralelo com a sua produção.

Palmer (2006) explica basicamente dois métodos de fabricação de cerveja caseira. Um deles é a utilização de uma substância denominada extrato de malte, que é diluído em água e posteriormente passa para o processo de fervura com adição de lúpulo, resfriamento, fermentação, maturação, envase e carbonatação. O segundo método conhecido como *allgrain*, refere-se ao método em que se utiliza grãos maltados que passam pela moagem, mostura, clarificação, filtragem, fervura com adição de lúpulo, resfriamento, fermentação, maturação, envase e carbonatação.

Assim, segundo Palmer (2006), é na mostura em que os controles de temperatura e tempo são relevantes para definir a qualidade e objetivo final da cerveja, tudo isso em função das ações enzimáticas que ocorrem quando o malte triturado está em meio aquoso.

Analisando o controle de temperatura por meio de um fogareiro e válvula de vazão de gás por exemplo, manualmente existem inconvenientes, já que esta ação depende exclusivamente da atuação humana. Além disso, o controle do tempo por meio de relógios pode ocasionar erros por falha humana o que pode resultar em desperdício de matéria prima e até necessidade de reinício de processo.

Para melhorar a precisão de temperatura e tempo é que se desenvolveu este trabalho, que se trata da criação de um protótipo para o controle semiautomático de uma planta cervejeira caseira com o intuito de melhorar a qualidade do produto final e facilitar o processo de produção por parte do usuário.

## 1.1 PROBLEMA

Quais as consequências quando se ultrapassa temperaturas na mostura de um processo cervejeiro? Será que a qualidade final do produto ou objetivo de cerveja a ser fabricada satisfazem o esperado nos processos cervejeiros artesanais? Os métodos manuais garantem a segurança de que se faça cerveja com especificações definidas em uma receita?

## 1.2 OBJETIVOS

### 1.2.1 Objetivos gerais

O objetivo geral deste trabalho é estudar o processo de fabricação de cervejas artesanais e identificar quais as etapas críticas que definirão a qualidade e objetivo final do produto. Em seguida, desenvolver um protótipo para o controle de brassagem utilizando os conceitos físicos de transferência de calor, instrumentação e controle, programação, execução de projeto, máquinas elétricas, eletrônica digital e analógica, dentre outros estudos necessários a automatização de parte do processo cervejeiro artesanal, a fim de facilitar a execução de uma boa fabricação de cerveja caseira.

### 1.2.2 Objetivos específicos

- Contextualizar cerveja e modos de fabricação artesanal existentes com o intuito de definir um modelo de planta para que se possa inserir o protótipo no controle da mesma.
- Descrever de forma detalhada o processo de fabricação de cerveja e suas respectivas etapas a fim de se obter conceitos para o desenvolvimento da planta em questão.
- Estudar os componentes necessários a confecção do protótipo para que se obtenha conceitos suficientes na montagem deste.
- Projetar e desenvolver o protótipo, aplicando-o no sistema a fim de verificar sua funcionalidade e possíveis melhoras.

### 1.3 JUSTIFICATIVA

Além do fato de as cervejas artesanais possuírem qualidade maior diante das cervejas produzidas em larga escala, e os atuais métodos de fabricação de cervejas artesanais apresentarem um determinado esforço conforme a escassez de equipamentos foi o que motivou o desenvolvimento deste trabalho.

Conforme colocado por Palmer 2006, os principais métodos de fabricação de cervejas artesanais do tipo *allgrain* são: *Grain Bag* (também conhecido por saco de grão) e fundo falso. A seguir estão descritos de forma geral como funciona cada um destes métodos.

#### 1.3.1 “*Grain Bag*” ou Saco de Grão

Também conhecido pela sigla BIAB (*Brew In a Bag*), o *Grain Bag* é um dos métodos mais simples para a fabricação de cerveja artesanal principalmente pela



simplicidade dos equipamentos, comentado por Palmer (2006). Basicamente, neste método utiliza-se um caldeirão com um filtro de tecido em seu interior onde ficam alocados os grãos de malte já moídos. A figura 1 representa um exemplo desta disposição caldeirão/filtro para melhor entendimento.

**Figura 1 – Saco de Grãos**



**Fonte: SMITH (2009)**

É neste formato que funciona o método do saco de grão no momento de mostura, que é seguido pela filtragem, onde o saco de grão é removido, ficando apenas o líquido gerado da mostura (mosto) no caldeirão. (BROUWLAND, 2017)

A etapa de remoção do saco de grão mostra-se um pouco perigosa já que o mosto está em uma temperatura na faixa de 67°C e o processo é manual, conforme o manual de utilização do produto *The Brew Bag* da Brouwland®.

Pode-se observar que o trabalho braçal neste método é maior e dificulta a tarefa quando realizado em grandes proporções de produção já que a remoção do saco de grãos será de forma manual.

### 1.3.2 Fundo falso

Segundo Palmer (2006), neste método utilizam-se basicamente 3 caldeirões, mostura/fervura, filtragem e enxague. No primeiro momento, é realizado no caldeirão de mostura/fervura a sacarificação dos grãos moídos em meio aquoso e, posteriormente a esta etapa, é realizada a filtragem. Este processo depende da transferência do mosto para o caldeirão de filtragem para a separação dos grãos moídos, deixando apenas o líquido que será posteriormente fervido no caldeirão de fervura. A figura 2 mostra um exemplo de caldeirão de mostura juntamente com um fundo falso para a filtragem.

**Figura 2 – Caldeirão de mostura (à direita) e fundo falso (à esquerda)**



**Fonte: Autoria própria**

Visto a figura 2, o caldeirão de filtragem tem o mesmo formato do caldeirão de fervura, contudo é alocado em seu interior o fundo falso com uma saída tipo torneira na parte inferior, permitindo a retenção dos grãos e passagem do líquido. Como pôde-se observar, a mostura ocorre em um caldeirão e passa para a filtragem, assim este mesmo caldeirão de mostura servirá como caldeirão de fervura após sua higienização e eliminação das pequenas partes sólidas que aderem em sua parede.

Existem outras formas de fundo falso utilizados, que são também chamados de bazuca, possuindo formatos variados e eficiências diversificadas para específicos objetivos conforme a publicação de Palmer (1995). Contudo estes formatos não serão abordados detalhadamente já que não faz parte do objetivo do trabalho.

Dessa forma, nota-se que o processo de transferência de grãos gera inconvenientes quando o processo é manual, visto a necessidade de utilização de uma caneca para auxiliar a etapa. Mas para a fabricação de maiores quantidades de cerveja não existe a necessidade de grande esforço braçal como apresentado pelo método do saco de grãos. Por isso é que se escolheu o método do fundo falso para o desenvolvimento do trabalho em questão, mas com algumas modificações que melhoram as condições de fabricação para o cervejeiro artesanal.

#### 1.4 ESTRUTURA DO TRABALHO

Este trabalho divide-se em grupos para um melhor entendimento do leitor. Primeiramente como já apresentado, tem-se uma introdução com objetivos do desenvolvimento e a justificativa do estudo. A partir deste ponto são apresentados os conhecimentos necessários para a compreensão do processo cervejeiro de forma detalhada seguido pelos conceitos em eletrônica e do sistema de controle semiautomático desenvolvido.

Em seguida são mostrados os métodos que se utilizaram para o desenvolvimento físico e não físico (programações) do trabalho e por fim serão expostos os resultados, juntamente com discussões, conclusões e melhoras para futuros estudos na área.

## 2. REFERENCIAL TEÓRICO

### 2.1 HISTÓRICO DA CERVEJA

Relatos sobre o surgimento e consumo de uma bebida fabricada a partir de grãos de cevada conhecida hoje como cerveja, existem há aproximadamente 5000 anos atrás entre os sumérios e egípcios (CERVESIA, 2017).

Durante a idade média a cerveja começou a ser produzida pelos monges dentro dos mosteiros, sendo servida para os hóspedes e também utilizada como alimento em épocas de jejum, onde só era permitida ingestão de líquidos (origem do apelido da cerveja de “pão líquido”). A partir do século 18, sua produção começou a ter um maior cunho científico, onde eram utilizadas tecnologias modernas, como o termômetro por exemplo (ARTWINE, 2011).

Atualmente a cerveja é responsável por 1,6% do PIB (Produto Interno Bruto) nacional e tem um recolhimento anual maior que 20 bilhões em tributos, e só no estado de São Paulo o mercado cervejeiro gera aproximadamente 2,2 milhões de empregos diretos, indiretos e induzidos, mostrando assim a sua importância econômica no cenário nacional (CERVBRASIL, 2015).

Segundo o artigo 36 do decreto 6.871 de 4 de junho de 2009, é considerado cerveja quaisquer bebidas provenientes da fermentação alcoólica do mosto por ação de leveduras. O mosto é contido por malte de cevada, água potável e adição de lúpulo. No inciso 4, a legislação permite o uso de até 45% de adjuntos cervejeiros na fabricação (BRASIL, 2009).

Contudo, tal decreto vai contra a lei da pureza alemã de 1516 (também conhecida como *Reinheitsgebot*), instituída com o intuito de padronizar a qualidade das cervejas, onde só é permitido a utilização de água, malte, lúpulo e levedura (CERVESIA, 2017).

Assim, observa-se que o uso de determinados adjuntos cervejeiros diferentes de grãos maltados estão presentes nas cervejas convencionais do padrão brasileiro, o que geralmente não ocorre na produção da cerveja artesanal. Algumas fábricas

seguem a lei da pureza, como a cerveja estilo *Pale Ale* da cervejaria Eisenbahn® visto em destaque (cor vermelha) na figura 3.

**Figura 3 – Rótulo da cerveja Eisenbahn estilo Pale Ale**



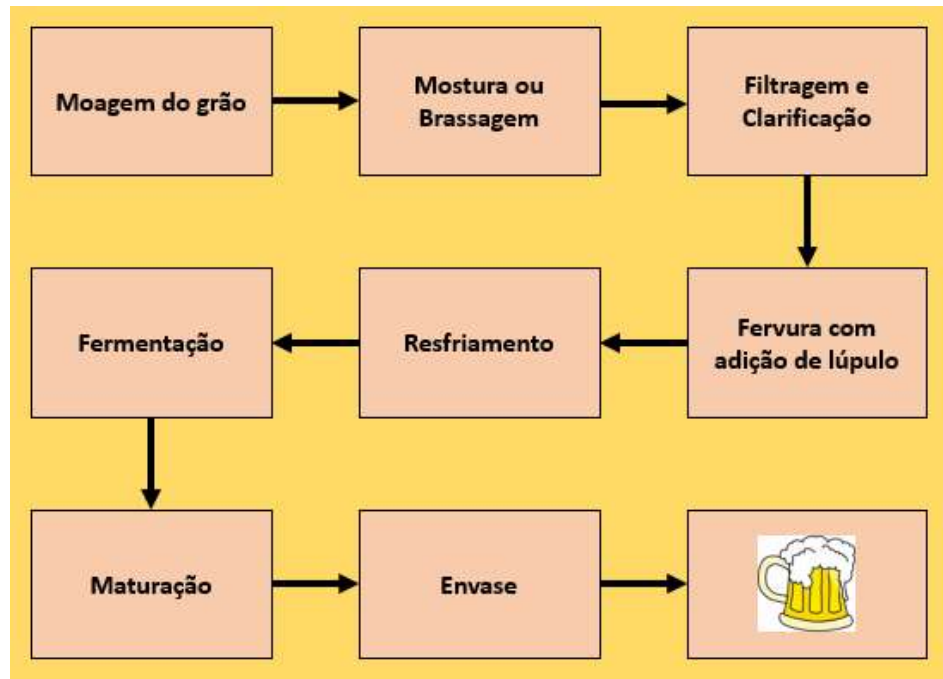
Fonte: Supercoleção (2014)

Dessa forma, o objetivo de seguir esta lei é a preservação da qualidade, justamente o que não ocorre nas grandes cervejarias brasileiras visto o decreto 6.871 de 2009. Essa questão é que desperta o interesse pela fabricação da própria cerveja, já que na produção caseira, permite-se a preservação da qualidade com adição dos melhores ingredientes, ficando livre de adjuntos e a gosto do fabricante, dando liberdade cervejeira ao cervejeiro artesanal.

## 2.2 DETALHAMENTO DO PROCESSO CERVEJEIRO

Segundo Palmer (2006), o processo de fabricação de cerveja é constituído basicamente das seguintes etapas: moagem do grão, mostura ou brassagem, filtragem e clarificação, fervura com adição de lúpulo, resfriamento, fermentação, maturação e envase. A figura 4 a seguir mostra um fluxograma básico do processo de fabricação da cerveja.

Figura 4 – Processo Cervejeiro



Fonte: Autoria própria

Vale salientar que existem processos industriais que estão sob sigilo, ou seja, não é possível ter acesso a determinadas etapas de fabricação já que estão estabelecidas como sigilo segundo a Lei N°10.603, de 17 de dezembro de 2002 (BRASIL, 2002). Com base nisso, vale salientar que o processo adotado neste trabalho respeita as etapas mostradas no fluxograma da figura 4 apresentada anteriormente.

Baseado neste fluxograma, segue a descrição mais detalhada de cada etapa de fabricação da cerveja, onde é possível observar a relevância de cada uma e quais serão as alterações causadas por estas no produto final.

### 2.2.1 Moagem do Grão

Conforme Cervesia (2017), a moagem do grão no processo cervejeiro tem por objetivo tornar o interior dos grãos acessível ao meio aquoso para permitir a melhor ação enzimática no momento da brassagem. Idealmente o processo de moagem deve preservar a casca do grão praticamente inteira e quebrar o interior em partes

para melhor velocidade da reação. A preservação da casca é pertinente para auxiliar no processo de filtração e clarificação, pois esta faz a função filtro, que será abordada mais adiante. Além disso não é ideal moer o grão ao ponto de pó, pois este tenderia a entupir o filtro e não permitir o fluxo do mosto, prejudicando o processo.

No quesito de equipamentos para auxílio a moagem, a empresa Guzzo® tem uma linha específica para moagem de cereais, utilizando um sistema de rosca transportadora em seu interior e discos que fazem a quebra do grão, apresentado na figura 5.

**Figura 5 – Moinho de Disco**



**Fonte: Guzzo®**

Vale salientar que existem outros modelos existentes no mercado que podem fazer a mesma função com melhor eficiência, como os moinhos de rolo fabricado pela Brewmetal®. Contudo as diferenças monetárias são consideráveis e dessa forma optou-se pela utilização do moinho de discos apresentado.

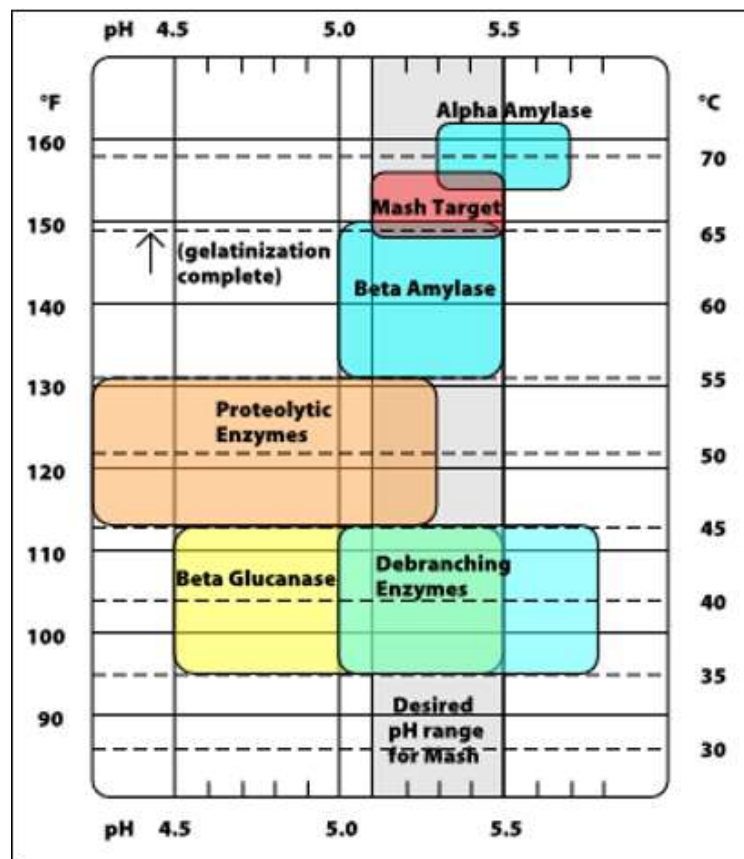
### 2.2.2 Mostura ou Brassagem

O processo de moagem do grão é seguindo pelo processo de mostura que também é conhecido por brassagem conforme apresentado na figura 4. Esta etapa de brassagem segundo Palmer (2006), consiste em inserir o malte moído em meio

aquoso, para a ativação das enzimas que irão converter os amidos em açúcares fermentáveis e não fermentáveis.

Basicamente este processo é a etapa em que uma quantidade de malte moído é colocado dentro do caldeirão que possui uma determinada quantidade de água em temperatura específica, que irá ativar as enzimas e produzir açúcares fermentáveis e não fermentáveis. Contudo a produção destes açúcares depende da ação das enzimas que possuem variação de atividade conforme a temperatura e o pH do meio em que estão presentes, mostrada na figura 6 apresentada por Palmer (2006).

Figura 6 – Escala das enzimas presentes na brassagem



Fonte: Palmer (2006)

Analisando a figura 6 pode-se observar as principais enzimas que irão atuar na transformação dos amidos em açúcares, e que cada enzima depende do pH e da temperatura do meio. Dessa forma o controle de temperatura no processo de brassagem é crucial para que se possa atingir um objetivo específico no gosto final



da cerveja, já que cada enzima que atua, transforma o amido em diferentes tipos de açúcares.

Também é pertinente expor que em temperaturas acima de 75°C as enzimas sofrem uma mudança denominada de desnaturação, ou seja, nesta etapa elas têm suas atividades finalizadas. Também faz parte do processo de brassagem uma etapa chamada *mash-out*, onde o mosto é aquecido até a faixa de temperatura de 75°C e permanece nesta por um determinado tempo, geralmente de 10 minutos. Dessa forma as ações enzimáticas são finalizadas pela desnaturação e o mosto está pronto para ser filtrado e clarificado (GRAINFATHER, 2017).

### 2.2.3 Filtragem e Clarificação

A filtragem e clarificação consiste na separação dos grãos moídos do mosto cervejeiro, conhecido também por bagaço. Na filtragem o mosto contém além de uma substância líquida rica em açúcares, uma grande parcela de grãos e cascas. Dessa forma há a necessidade da separação dos grãos e cascas para que se possa direcionar apenas o líquido para a fervura (PICCINI, 2002).

A separação pode ser feita de diversas formas, mas no caso deste trabalho, optou-se por utilizar o método do fundo falso.

Posteriormente à filtragem, existe o processo de clarificação, onde o mosto que passou uma vez pelo sistema do fundo falso, circulará novamente algumas vezes. Assim as partículas sólidas de pequenos tamanhos ficam retidas no bagaço, atenuando a turbidez do mosto para dar maior clarificação. (PICCINI, 2002)

Segundo Grainfather (2017), para a finalização do processo de clarificação, utiliza-se a técnica de lavagem dos grãos que é simplesmente inserir água em temperatura igual ao do mosto (75°C) na cama de grãos, que é formada quando o nível do mosto do caldeirão for inferior ou igual ao nível dos bagaços. Assim, além de extrair o açúcar presente no bagaço, ocorre a diluição parcial do mosto, tornando-o mais claro. A figura 7 mostra o exemplo de filtragem e clarificação do mosto.

**Figura 7 – Filtragem e clarificação**



**Fonte: Nbrewclub (2017)**

Vale salientar que no momento da lavagem dos grãos (ou passagem do *sparge*), não se deve desfazer a cama de grãos formada, então recomenda-se que a água seja espalhada de forma semelhante a de um regador, a fim de preservar a cama de grãos para que esta continue fazendo a etapa de clarificação, conforme assinala Msbeer (2017). Após essa etapa, o processo segue para a fervura com adição de lúpulo.

#### 2.2.4 Fervura com adição de lúpulo

O processo de fervura consiste em elevar a temperatura do mosto filtrado e clarificado para a temperatura de ebulição. Esta temperatura de ebulição é próxima dos 100 °C e é nesta etapa que acontece a inserção de lúpulo (PALMER, 2006).

O motivo pelo qual o lúpulo é inserido na fervura se dá porque é nesta temperatura em que ocorre maior extração de ácidos responsáveis pelo amargor e aroma do mosto. Os momentos que em se inserem os lúpulos variam de acordo com cada receita, mas existe a recomendação de inserir o lúpulo de maneira ponderada

em determinados momentos da fervura que geralmente tem duração de 60 a 75 minutos (Bamforth, 2009).

Basicamente a fervura com adição de lúpulo trata-se da definição do amargor e aroma da cerveja que será produzida, podendo deixar a gosto do cervejeiro escolher se prefere uma cerveja mais amarga e aromática por exemplo.

### 2.2.5 Resfriamento

Após o final da fervura, o mosto que agora possui alto teor de açúcar extraído na mostura e amargor e aroma extraído do lúpulo, passa para a etapa de resfriamento visto a figura 4 citada anteriormente.

O processo de resfriamento do mosto cervejeiro deve ocorrer de maneira rápida por 3 principais motivos. O primeiro deles é pelo fato de que quanto mais rápido o resfriamento, mais intenso são os aromas que ficarão presentes no mosto (HUANG, 2013). O segundo motivo é pelo fato de que quanto mais rápido ocorre o processo de resfriamento, a chance de contaminação do mosto por micro-organismos externos é menor (HUANG, 2013). O terceiro motivo é pelo fato de que a levedura que será introduzida no mosto no momento de fermentação possui uma temperatura ideal de trabalho, ou seja, não realizará a fermentação corretamente caso seja inserida em temperaturas diferentes das ideais (FERMENTIS, 2017).

Assim o resfriamento pode ser feito de diversas formas. Uma delas é com a utilização de um *chiller* de imersão, ou seja, uma serpentina feita em material com boa condução de calor como o alumínio ou cobre, e em seu interior circula água em temperatura menor a do mosto. Quando o conjunto é imerso no mosto este passa a trocar calor resfriando o mesmo (INCROPERA, 2011). A figura 8 apresenta um exemplo de um *chiller* de imersão.

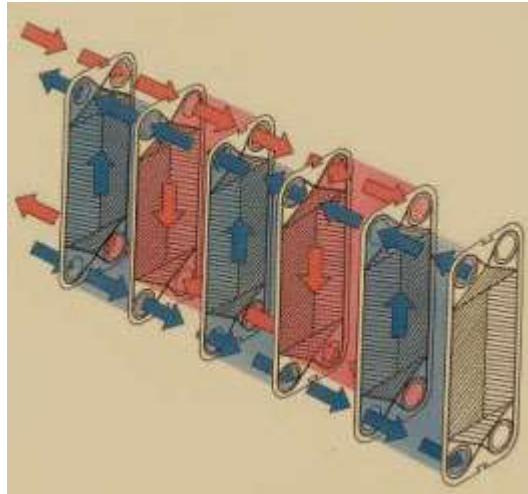
**Figura 8 – Chiller de imersão**

**Fonte: Autoria própria**

Este sistema costuma ser utilizado por cervejeiros artesanais pelo fato de ser menos oneroso quando comparado com outros modelos. Contudo dependendo da quantidade de mosto cervejeiro a ser resfriado, o *chiller* de imersão deve ser maior e ter um fluxo de água gelada em seu interior de maior intensidade. (INCROPERA, 2011)

Existe um outro modelo de equipamento para resfriamento de mosto cervejeiro que é conhecido como *chiller* de placas. Este modelo baseia-se na troca de calor em substâncias em movimento, ou seja, trata-se de um sistema em que o mosto cervejeiro circula dentro de um meio que possui uma divisão interna, e no lado do material condutor circula água em sentido contrário, conforme a figura 9.

**Figura 9 – Chiller de placas**



**Fonte: WE Consultoria® (2017)**

Dessa forma, pelo fato da planta cervejeira do trabalho em questão ter a capacidade projetada de 90 litros, optou-se pela utilização do *chiller* de placas para o processo de resfriamento do mosto, já que é feito em aço inox e mostra melhor rapidez no processo de troca de calor que o *chiller* de imersão.

### 2.2.6 Fermentação

De acordo com Palmer (2006) o processo de fermentação acontece quando leveduras (microrganismos) passam a consumir os açúcares presentes no mosto cervejeiro. Este consumo gera diferentes substâncias. Essas substâncias, a grosso modo são: o gás carbônico, novas leveduras e álcool em meio aquoso ou a cerveja propriamente dita.

Segundo Bamforth (2009), nem todas as substâncias presentes no mosto cervejeiro são consumidas pelas leveduras, assim as que permanecerem após a fermentação também serão responsáveis pelos sabores presentes no produto final.

No mosto têm-se então os açúcares como glicose, maltose e maltotriose, além dos aminoácidos, peptídeos e micronutrientes como ácidos graxos, zinco e vitaminas. Estas substâncias serão consumidas pela levedura no processo de fermentação. Vale salientar que as substâncias que não serão consumidas pela

fermentação são os polissacarídeos como as dextrinas, beta-glucanos, proteínas, taninos e lúpulo. Ao término da fermentação obtém-se álcool, álcoois superiores, diacetil, esterres, compostos de enxofre e ácido carboxílico (BAMFORTH, 2009).

Geralmente o processo de fermentação deve ter temperatura controlada (como visto anteriormente), já que a levedura possui uma temperatura ideal de trabalho. Assim a produção artesanal de cerveja permite que se utilize baldes ou bombonas de plástico atóxico como tanque de fermentação, o que deixa a aquisição destes menos onerosa comparada aos fermentadores específicos de aço inox 304.

### 2.2.7 Maturação

A maturação é o processo em que após a fermentação e a estabilização da densidade final do mosto. Este processo trata-se da diminuição gradual da temperatura do tanque fermentador, afim de ocorrerem reabsorções de aromas por parte da cerveja, além de deixa-la com aspecto mais límpido e diminuir o teor de diacetil, ácido sulfídrico e acetaldeído. (VENTURINI FILHO, 2000)

Vale salientar que existe o processo chamado *dryhopping*, que é a adição de lúpulo no momento da maturação, deixando a cerveja ainda mais aromática (PALMER, 2006).

### 2.2.8 Envase

Após a maturação a cerveja está pronta para ser engarrafada ou envazada em barris. A escolha por garrafas ou barril é uma estratégia utilizada pelo cervejeiro para a distribuição e armazenagem do produto.

Conforme colocado por Palmer (2006), o processo de *primming* é necessário quando se utilizam garrafas. O *primming* a grosso modo consiste em inserir uma pequena quantidade de açúcar para que a cerveja refermente na garrafa e adquira o gás carbônico no líquido (ou carbonatação).

Outra forma de envase comentado anteriormente é a utilização de barris que comumente usam a carbonatação forçada, ou seja, gás carbônico é injetado em seu interior sem a necessidade de refermentação. É pertinente expor que o processo de fabricação da cerveja dura de 3 a 4 semanas geralmente, desde o processo de brassagem até o envase final (VENTURINI FILHO, 2000).

É pertinente expor que a carbonatação forçada também pode ser utilizada em garrafas ou latas, mas para executar este processo é necessário um equipamento específico para este fim, como mostrado no catálogo do fabricante da enchedora modelo ECOZEGLA-ET-F-90/20GA da empresa Zegla®.

Após este processo a cerveja está pronta para ser consumida, degustada ou harmonizada com outros alimentos.

### 2.3 RELEVÂNCIA DO CONTROLE DE TEMPERATURA E LUPULAGEM NA BRASSAGEM

Entendida a importância no controle de temperatura no momento da mostura conforme as ações enzimáticas colocadas anteriormente é que motivou o desenvolvimento do projeto em questão. Além dos respectivos controles de temperatura, o controle dos tempos de lupulagem é importante para que se obtenha o objetivo final de aroma e amargor, já que esta deve ser feita de forma ponderada durante a fervura. É por isso que a utilização de um controle de tempo foi escolhida para fazer parte do equipamento.

### 2.4 MÉTODO DE RECIRCULAÇÃO CONTÍNUA COM FUNDO FALSO

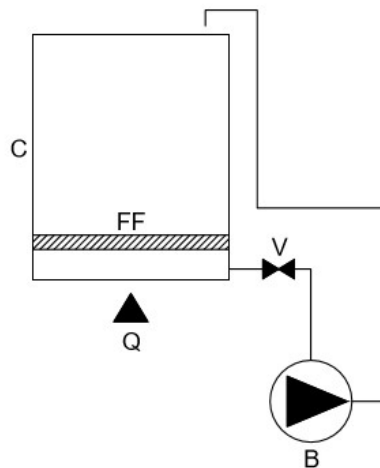
O método de fabricação de cerveja que utiliza um fundo falso apresentado anteriormente foi adotado para sua utilização neste trabalho. Contudo como todo método este possui algumas desvantagens sendo a principal delas a diferença de temperatura entre o mosto contido abaixo e acima do fundo falso. Quando o sistema de aquecimento é ligado a tendência é aquecer mais o mosto da parte inferior do

fundo falso ocasionando a desnaturação proteica por superaquecimento visto anteriormente na descrição da mostura.

Além deste problema, existe o efeito de escurecimento enzimático por superaquecimento que é determinado como reação de Maillard e caramelização, que além de ocasionar a desnaturação proteica, gera aromas e sabores não desejados para o objetivo de cerveja, conforme Bastos (2011).

Para o melhor entendimento do sistema, a figura 10 mostra a disposição física do modelo adotado.

**Figura 10 – Sistema de recirculação contínua**



**Fonte: Autoria própria**

Como visto na figura 10, o sistema de recirculação contínua é composto por um caldeirão de mostura (C) com fundo falso (FF), uma válvula tipo esfera (V) de saída na parte inferior abaixo do fundo falso do caldeirão, bomba para a recirculação (B) do mosto em andamento e um aquecedor (Q), neste caso a gás, com chama em contato direto com o fundo do caldeirão. Assim quando se inicia o aquecimento, é pertinente que a bomba esteja acionada para que todo o líquido seja homogeneizado consequentemente homogeneizar a temperatura.

Quando se tem a chama do queimador ligada, o líquido contido na parte inferior do fundo falso tende a ficar mais quente que o líquido da parte superior do fundo falso, assim a bomba transfere o líquido da parte inferior para a parte superior



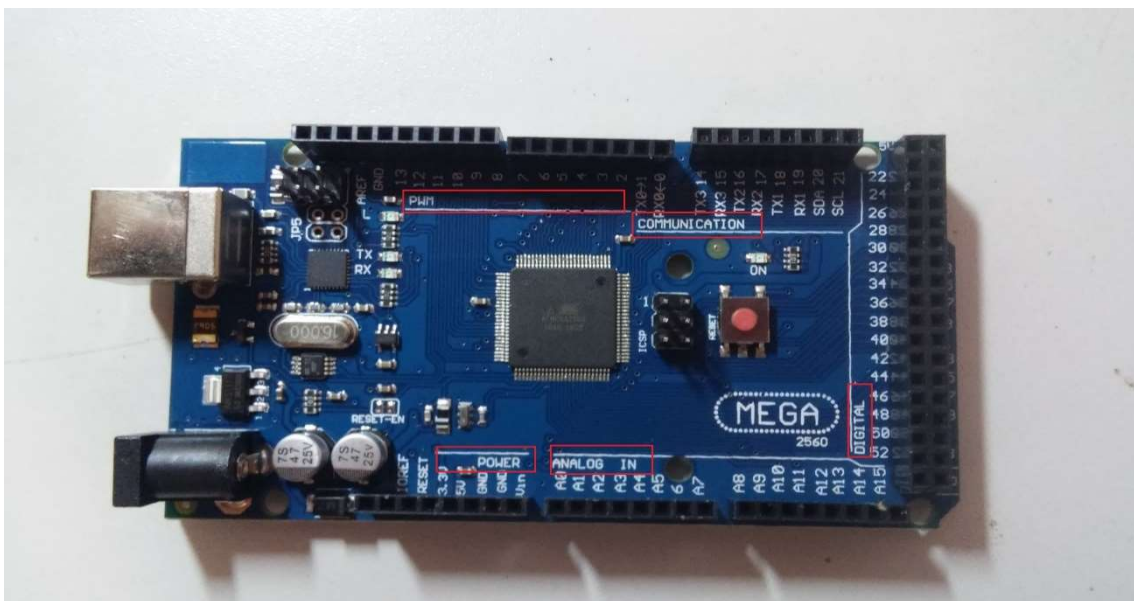
e o líquido da parte superior passa através do fundo falso para a parte inferior homogeneizando o sistema, evitando assim a desnaturação proteica e as reações de caramelização e Maillard.

## 2.5 PLATAFORMA ARDUINO MEGA 2560

A plataforma microcontrolada Arduino Mega 2560 trata-se basicamente de um microcontrolador Atmega 2560 acoplado a uma placa que faz a ligação deste com componentes básicos como cristal oscilador, dispositivos de regulagem de tensão, indicadores de funcionamento, entrada para comunicação, dentre outros que tornam possível a comunicação do microcontrolador para a programação do mesmo via cabo USB (*Universal Serial Bus*), além da facilidade para conectá-los com periféricos de atuação e leitura observado no *datasheet* do fabricante (MANTECH, 2017).

A figura 11 apresenta a disposição física do microcontrolador Atmega 2560 na plataforma Arduino Mega 2560, onde pode-se observar em destaque as entradas, saídas, portas de comunicação, alimentação, etc.

**Figura 11 – Plataforma Arduino Mega 2560**



Fonte: Autoria própria

Segundo Mcroberts (2011), a plataforma Arduino pode interagir com o ambiente atualizando as saídas por meio da leitura das entradas, ou seja, este é capaz de realizar tarefas programadas como por exemplo ligar uma lâmpada em um determinado intervalo de tempo e desliga-la quando necessário por meio da leitura de uma porta de entrada (botão) ou até mesmo quando um determinado tempo for decorrido.

Dessa forma é pertinente expor as principais características da plataforma microcontrolada Arduino Mega 2560 quanto seus limites de operação capacidades conforme Mantech (2017) adaptado.

**Tabela 1 – Características principais do Arduino Mega 2560**

Microcontrolador	ATmega2560
Tensão de operação	5 V
Tensão de entrada (Recomendada)	7 – 12 V
Faixa de tensão de operação	6 – 20 V
Portas digitais de entrada ou saída	54 (14 podem ser configuradas como saída PWM)
Entradas analógicas	16
Corrente máxima por porta digital	40 mA
Memória Flash	256 KB
Cristal de clock	16 Mhz

**Fonte: MANTECH (2017) adaptado**

É relevante dizer que existem diversas plataformas Arduino, e que cada uma delas é utilizada conforme a necessidade de aplicação. Neste caso em específico o projeto necessitou de mais de 14 saídas e entradas digitais eliminando a possibilidade da utilização de outros modelos de plataforma Arduino.

## 2.6 SENSOR DE TEMPERATURA DS18B20

O sensor de temperatura DS18B20 é um sensor de temperatura que pode ser utilizado em aplicações de controles termostáticos, sistemas industriais, termômetros, dentre outras aplicações referentes a leitura de temperatura (MAXIM INTEGRATED PRODUCTS, 2015).

Conforme o fabricante este sensor de temperatura possui comunicação *onewire*, que possibilita a utilização de até 8 sensores em uma única linha de comunicação. Suas características e limites de operações principais são: tensão de operação de 3 – 5,5 volts, faixa de leitura de temperatura de -55 – 125°C.

A motivação pela escolha deste modelo de sensor foi associada ao seu valor relativamente pouco oneroso e por estar dentro dos limites de operação da plataforma microcontrolada Arduino Mega 2560 descrito anteriormente.

## 2.7 IHM – INTERFACE HOMEM MÁQUINA

Como o próprio nome descreve, a interface homem máquina (IHM) refere-se a qualquer sistema que permita o operador comunicar-se com o sistema processado de forma a visualizar e inserir informações. Basicamente uma IHM é composta por botões, sejam eles virtuais ou físicos, para a inserção de dados e um *display* para visualização de informações (PAREDE, 2011).

Assim, para que se possa comunicar-se diretamente com a plataforma Arduino Mega 2560, optou-se por desenvolver uma IHM constituída por um *display* de cristal líquido (LCD), botões de pulso e uma sirene.

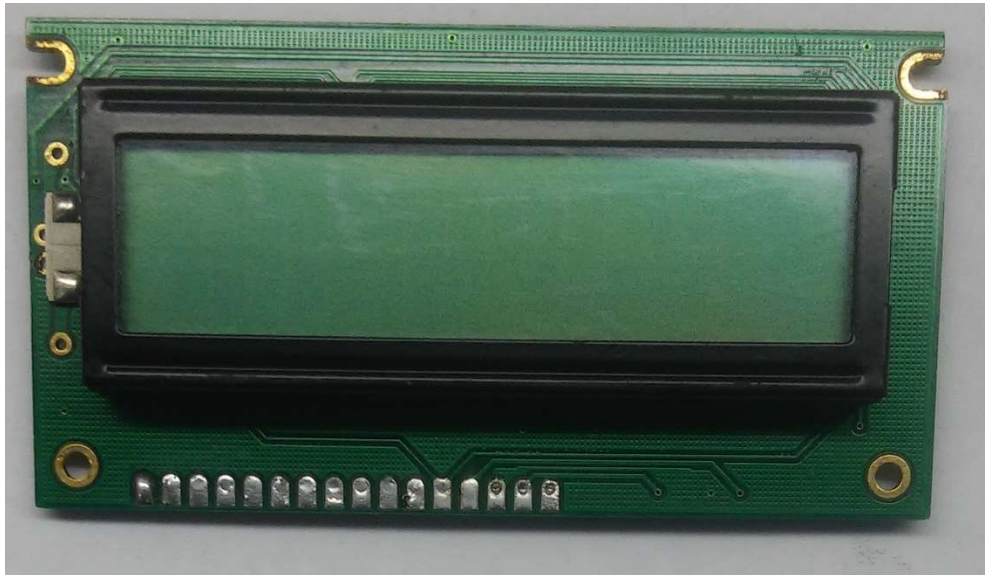
### 2.7.1 Módulo LCD

O módulo LCD é constituído de uma tela fabricada em material de vidro e utiliza os princípios da polarização da luz associada a um material de cristal líquido. Conforme níveis de tensão aplicados em seus terminais internos, este gera um

ponto na tela, que combinado com o driver de acionamento HD44780U da fabricante HITACHI®, é possível escrever caracteres alfanuméricos na tela, além dos caracteres especiais (HITACHI, 1998).

Os módulos LCD possuem diferentes formatos no quesito de quantidade de caracteres por linha e quantidades de linhas possíveis para a escrita. Dessa forma a utilização para a tela da IHM foi um módulo LCD de 16 colunas por 2 linhas conforme visto na figura 12.

**Figura 12 – Módulo LCD**



**Fonte: Autoria própria**

Os terminais de saída no caso deste módulo serão descritos a seguir o melhor entendimento das ligações a serem feitas com a plataforma microcontrolada utilizada. Assim a tabela 2 descreve os pinos referentes a este módulo junto com as características gerais do mesmo.

**Tabela 2 – Pinagem do modulo display LCD**

Pino	Símbolo	Descrição
1	GND	Alimentação geral (0V)
2	Vcc	Alimentação geral (5V)
3	Vo	Alimentação de contraste (0-5V)
4	RS	Seleção (0-dado 1-instrução)
5	R/W	Seleção (0-leitura 1-escrita)
6	E	<i>Chip Select</i> (0-habilita 1-desabilita)
7	B0	Barramento de dados
8	B1	Barramento de dados
9	B2	Barramento de dados
10	B3	Barramento de dados
11	B4	Barramento de dados
12	B5	Barramento de dados
13	B6	Barramento de dados
14	B7	Barramento de dados
15	A	Anodo da luz de fundo
16	K	Catodo da luz de fundo

Fonte: Shenzhen Eone Electronics (2005) adaptado

Com base na tabela referente a pinagem e com a utilização de bibliotecas de programação já prontas, permite-se então a interação deste módulo com a plataforma microcontrolada Arduino Mega 2560 de maneira simples para a escrita de dos dados referentes ao objetivo de projeto.

### 2.7.2 Botões de pulso

Os botões de pulso escolhidos para constituir a IHM em questão foram os botões da fabricante HOKURIKU®, que possuem a função de fechar um contato elétrico mecanicamente quando o botão for acionado manualmente. Conforme colocado por Mcroberts (2011), é necessário a utilização de resistores de *pull-up* ou

*pull-down* quando se utiliza este tipo de botão para a garantia que o barramento esteja em nível lógico alto ou baixo, e evitar possíveis ruídos que geram mal funcionamento na leitura de uma entrada (quando o botão é pressionado ou não) na plataforma microcontrolada.

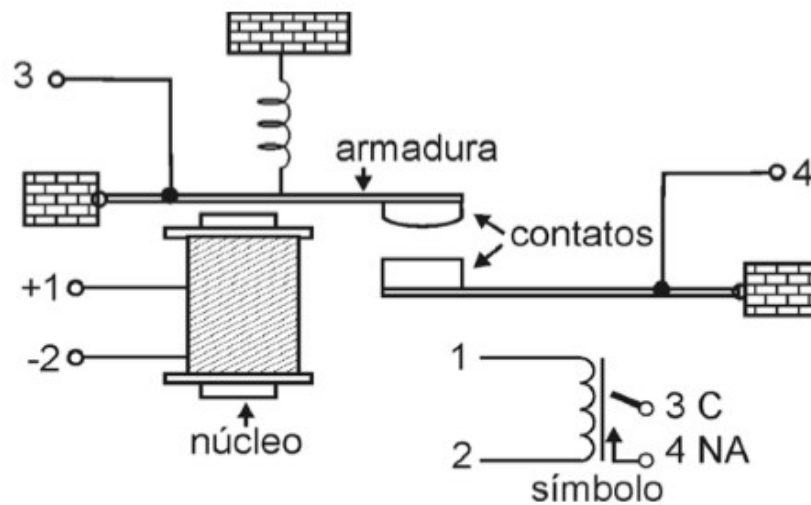
### 2.7.3 Sirene

A sirene tem a função de emitir um som em frequência audível quando aplicada uma determinada tensão em seus terminais. Assim segundo a fabricante Velleman®, a sirene em questão e escolhida para ser implementada na IHM possui tensão de trabalho de 5 volts, corrente de consumo de 20 mA, emite um ruído de 87 dB com uma frequência de 3,2 KHz.

## 2.8 MÓDULO RELE DE 8 CANAIS

Rele pode ser definido como um elemento que faz a comutação física de um circuito elétrico por meio da ação eletromecânica. De maneira geral o rele faz a função de uma chave liga/desliga por meio de um sistema eletromagnético, que quando energizado atraca os contatos fechando o circuito (BRAGA, 2012). A figura 13 a seguir mostra o esquemático da estrutura física simplificada de um rele.

Figura 13 – Esquemático Rele

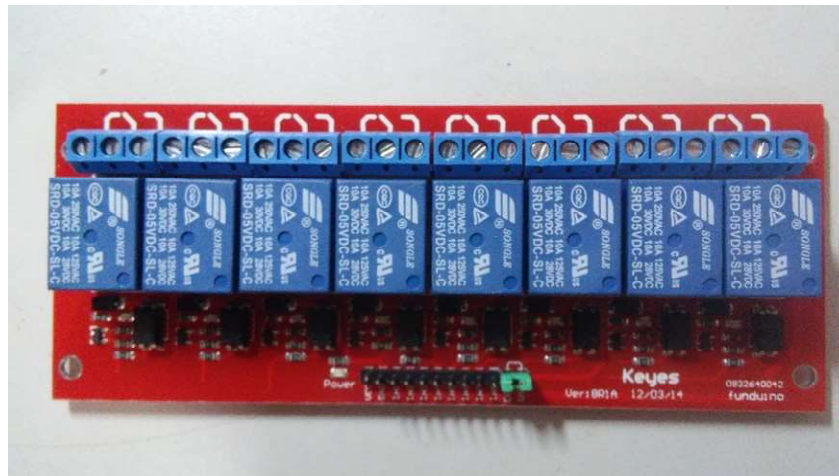


Fonte: BRAGA (2012)

Conforme a figura 13, quando uma corrente circula pelos contatos 1 e 2, um fluxo magnético é gerado no núcleo e este atrai a armadura, fechando os contatos 3 e 4, permitindo a passagem da corrente. Assim se existir uma diferença de potencial aplicada em uma determinada carga e esta estiver ligada em série com os contatos 3 e 4, quando energizada a bobina pelos contatos 1 e 2, a carga irá se energizar e funcionar o sistema.

O uso do relé se dá ao fato da necessidade de acionamento de cargas que consomem maior quantidade de corrente como um motor elétrico por exemplo. Como visto na descrição do controlador Arduino Mega 2560 utilizado, este não suporta uma corrente de saída suficiente para alimentar as cargas que foram utilizadas de maneira direta. A solução para este problema foi a opção de utilização do módulo relé apresentado na figura 14 a seguir.

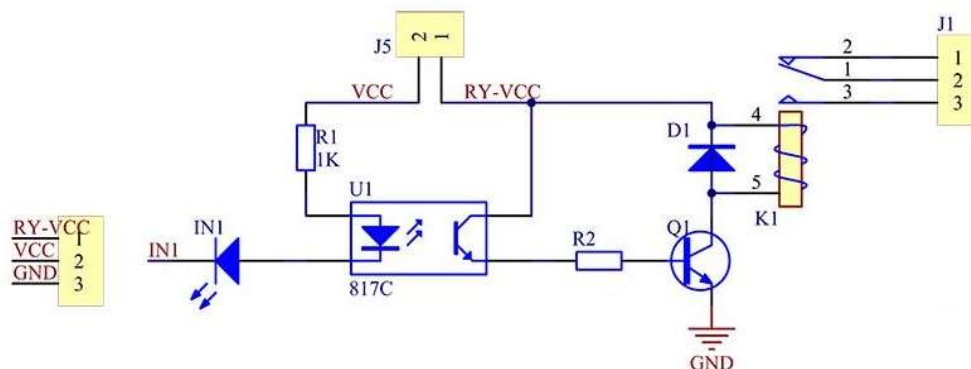
Figura 14 – Módulo rele 8 canais



Fonte: Autoria própria

Visto a figura 14, nota-se que além dos 8 relés contidos, este módulo possui um sistema de opto acoplamento individual de para cada rele conforme o manual técnico da Serial Link® (2011). Esta configuração permite uma proteção para o controlador, já que este sistema possui opto acopladores, que são mostrados na figura 15 a seguir.

Figura 15 – Esquemático eletrônico de 1 canal do modulo rele



Fonte: Osoyoo (2017) adaptado

A Figura 15 representa o esquema eletrônico individual de 1 canal do módulo. Neste circuito, pode-se observar que o sistema funciona com uma tensão de 5 volts e possui um LED indicador de funcionamento quando os contatos estão atracados, além de uma entrada de controle. Basicamente quando um sinal de nível lógico alto

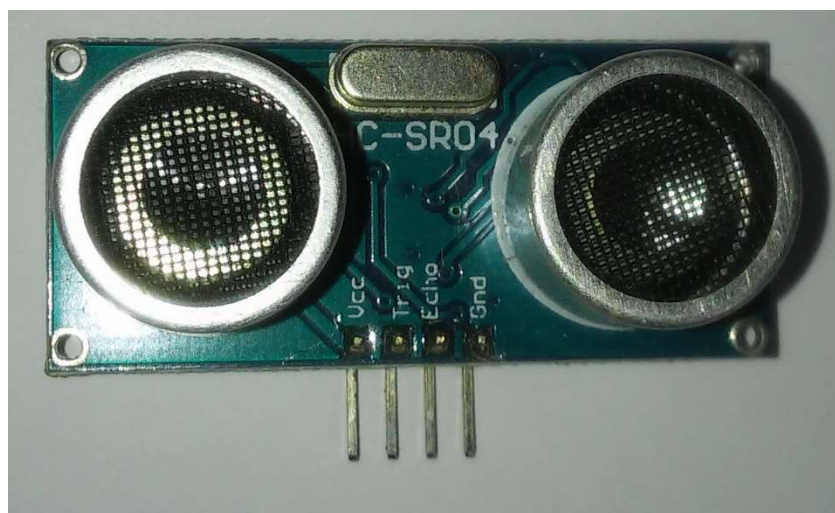


é aplicado na entrada de controle, este realiza a condução de corrente pelo opto acoplador, que irá conduzir uma corrente na base do transistor e este irá conduzir uma corrente suficiente para atracar os contatos do rele.

## 2.9 SISTEMA DE MEDIÇÃO DE NÍVEL

O sistema de medição de nível de água utilizado nos caldeirões de mostura e de enxague será composto por um módulo sensor ultrassônico do tipo HC-SR04. Este módulo é composto como elementos principais um emissor de ultrassom na frequência de 40 khz e seu respectivo receptor, conforme KT-elektronik® (2012). A figura 16 apresenta o modelo deste módulo para maior clareza da disposição física de seus componentes.

**Figura 16 – Sensor Ultrassônico HC-SR04**



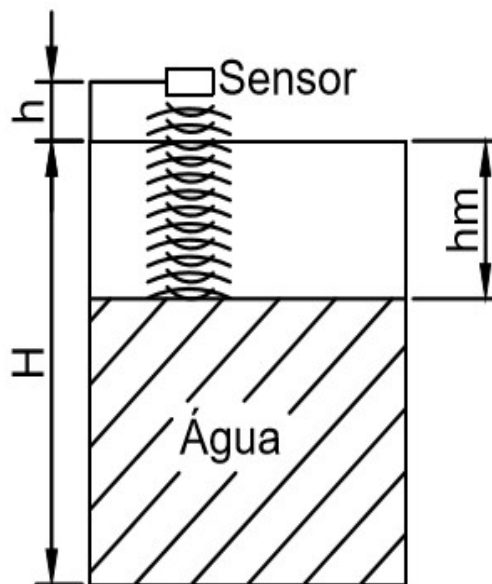
**Fonte: Autoria própria**

O seu funcionamento é baseado na reflexão sonora do sinal ultrassônico emitido em um determinado objeto, ou seja, quando se emite o sinal ultrassônico, a contagem de tempo é feita por parte de um microcontrolador, até o momento em que este retornará. Dessa forma é permitido calcular a distância do objeto, já que a velocidade do som no ar tem valor definido e o tempo foi cronometrado. Utilizando-se do conceito de que a distância total percorrida é dada pelo produto da velocidade

e tempo pode-se então definir o valor total dessa distância. Contudo este resultado deve ser dividido por 2, pois a distância proveniente do cálculo é duas vezes a distância do objeto, conforme descrito por KT-elektronic® (2012).

No caso da aplicação do sensor neste trabalho, é pertinente expor algumas considerações. O objeto que terá a distância medida será o nível de água que varia conforme a quantidade dentro do caldeirão, assim se o sensor ficar posicionado acima do caldeirão e em direção fundo do mesmo, quanto maior o nível de água, menor a distância mensurada, conforme a figura 17.

**Figura 17 – Sistema de medição de nível**



**Fonte: Autoria própria**

Visto a figura 17, segue então a descrição do sistema de medição de nível. A altura do caldeirão ( $H$ ) é a altura máxima que o sistema pode armazenar. Quando o sensor ultrassônico emite o sinal, este é refletido e receptado para ser processado, obtendo-se então a altura da parte vaga do caldeirão ( $hm$ ) somado com a altura do sensor em relação ao nível máximo do mesmo ( $h$ ). Sabendo o diâmetro do caldeirão junto sua altura máxima subtraída da altura vaga é possível calcular o volume de água presente.

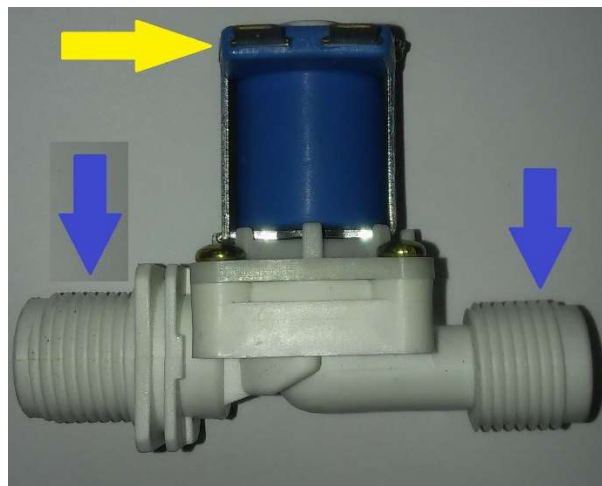
## 2.10 VÁLVULAS SOLENOIDE

Em aplicações de automação seja ela no meio industrial ou residencial, para qualquer tipo de manipulação de fluidos, gases ou vapores existem as válvulas solenoides. Por isso é que há a necessidade do entendimento do seu funcionamento antes da aplicabilidade para cada caso, conforme o catálogo de válvulas solenoides da empresa fabricante Jefferson®.

Basicamente uma válvula solenoide é composta por 2 sistemas. O primeiro deles é o sistema eletromagnético que tem funcionamento semelhante ao módulo rele explicado anteriormente. Neste sistema quando a bobina é energizada, a ação eletromagnética gera uma ação mecânica de movimento do núcleo móvel. O segundo sistema é o sistema mecânico do corpo da válvula, ou seja, quando o sistema eletromagnético é excitado o núcleo móvel se movimenta e atua no sistema mecânico do corpo da válvula, abrindo uma via como por exemplo. Analogamente é como abrir uma torneira utilizando um contato elétrico (Jefferson, 2017).

É pertinente expor que não serão classificados os tipos de válvulas já que neste trabalho optou-se por utilizar uma válvula solenoide de duas vias e normalmente fechada, ou seja, é uma válvula solenoide simples que quando energizada permite a passagem do fluido entre dois pontos. A figura 18 mostra uma das válvulas escolhidas neste trabalho.

**Figura 18 – Válvula solenoide de água 2 vias normalmente fechada**



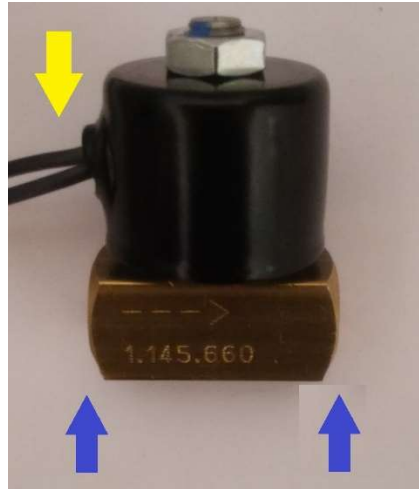
**Fonte: Autoria própria**

Visto a figura 18, esta tem em destaque a parte superior (indicada pela seta amarela) que são os polos da alimentação elétrica da bobina e a parte inferior onde estão destacadas a entrada e saída de água (pela seta azul). É importante que se tenha o conhecimento das características desta válvula solenoide escolhida, para a justificativa do seu uso.

A primeira característica pertinente é sua utilização para água e o seu número de vias, assim esta possui duas vias que são a entrada e a saída de água. A segunda característica pertinente é sua tensão de operação da bobina que é de 12 volts. A terceira é o seu estado de repouso que é normalmente fechada, que quando energizada permite o fluxo de água e quando não energizada o fluxo de água é interrompido.

Outras características pertinentes são as condições de trabalho referentes a temperatura e pressão da água. Conforme o fabricante Marcoval® a temperatura máxima de trabalho é de 60 °C e a pressão de operação é de 0,2 à 8 kgf/cm<sup>2</sup>. Seu corpo é plástico e possui ciclo de trabalho de 50 mil operações com conexões de ½" com entrada e saída 180° de defasagem. Dessa forma esta válvula solenoide foi escolhida para a utilização no sistema de entrada de água cervejeira, a qual está dentro das condições de trabalho planta.

Analogamente a válvula solenoide de água, é necessário a definição da válvula solenoide de gás responsável pela liberação do fluxo de gás para o queimador que é a fonte de calor para aquecimento dos 3 caldeirões do sistema em questão. Seu princípio de funcionamento é o mesmo que a válvula de água e sua tensão de operação da bobina é de 12 volts. A seguir é apresentada a figura 19 que mostra a válvula em questão.

**Figura 19 – Válvula solenoide de gás**

Fonte: Autorial própria

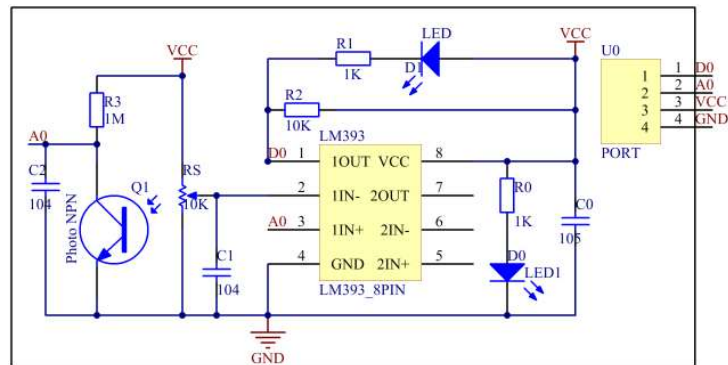
Conforme visto a figura 19, a seta da cor amarela indica os polos para a conexão de alimentação elétrica e as setas da cor azul indicam a entrada e saída de gás. Segundo o fabricante Gasnag® esta trabalha com pressões de 0 à 11,95 kgf/cm<sup>2</sup>, tem o corpo feito em latão, entrada e saída com defasagem de 180° e tem seu estado natural normalmente fechada. Dessa forma esta válvula foi escolhida para sua utilização no sistema de acionamento do gás do queimador.

## 2.11 SENSOR DE CHAMA

Tratando-se do sistema de aquecimento dos caldeirões por meio da combustão de gás butano de uso doméstico, é importante que se tenha o cuidado com a segurança. Dessa forma o sensor de chama é uma das alternativas para que se tenha este controle.

O sensor de chama escolhido foi o sensor *Arduino flame sensor module* que é constituído por um fototransistor, um circuito integrado LM393 (amplificador operacional duplo) e um conjunto de resistores, capacitores e elementos semicondutores que obedecem a configuração conforme a figura 20.

**Figura 20 – Sensor de chama Arduino**



**Fonte: Theorcircuit (2017) adaptado**

A tensão de alimentação deste circuito é de 3,3-5 volts, detecta chamas de espectro na faixa de 760-1100 nm, possui 4 pinos sendo estes 2 de alimentação (GND e VCC), 1 saída digital e 1 saída analógica. De acordo com a figura 20, este módulo possui um resistor variável que permite o ajuste de disparo do sinal de alarme de chama (pino D0) para possíveis calibrações além de uma saída analógica (A0) que permite a leitura de intensidade da fonte de fogo. A figura 21 mostra a disposição física do módulo em questão.

**Figura 21 – Módulo sensor de chama Arduino**



**Fonte: Autoria própria**

Pode-se observar a presença de 2 leds (um próximo ao pino VCC e outro próximo ao pino GND) que são responsáveis pela indicação de alimentação e indicação de disparo de sinal de chama.

Assim este sensor foi escolhido com o intuito de identificar se existe a presença de chama e evitar possíveis acionamentos das válvulas solenoide de gás

quando não há presença de fogo, evitando o vazamento e conseqüentemente riscos ao operador do sistema.

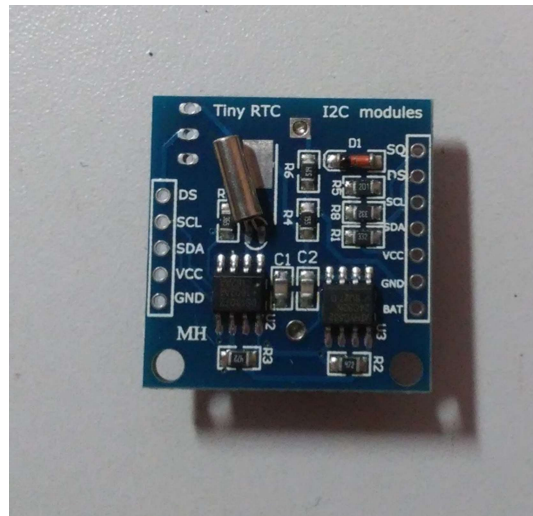
## 2.12 MÓDULO RTC

De acordo com a descrição do processo de mostura, além da importância do controle das temperaturas em que o meio se encontra, é essencial controlar o tempo em que o mosto deve permanecer nestas como citado anteriormente no capítulo de descrição do processo cervejeiro.

Para que se tenha um bom desempenho dos sistemas de controle de temperatura e segurança do sistema, e ao mesmo tempo controlar o tempo em que o processo está acontecendo, optou-se pela utilização do módulo RTC(*Real Time Clock*) DS1307. Este módulo é basicamente um relógio em tempo real que faz a contagem dos segundos, minutos e horas, dias e até considerações de anos bissextos. Quando conectado a uma bateria, este não depende de fonte externa para continuar a contagem caso ocorra falta de alimentação no sistema, conforme sugerido no *datasheet* da Maxim Integrated™ (2015).

Dessa forma, é este módulo que irá informar a contagem exata para o Arduino tome as ações programadas. Na figura 22, tem-se a disposição física de um módulo RTC DS1307 proposto por Cooper (2016) adaptado.

**Figura 22 – Módulo relógio em tempo real**



**Fonte: Autoria própria**

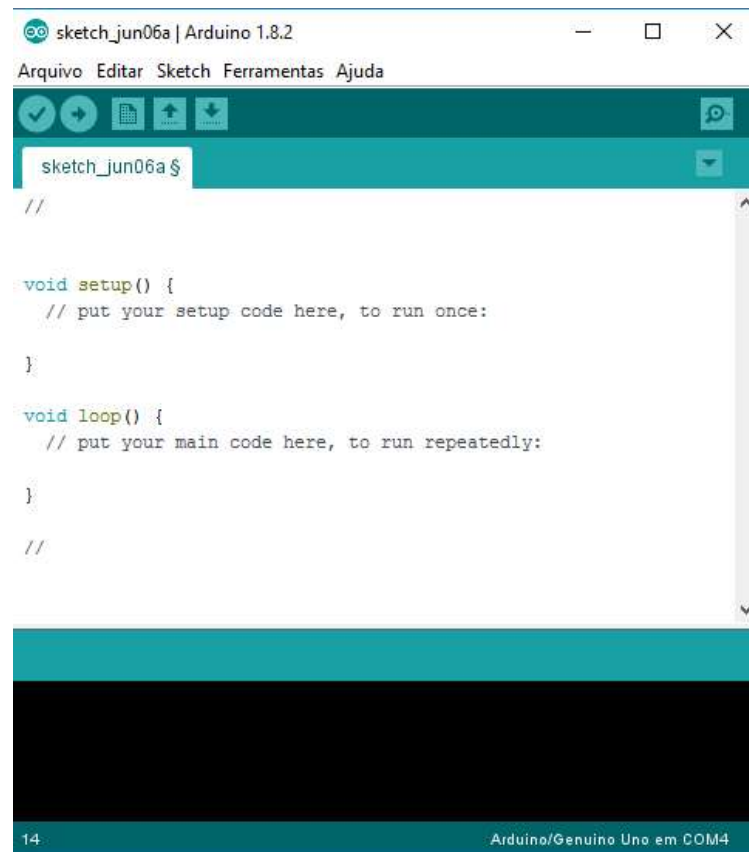
Este módulo tem alimentação em 5 volts e sua transferência de informação é por meio de comunicação I<sup>2</sup>C. Os pinos referentes a comunicação deste com um microcontrolador são o pino SCL e o pino SDA (MAXIM INTEGRATED, 2015).

Para sua utilização na contagem de tempo aplicada neste trabalho, serão utilizados os pinos de alimentação (GND e VCC) ligado a fonte de 5 volts e os pinos SCL e SDA conectados individualmente em 2 entradas analógicas da plataforma Arduino conforme proposto por Cooper (2016). Dessa forma é possível controlar os tempos referentes as rampas de temperatura aplicadas no processo cervejeiro.

### 2.13 SOFTWARE UTILIZADO NA PROGRAMAÇÃO

O *software* para o desenvolvimento dos programas que serão gravados na plataforma microcontrolada Arduino (neste caso a versão Mega 2560) é um *software* livre de custos baseado em linguagem C (MCROBERTS, 2011). A figura 23 mostra sua tela inicial com sua estrutura básica para a programação.



**Figura 23 – Software de programação da plataforma Arduino**

```
sketch_jun06a | Arduino 1.8.2
Arquivo Editar Sketch Ferramentas Ajuda
sketch_jun06a $
//
void setup() {
  // put your setup code here, to run once:
}
void loop() {
  // put your main code here, to run repeatedly:
}
//
14 Arduino/Genuino Uno em COM4
```

**Fonte: Autoria própria**

Na figura 23, pode-se observar o termo `void setup(){}`, que corresponde a parte do programa que irá ser executada apenas uma vez. Esta parte do programa basicamente é a definição de valores que não irão sofrer alteração ao longo do processamento dos dados ou como por exemplo a definição de qual pino da plataforma será uma entrada ou saída digital. A parte onde esta escrito `void loop(){}`, refere-se a parte do programa que irá ser executada sucessivamente quando chega ao final segundo McRoberts (2011).

Assim, tal *software* será utilizado para o desenvolvimento do programa que irá executar a supervisão e controle do processo proposto neste trabalho.

## 2.14 BOMBA CENTRÍFUGA

Conforme o guia básico de bombas da Procel (2009), bombas hidráulicas são máquinas operatrizes que realizam a transferência de energia a um líquido para que este possa ser transferido de um ponto a outro.

Como visto no item 2.4 referente ao método aplicado a fabricação de cervejas artesanais, este é baseado em um sistema de recirculação contínua onde é necessário a utilização de uma bomba para que o líquido seja transferido entre os caldeirões ou recirculado para que o sistema funcione de forma esperada conforme a figura 10.

Para a escolha da bomba em questão algumas considerações são pertinentes. Em primeira instância, a bomba fica localizada na parte inferior dos caldeirões e assim não há a necessidade de utilização de uma bomba que gera vácuo, já que a gravidade permite que o líquido chegue até o cabeçote (elemento que realiza o bombeamento) para posteriormente ser transferido. Assim a escolha da bomba foi do tipo centrífuga que obedece este critério conforme Procel (2009).

Em segundo lugar, como tratamos de líquidos em temperaturas de até a temperatura de fervura (100°C aproximadamente), então esta bomba deve necessariamente suportar esta temperatura. Outro ponto importante a se considerar é referente a higienização, então esta deve ser desmontável para sua limpeza.

No caso deste trabalho, os caldeirões de mostura e fervura possuem um volume máximo de 90 litros, então para o sistema adotou-se um critério que em menos de 5 minutos todo o líquido contido no caldeirão deve ser recirculado por completo na recirculação e no resfriamento.

Baseado nestes critérios elementares, a bomba escolhida é semelhante ao modelo TP40 G3 da marca Komeco®, com vazão de 30 litros por minuto, feita em corpo de inox, conexões de rosca  $\frac{3}{4}$ " tipo macho, tensão de operação de 127 volts, 120 watts de potência consumida e suporta temperaturas de até 110°C.

## 2.15 CONEXÕES, TUBULAÇÕES, VÁLVULAS E QUEIMADORES

Para a transferência de líquidos são necessárias tubulações, conexões e válvulas para que as manobras do sistema sejam realizadas. Dessa forma como tratamos de elementos alimentícios em alta temperatura, as tubulações e conexões escolhidas são do tipo flexível e atóxica da marca Ibirá®.

Já as válvulas necessárias a manobra dos líquidos por conveniência foram determinadas a utilização das válvulas tipo esfera feita em aço inox 304, conexão 3/4" tipo fêmea que suportam alta temperatura da marca Aceptil® mostradas pela figura 24.

**Figura 24 – Válvula tipo esfera em aço inox**



**Fonte: Autoria própria**

Nos elementos referentes ao controle de gás, optou-se pela escolha de tubulações e reguladores da marca Aliança® que obedecem a norma NBR8473 para manter a segurança do uso dos queimadores. Os queimadores são da marca Cagafogo®, que são de fabricação específica para seu uso na fabricação de cervejas artesanais.

Assim os elementos de manobra estão definidos para que se tenha uma boa prática de fabricação cervejeira, segurança e praticidade.

## 2.16 ESTRUTURA FÍSICA DE SUSTENTAÇÃO

A estrutura de sustentação dos caldeirões deve obedecer ao formato de um sistema que possibilite a recirculação contínua no processo de mostura e também suporte o peso que dos caldeirões. Para o desenvolvimento desta estrutura, em função da carga de peso que será aplicada ao mesmo, pode-se utilizar o material aço SAE 1020.

Referente ao formato que definirá a disposição física dos caldeirões, os critérios definidos são que os caldeirões de mostura e fervura fiquem a uma altura aproximada de 1,5 metros do solo e ambos no mesmo nível. Já o caldeirão de enxague deve ter o seu fundo localizado no mínimo da altura do topo do caldeirão de mostura, para que não necessite a utilização de mais uma bomba para a etapa de enxague, aproveitando a força da gravidade.

Outros materiais podem ser utilizados na confecção da estrutura física de sustentação como o aço INOX 304, mas em função de seu alto valor comercial, a escolha foi pelo aço SAE1020 a fim de redução de custos.

## 2.17 ESTRUTURA FÍSICA ELÉTRICA E ELETRÔNICA

Para a definição da estrutura física elétrica e eletrônica serão definidos alguns materiais referentes a sua fixação e montagem. Para fixar os elementos elétricos e eletrônicos, a caixa escolhida foi do tipo hermética da marca Schuchmacher® com as dimensões de 30 centímetro de comprimento, 20 centímetros de largura e 15 centímetros de altura. Esta caixa é de uso industrial, feita em material termoplástico de alta resistência mecânica e impactos.

Os cabos internos escolhidos para o sistema de potência foram cabos flexíveis para facilitar que facilite a montagem e organização, com secção transversal de 1,5 milímetros da marca Megatron®.

Os demais elementos para a confecção do sistema de controle escolhidos foram diversos, como prensa cabo, *jumper* para Arduino, botão alavanca 3 posições, lâmpadas indicadoras, dentre outros.

### 3. MATERIAIS E MÉTODOS

Para o desenvolvimento deste trabalho, que teve a orientação do professor Dr. Frederic Conrad Janzen, primeiramente foi realizado o estudo referente ao processo artesanal de fabricação cervejeira, com o intuito de desenvolver a planta para o funcionamento do sistema. A segunda etapa do trabalho foi o estudo para a escolha dos elementos fundamentais para a construção do sistema físico mecânico.

Basicamente os componentes que constituíram na etapa de construção mecânica do sistema foram:

- Caldeirão em aço INOX com capacidade de 98 litros
- Caldeirão em alumínio com capacidade de 68 litros
- Fundo falso em INOX
- Fogão cervejeiro em aço SAE 1020
- 3 unidades de queimadores cervejeiro
- 8 válvulas tipo esfera em aço INOX
- 3 unidades de chama piloto
- 30 metros de tubulação atóxica
- Bomba centrífuga 30 litros/minuto
- Resfriador de placas

Após a aquisição destes, a construção do mesmo foi executada para dar segmento a parte responsável pelo controle automático do processo. Para isso foi necessária a aquisição dos seguintes materiais:

- Plataforma microcontrolada Arduino Mega 2560
- Display LCD 16X2
- Módulo rele 8 canais
- 3 sensores de chama
- 3 sensores de temperatura

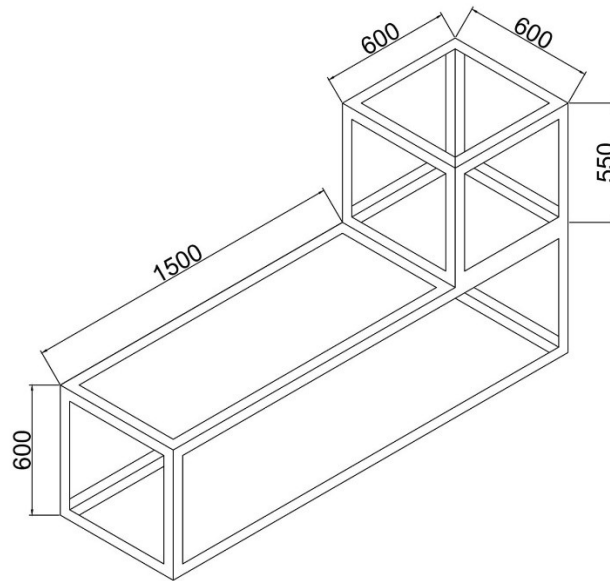
- 2 sensores ultrassônico
- 3 válvulas solenoide para gás
- 2 válvulas solenoide para água
- Fonte de alimentação chaveada 12 volts com corrente máxima de 15 amperes
- Caixa hermética
- 6 botões tipo alavanca
- 5 lâmpadas indicadoras 12 volts
- 1 lâmpada indicadora 127 volts
- Cabos diversos
- 4 botões de pulso tipo *push-button*
- 1 sirene eletrônica 5 volts
- Módulo relógio em tempo real (RTC)
- CI LM7805
- 2 capacitores eletrolíticos de 2200 microfards de 16 volts

Feita a aquisição destes materiais, realizou-se a montagem do mesmo seguida pela inserção desta na planta, possibilitando o segmento da próxima etapa que foi a programação do *software* que controla todo o sistema e realizado o teste de funcionamento.

## **4. METODOLOGIA**

### **4.1 MONTAGEM DA ESTRUTURA FÍSICA DE SUSTENTAÇÃO**

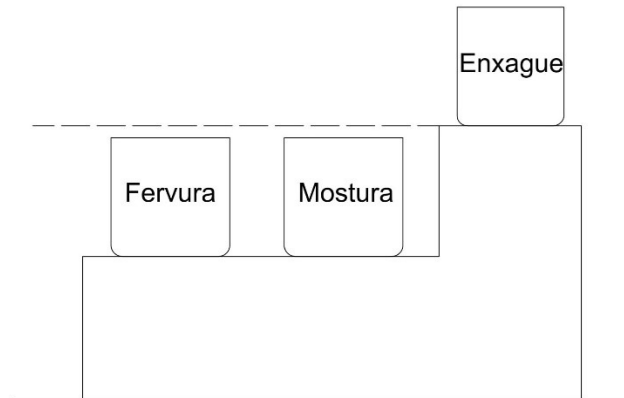
Na montagem da estrutura física de sustentação, em primeiro momento elaborou-se um esboço da mesma antes de sua execução para o dimensionamento correto conforme a dimensão dos caldeirões. A figura 25 apresenta o esboço para a confecção da estrutura.

**Figura 25 – Esboço do fogão cervejeiro**

**Fonte: Autoria própria**

Este esboço trata-se do esboço da perspectiva isométrica do fogão cervejeiro onde suas principais medidas estão apresentadas em milímetros. A escolha pelas dimensões teve como referência além do diâmetro dos caldeirões, a altura necessária para que se possa trabalhar na supervisão e operações manuais de maneira ergonômica.

Outro ponto a considerar é que a altura do caldeirão de enxague deve permanecer no nível igual ou superior aos caldeirões de mostura e fervura, para que se possa aproveitar a força da gravidade no momento do enxague do bagaço. Assim a disposição física dos caldeirões é apresentada a seguir pela figura 26 em vista frontal.

**Figura 26 – Vista frontal da disposição física dos caldeirões**

**Fonte: Autoria própria**

A linha pontilhada refere-se ao nível em que o caldeirão de enxague se encontra, sendo este maior que o nível máximo do caldeirão de mostura, obedecendo então as condições escolhidas. Dessa forma o projeto da estrutura mecânica foi definido e executado para o segmento do trabalho mostrado na figura 27.

**Figura 27 – Fogão cervejeiro**

**Fonte: Autoria própria**



## 4.2 SISTEMA DE ACIONAMENTO DO QUEIMADOR

Para o funcionamento do sistema de queima de gás de maneira automática, foi necessário o desenvolvimento de um sistema com chama piloto. Este sistema é composto por dois elementos principais: o queimador cervejeiro e a chama piloto.

A chama piloto é acionada de forma manual e possui um regulador de fluxo de gás para que fique acesa com baixa intensidade tendo sua funcionalidade de fornecer fogo suficiente para iniciar a combustão do gás que será liberado pelo queimador cervejeiro, conforme a figura 28.

**Figura 28 - Sistema de aquecimento com chama piloto**



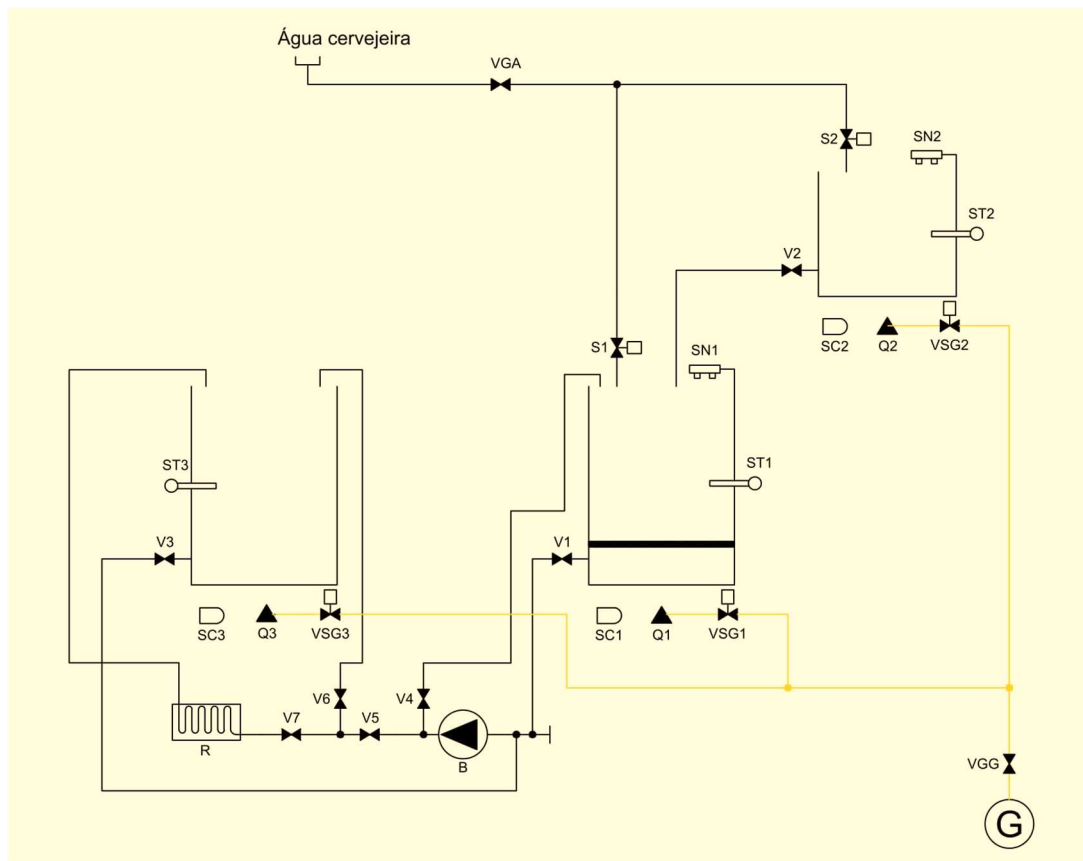
**Fonte: Autoria própria**

Assim, quando um sinal de 12 volts é acionado por parte do controlador, este aciona a válvula solenoide a gás que irá iniciar a combustão principal e aquecer o caldeirão em questão. É pertinente expor que existe um sistema de leitura de chama para a segurança do sistema, caso o sistema não se comporte como o esperado.

### 4.3 PLANTA CERVEJEIRA E PROCESSO DE FABRICAÇÃO

Montada a estrutura mecânica e o sistema de aquecimento, passou-se então para o desenvolvimento da planta cervejeira. Primeiramente foi elaborado um projeto que representa todos os elementos que compõem a planta para o planejamento do controle do processo. A figura 29 mostra a planta cervejeira com seus elementos dispostos de maneira a expor com maior clareza sua funcionalidade.








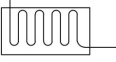



**Figura 29 - Planta cervejeira**



**Fonte: Autoria própria**

Na figura 29 pode-se observar a distribuição dos caldeirões de maneira semelhante ao esboço da figura 26. A simbologia composta na planta é descrita na figura 30.

**Figura 30 - Simbologia da planta cervejeira**

LEGENDA	
	Sensor de nível ultrassônico
	Sensor de temperatura impermeável
	Sensor de chama
	Queimador
	Válvula Solenóide
	Válvula tipo esfera manual
	Bomba
	Resfriador de Placas
	Fonte de água cervejeira
	Fim de linha hidráulica/gás
	Fonte de gás

**Fonte: Autoria própria**

No projeto da planta cervejeira, as linhas na cor preta representam as tubulações hidráulicas que são responsáveis pela condução do fluido no processo. Já as linhas amarelas referem-se as tubulações responsáveis pela condução do gás necessário ao acionamento dos queimadores e chama piloto.

Antes de iniciar o andamento do processo, além de inserir suas as variáveis no controlador por parte do operador, é necessário atender algumas condições. Estas condições são os estados das válvulas tipo esfera manual (V1, V2, V3, V4, V5, V6 e V7) que devem estar todas fechadas com exceção das válvulas V1 e V4.

A planta é composta por uma entrada de água cervejeira que é seguida de uma válvula geral de água (VGA) que deve ser aberta manualmente antes de iniciar o processo junto com a válvula geral de gás (VGG) e sua chama piloto acesa. Após a válvula geral de água, existe uma conexão tipo “T” que leva esta água para duas

válvulas solenoides de água (S1 e S2) e estas seguem para o caldeirão de mostura e enxague respectivamente. Quando abertas estas permitem a passagem de água e iniciam o enchimento dos caldeirões.

Para o controle do nível de água dos caldeirões, existem os sensores de nível (SN1 e SN2) responsáveis pela leitura do nível de água presente no caldeirão de mostura e enxague respectivamente. Assim quando aberto as válvulas solenoides de água, e o nível definido pelo operador ser atingido, estas válvulas são fechadas automaticamente.

A partir deste ponto é iniciado o aquecimento com a abertura das válvulas solenoides de gás (VSG1 e VSG2) respectivamente. Ao mesmo tempo a bomba centrífuga (B) é acionada para homogeneização da temperatura do fluido. Neste momento o controle das temperaturas é realizado conforme inserido pelo operador.

Quando a temperatura alvo é atingida os a válvula VCG1 é fechada e uma mensagem é mostrada para o operador para que inicie a inserção do malte no caldeirão de mostura. A cada mensagem exibida para o operador, é necessário que confirme a ação por meio de um botão da IHM para dar continuidade ao processo. A temperatura do caldeirão de enxague é controlada em paralelo ao restante do processo de mostura até o momento de sua utilização.

Inserido o malte no caldeirão de mostura, inicia-se a contagem do tempo que é exibida na tela e ao mesmo tempo o controle de temperatura acontece. É pertinente expor que a bomba centrífuga é acionada em todo o momento que o queimador do caldeirão de mostura é ligado para evitar a caramelização e o efeito de Maillard visto no capítulo referente ao método de recirculação contínua.

Assim, após as rampas de temperatura e o tempo de mostura chegarem ao fim (inativação enzimática ou *mash-out*), inicia-se a clarificação, momento este em que os queimadores Q1 e Q2 são desligados definitivamente e uma mensagem na tela é exibida para que a bomba centrífuga seja acionada manualmente até atingir a clarificação desejada.

Com este objetivo cumprido, é necessário que se inicie a transferência do mosto para a panela de fervura e se inicie o enxague. Para dar continuidade ao processo uma mensagem é exibida para o operador que deverá fechar a V4 e abrir

a V5 e V6. Após isso o botão para continuar deve ser pressionado e inicia-se então a transferência. No momento que o nível de mosto começa a ficar abaixo do bagaço, deve-se abrir a V2 para que ocorra o enxague do bagaço.

Terminado a transferência e enxague, o operador pressiona o botão de confirmação na IHM e a bomba centrífuga é desligada e o queimador Q3 é acionado por meio da abertura da VSG3 iniciando o processo de fervura. Atingida a temperatura de fervura a contagem do tempo de forma regressiva é mostrada na tela e quando os tempos de lupulagem iniciarem, a sirene é acionada. Toda etapa de inserção de lúpulo é feita manualmente e deve ser confirmada pelo operador por meio do botão de confirmação para dar continuidade ao processo.

Com a finalização da fervura e lupulagem, inicia-se então o resfriamento. Neste momento é exibida uma mensagem na tela para o operador para fechar V1, V4, V6 e abrir V5, V7 e V3. Após realizada a ação, a tecla confirma deve ser pressionada, a bomba centrífuga é acionada e inicia-se o resfriamento.

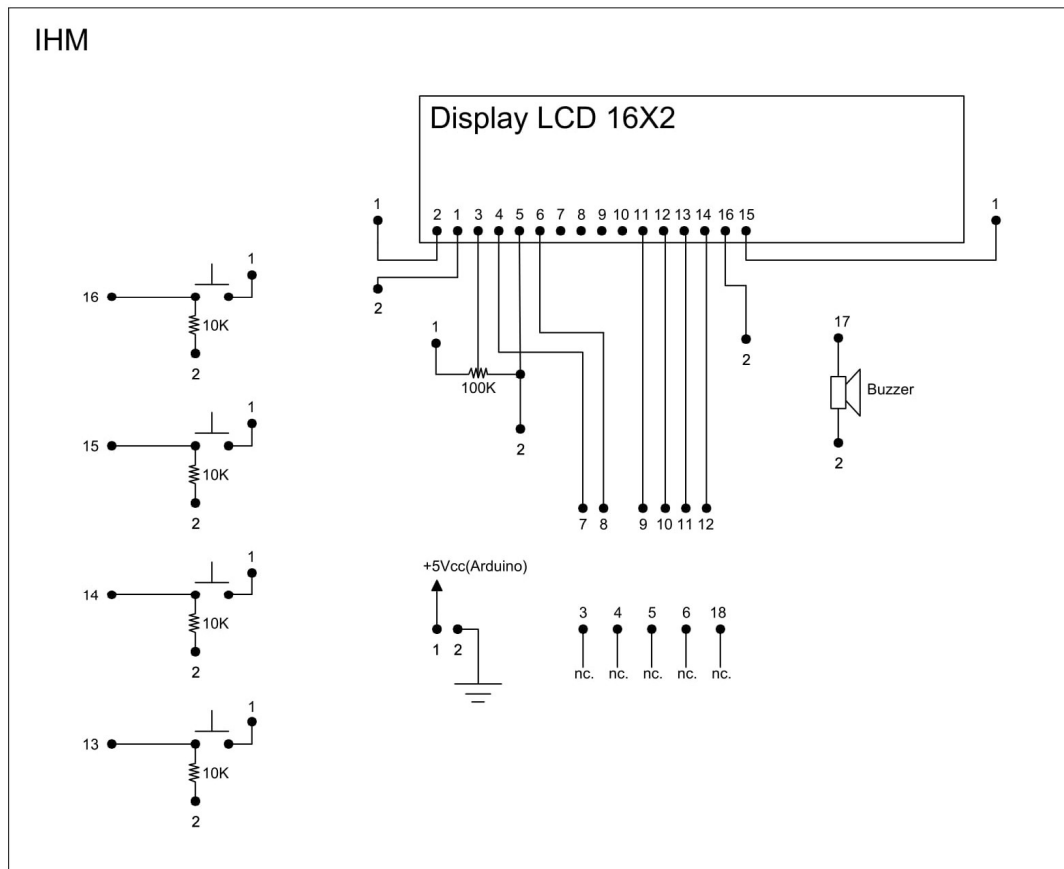
Com a finalização do resfriamento o programa chega ao fim, todos seus atuadores são desligados. Neste momento inicia-se a transferência do mosto para o fermentador de forma manual acionando B pela tecla do painel, e o processo de brassagem chega ao fim.

#### 4.4 CIRCUITO ELETRÔNICO DA IHM

O circuito eletrônico da IHM é constituído por um display LCD, 4 botões de pulso, uma sirene de 5 volts e um potenciômetro. A ligação do display foi embasada na tabela 1 que se referem a pinagem do mesmo que foram conectadas nos bornes de ligação da IHM.

Os botões de pulso possuem resistores de *pull-down* que garantem a leitura correta do barramento por parte da plataforma Arduino e todos os sinais de entrada, saída e alimentação estão ligados diretamente aos bornes da IHM que estão dispostos na figura 31.

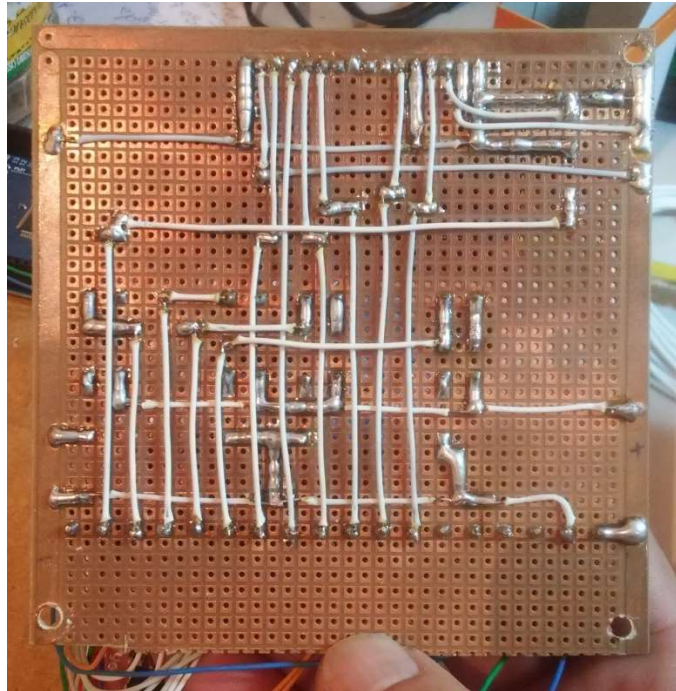
**Figura 31 - Circuito eletrônico da IHM**



**Fonte: Autoria própria**

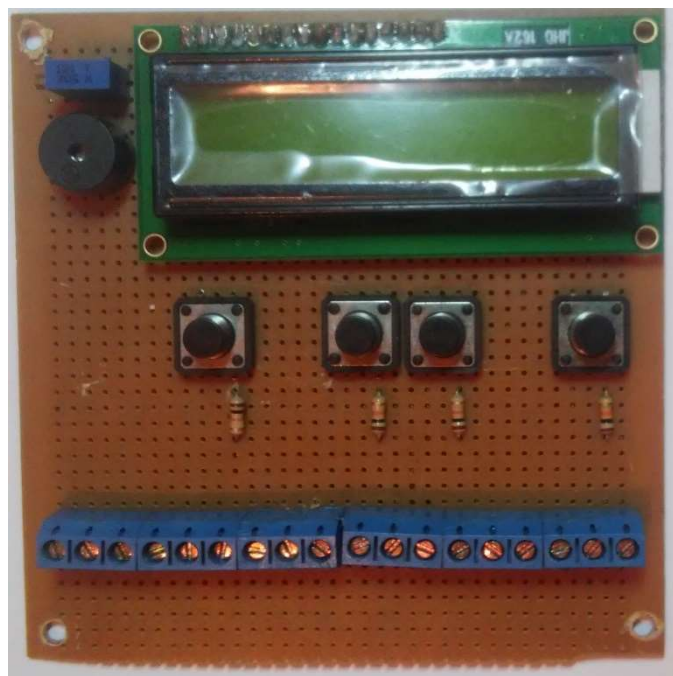
Desenvolvido o circuito em questão iniciou-se sua montagem. Esta foi realizada em uma placa ilhada juntamente com os componentes necessários para sua montagem. A figura 32 e figura 33 mostram respectivamente a montagem da IHM seguindo o circuito projetado na figura 31.

**Figura 32 - Verso da IHM**



Fonte: Autoria própria

**Figura 33 - Frente da IHM**



Fonte: Autoria própria

Pode-se observar que os botões foram distribuídos de forma a obedecer um determinado padrão, sendo este o botão da esquerda como tecla para voltar, o botão central da esquerda como tecla para baixo, o botão central da direita como tecla para cima e o botão da direita como tecla de confirmação ou avanço.

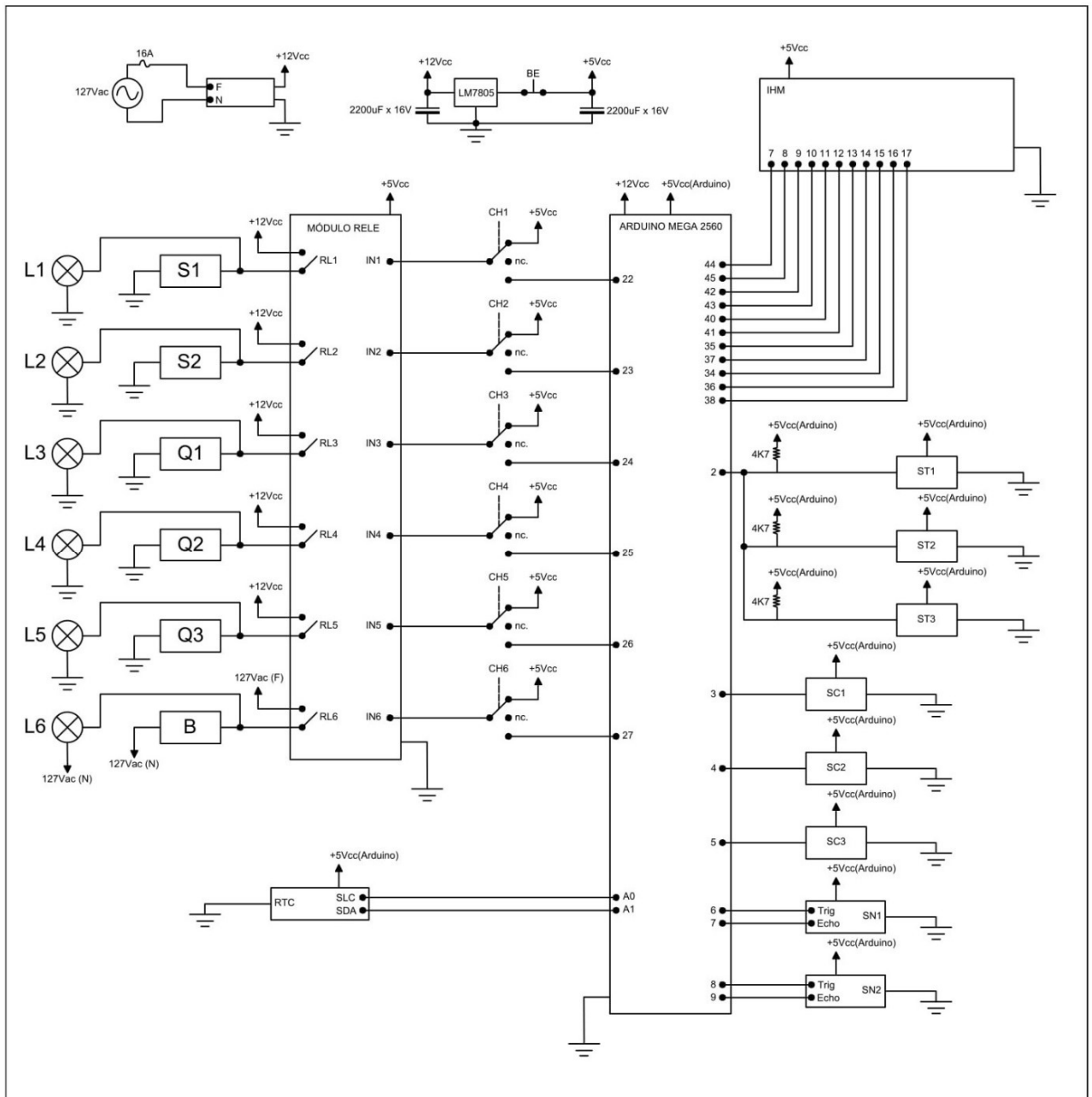
Os bornes de ligação posteriormente foram conectados ao restante do painel, que será abordado a seguir.

#### 4.5 CIRCUITOS ELÉTRICOS DE POTÊNCIA, SUPERVISÃO E CONTROLE

Feita a montagem da IHM, passou-se então para o desenvolvimento físico do painel de controle, fixado em caixa termoplástica, e utilizando-se dos equipamentos descritos nos materiais e métodos. Suas conexões primeiramente foram projetadas conforme a figura 34 a seguir.



Figura 34 - Circuito do Pannel



Fonte: Autoria própria

No circuito do painel é pertinente expor algumas características na sua composição. Em primeiro momento a escolha para a fonte de alimentação segue conforme a corrente utilizada no circuito que é a soma de todas as correntes individuais aproximadas segundo a equação a seguir.

$$I = IS1 + IS2 + IQ1 + IQ2 + IQ3 + ILS(nL) + Iarduino + Irele + ihm$$

$$I = 0,5 + 0,5 + 0,5 + 0,5 + 0,5 + 0,1 * 5 + 2 + 0,5 + 0,5$$

$$I = 6 A$$

onde,

- $I$  = Corrente total aproximada (A)
- $IS$  = Corrente da válvula solenoide (A)
- $IQ$  = Corrente da válvula solenoide do queimador (A)
- $ILs$  = Corrente das lâmpadas indicadoras (A)
- $nL$  = Número de lâmpadas indicadoras
- $Iarduino$  = Corrente do Arduino (A)
- $Irele$  = Corrente do modulo rele (A)
- $ihm$  = Corrente da IHM (A)

Basicamente as correntes individuais do sistema de controle e potência que utilizam a tensão de 12 V resultam em uma corrente total aproximada de 6 A de acordo com a equação da corrente total aproximada. A fonte escolhida foi uma fonte chaveada industrial com 15 A de corrente máxima, esta escolha foi em função da pequena diferença entre custos de uma fonte do mesmo modelo de 10 A.

Referente a bomba centrífuga, esta trabalha em uma tensão alternada de 127 volts, dessa forma não houve a necessidade de conectá-la na fonte de 12 volts e considerar no cálculo da corrente para a utilização da fonte.

O circuito da figura 34 possui como componentes principais a fonte chaveada de 12 volts, um sistema de regulagem de tensão que utiliza um circuito integrado LM7805 junto com capacitores eletrolíticos em sua entrada e saída, que faz a regulagem da tensão de 12 volts para 5 volts e alimenta somente o módulo rele.

É importante observar que em sua saída existe um botão de emergência que abre o circuito que alimenta o módulo rele. Em caso de emergência, quando este botão é pressionado o circuito que alimenta o modulo rele é desligado e todos os atuadores do sistema são desligados. Os atuadores do sistema são as válvulas solenoides de água (S1 e S2), os queimadores (Q1, Q2 e Q3 que são acionados

quando se acionam as válvulas solenoides a gás descritas anteriormente que por consequência acionam os queimadores) e a bomba centrífuga (B).

As chaves tipo alavanca de 3 posições (CH1, CH2, CH3, CH4, CH5 e CH6) são responsáveis por 3 escolhas no momento do controle. Todos os atuadores podem ser acionados manualmente conforme a necessidade, podem ser desligas por completo ou podem estar em modo automático (a plataforma Arduino que faz o controle conforme programado).

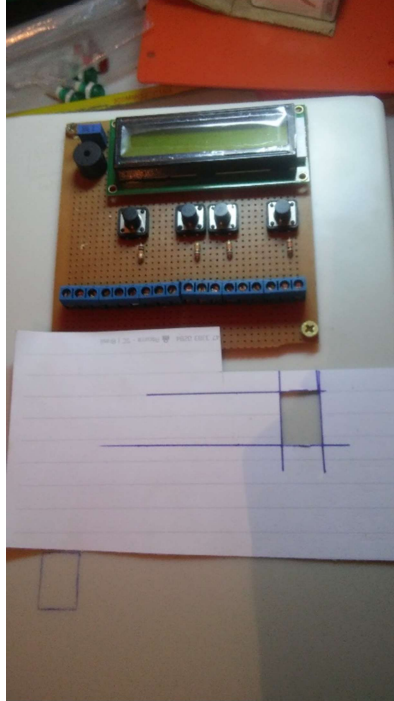
A conexão da IHM no painel de controle também é mostrada na figura 34, onde a correspondência de cada borne da IHM, quando se observa a figura 33, é contada a partir de 1 da esquerda para a direita até o borne número 18. Observa-se que alguns pinos da IHM não são utilizados pois por conveniência de projeto optou-se por bornes reserva ou de expansão.

Para a utilização do módulo RTC, sua alimentação é conectada na saída de 5 volts da plataforma Arduino e os pinos SCL e SDA são conectados nas entradas analógicas A0 e A1. Os sensores de chama são alimentados também pela saída de 5 volts do Arduino e seus bornes ligados individualmente nas entradas digitais.

Da mesma forma ocorre com os sensores de nível, contudo estes utilizam 2 portas digitais para o seu funcionamento. E os sensores de temperatura DS18B20, também são alimentados pela plataforma Arduino, mas todos utilizam apenas uma entrada digital para que se efetue a leitura, pois estes são do tipo *onewire* como descrito no embasamento teórico deste trabalho.

A figura 35, 36, 37 e 38 referem-se especificamente a montagem do painel de controle em questão.

**Figura 35 - Montagem do painel com gabarito**



**Fonte: Autoria própria**

**Figura 36 - Cortes, furos e fixação dos componentes do painel**



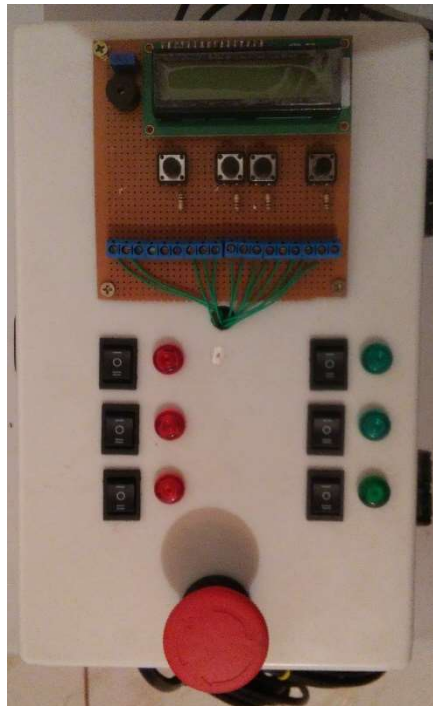
**Fonte: Autoria própria**

**Figura 37 - Cabeamento do painel**



**Fonte: Autoria própria**

**Figura 38 - Painel final**



**Fonte: Autoria própria**

## 4.5 PROGRAMA DESENVOLVIDO

Para o desenvolvimento do programa foi utilizado o *software* descrito no capítulo referente a este. Assim utilizou-se de algumas bibliotecas já desenvolvidas para a programação de determinados periféricos como a biblioteca do módulo RTC e a biblioteca para o sensor DS18B20. É pertinente expor que a linguagem de programação do mesmo é semelhante a programação em C, onde foram criados blocos de instrução para cada etapa que são descritas a seguir.

### 4.5.1 Sistema de seleção dos valores do processo

O sistema de seleção das variáveis do processo foi programado de forma que o operador possa navegar entre as variáveis antes do início do processo, e alterar os valores correspondentes.

Por base este sistema exibe na tela a descrição de cada variável em telas individuais e permite a alteração de valores pelas teclas centrais da IHM desenvolvida. As teclas das bordas irão avançar ou voltar passando de uma tela para a outra. A figura 39 mostra o exemplo da definição da quantidade de água para a mostura.

**Figura 39 - Seleção das variáveis do processo**



**Fonte: Autoria própria**

O programa permite então a escolha das seguintes variáveis definidas conforme cada receita de cerveja, sendo estas:

- Litros de água para mostura
- Litros de água para enxague
- Número de rampas de temperatura
- Tempo de cada rampa de temperatura
- Temperatura de cada rampa
- Tempo de fervura
- Número de lupulagens
- Tempo de cada lupulagem

Vale salientar que a quantidade máxima de rampas permitida pelo programa é de 5 rampas e a mínima é de 2 rampas. Raramente são utilizadas 5 rampas de temperatura em uma receita cervejeira para a mostura e *mash out*, assim optou-se pela limitação de 5 temperaturas. O número mínimo de rampas de temperatura é 2, já que se trabalha com uma receita simples de 1 temperatura de mostura, a segunda temperatura será de *mash out*, sendo assim definido apenas duas rampas de temperatura como mínimo.

A definição máxima de lupulagens tem um máximo de 7 e um mínimo de 1. O máximo de 7 é definido por raramente uma receita cervejeira utilizar mais de 7 lupulagens durante a brassagem e o mínimo deverá ser 1 pois caso não ocorra lupulagem, o produto final não é cerveja.

Definidas as variáveis e as rampas de temperatura, inicia-se então o processo descrito no capítulo referente a planta cervejeira e o processo de fabricação.

#### 4.5.2 Cálculo do nível de água

Para a definição do nível de água nos tanques de mostura e enxague, utilizou-se do sensor ultrassônico que calcula o nível de água presente no caldeirão. Algumas considerações devem ser feitas a respeito do uso deste sensor em sua programação.

Quando a válvula solenoide de água é ligada no caldeirão de mostura e seu enchimento se inicia, o sensor inicia a leitura da altura. O valor que retorna para o microcontrolador corresponde a distância total em centímetros somada com 5 (altura do sensor em relação ao nível do caldeirão). Uma variável foi criada para que se tenha a altura real do líquido, calculada pela subtração de 50 centímetro com o valor da distância medida, subtraído 5 (altura do sensor).

Após esta medição, divide-se o valor da altura real do líquido por 100 para que esta unidade fique em metros. A área da base do caldeirão de mostura é de 0,5 metros tendo uma área de 0,19650 metros quadrados. Então o volume de água presente é dado pela área da base multiplicado pela altura. Para que esta unidade seja dada em litros basta multiplicar o resultado do volume por 1000.

Quando o volume presente no caldeirão for igual ou maior que o valor definido pelo usuário, a válvula solenoide é fechada e inicia-se o enchimento do caldeirão de enxague que funciona da mesma forma, contudo tem seu diâmetro igual a 0,45 metros.

#### 4.5.3 Cálculo dos tempos de rampa

Para o cálculo dos tempos das rampas de temperatura, foi utilizado o módulo RTC junto com sua respectiva biblioteca, que quando o programa se inicia, este módulo tem seu horário zerado. Ao iniciar a contagem de tempo após as temperaturas serem atingidas, o programa faz uma comparação entre o horário atual e o horário que foi definido como início de mostura por exemplo.



O programa faz essa subtração e quando o valor de tempo definido em minutos pelo usuário ser alcançado o programa passa para a próxima etapa. Nos momentos de comparação de tempo, em paralelo o controle das temperaturas de mostura e enxague acontecem.

Caso o valor do relógio passe de uma hora, é feita a subtração das horas e multiplicado por 60 (minutos de uma hora) e somado com o valor dos minutos presentes para o horário de início de rampa e para o horário presente.

Por exemplo, se a mostura atingiu a temperatura alvo no horário 13:43, o tempo em minutos definido pelo operador foi de 80 minutos, teoricamente o horário de fim de rampa será às 15:03. O programa faz o seguinte cálculo: subtrai a hora atual da hora de início de rampa. Este é multiplicado por 60 e tem seu valor armazenado. Após isso o valor dos minutos da hora atual e a hora de início de rampa é subtraído e somado com o valor dos minutos armazenados, tendo então o resultado total em minutos, que é comparado com o tempo que o usuário definiu.

Dessa forma o controle de tempo é realizado e juntamente com o controle de temperatura descrito a seguir.

#### 4.5.4 Leitura de temperatura

Para a realização do controle de temperatura, utilizou-se do sensor DS18B20. O conjunto se compõe de 3 sensores que seus pinos de comunicação ligado diretamente a uma entrada da plataforma Arduino. Antes de iniciar o controle foi necessário a realização da leitura de quantos sensores estiveram presentes no barramento e qual o endereço de cada um deles. Após isto bastou-se apenas chamar o valor da temperatura pelo endereço de cada sensor, tendo então o valor da temperatura no caldeirão.

O controle realizado é um controle do tipo *on-off* que liga o queimador e desliga quando a temperatura alvo é atingida. Vale salientar que se utilizou uma faixa de inativação do queimador em 1°C que podem ser ajustadas apenas no programa e não pelo operador.

Esta faixa serve para que o queimador não fique ligando e desligando repentinamente prejudicando o sistema. Por exemplo, se a temperatura alvo é de 65°C, o queimador permanece ligado até o sensor efetuar a leitura deste valor e desligar o sistema. Este iria ligar o queimador quando o valor da temperatura diminuir para qualquer valor abaixo de 65°C (64,99 por exemplo), mas como definiu-se a faixa de inativação de 1°C, então o queimador só irá ligar quando o valor do sensor for menor que 64°C. Assim o controle de temperatura *on-off* é realizado seguido pelo controle de presença de chama.

#### 4.5.6 Sistema de segurança da chama

O sistema de segurança de chama é baseado no sinal enviado pelo sensor de chama. Quando há presença de chama este sensor envia um sinal lógico de nível alto para a plataforma Arduino que faz a leitura e toma as ações programadas.

Assim no momento em que o queimador em questão tem sua saída de gás aberta, é feito uma leitura no sinal do sensor de chama. Neste momento ocorre um *delay* de 2 segundos e se após isso não houver presença de chama, o sistema entra em alarme por segurança e todos os atuadores são desligados e uma mensagem é exibida na tela. O processo só terá continuidade quando o erro (por exemplo chama piloto apagada) for corrigido. Caso ocorra a presença de chama após o delay de 2 segundos, o sistema continua funcionando corretamente.

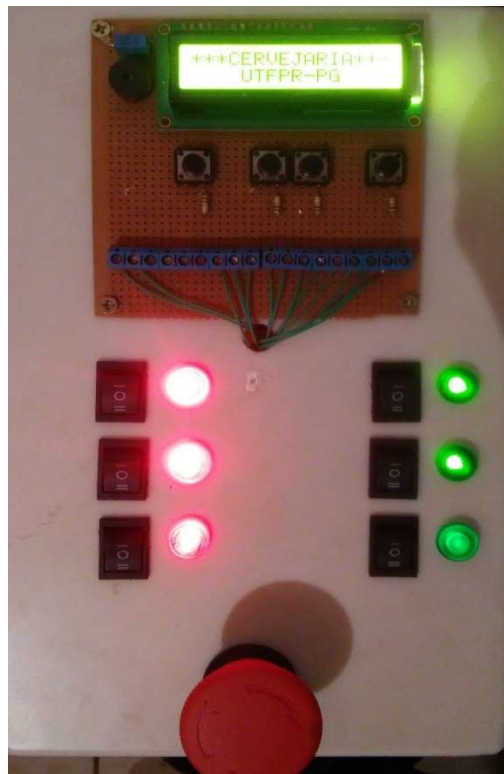
#### 4.5.7 Programação do processo em andamento

A programação geral das ações dos atuadores e mensagens na tela correspondem a descrição projetada no controle de processo descrito exatamente como o capítulo de planta cervejeira e processo de fabricação. Parte da programação foi disponibilizada e pode ser vista no Apêndice A deste trabalho.

## 5. RESULTADOS

Com o desenvolvimento da metodologia apresentada anteriormente os resultados foram satisfatórios em relação ao objetivo do trabalho em questão. A figura 40 apresenta o painel de controle em funcionamento mostrando também sua disposição física dos botões e indicadores.

**Figura 40 - Painel de controle em funcionamento**



**Fonte: Autoria própria**

Outro ponto importante a expor nos resultados foram os custos do sistema que estão apresentados na tabela 3.

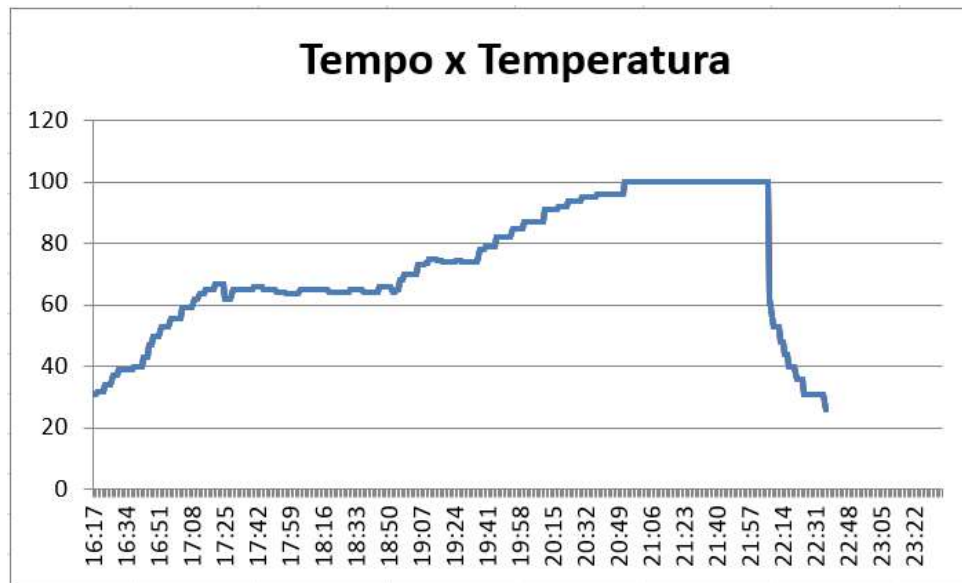
Tabela 3 – Custos totais do sistema

Item	Quantidade	Valor Unitário	Valor Total
Caldeirão inox 98 litros	2	450,00	900,00
Caldeirão alumínio 62 litros	1	220,00	220,00
Fundo falso inox	1	300	300,00
Fogão cervejeiro (estrutura)	1	550,00	550,00
Fogareiro cervejeiro	3	120,00	360,00
Botijão de gás	1	180,00	180,00
Válvula manual 3/4" e conexões	3	140,00	420,00
Tubulação atóxica 10 metros	1	100,00	100,00
Chiller de placas	1	600,00	600,00
Kit chama piloto	3	30,00	90,00
Arduino Mega 2560	1	80,00	80,00
Display LCD	1	20,00	20,00
Módulo rele 8 canais	1	80,00	80,00
Sensor de fogo	3	20,00	60,00
Sensor de temperatura	3	20,00	60,00
Sensor de ultrassônico	1	35,00	35,00
Válvula solenóide para gás	3	90,00	270,00
Válvula solenóide para água	2	60,00	120,00
Fonte de alimentação	1	80,00	80,00
Caixa para central	1	80,00	80,00
Diversos	1	50,00	50,00
Sistema chama piloto	3	120,00	360,00
		Total	5015,00

Fonte: Autoria própria

O sistema foi submetido a testes e teve um monitoramento de temperatura de forma manual com auxílio de um termômetro comum de bulbo de mercúrio no caldeirão de mostura que tiveram seus resultados exibidos no gráfico 1.

Gráfico 1 – Temperatura em relação ao tempo do processo



Fonte: Autoria própria

Visto o gráfico 1, observa-se uma queda de temperatura próxima do horário 17:25 que é o momento da inserção de malte no caldeirão. Este gráfico é resultado das medições de temperatura de uma receita de brassagem de uma cerveja estilo *american pale ale* que teve apenas duas rampas de temperatura, sendo elas de 65°C e de 76°. O tempo de permanência na primeira rampa foi de 75 minutos e na segunda de 10 minutos.

## 6. CONCLUSÕES

Muitas dificuldades foram apresentadas no desenvolvimento da montagem prática do painel, principalmente para a definição do posicionamento interno dos componentes como fonte, módulos e cabeamento.

Houve a necessidade da utilização do circuito integrado regulador de tensão 7805, pois quando utilizando o regulador interno da plataforma Arduino Mega 2560, este aqueceu demasiadamente o que ocasionou a danificação do mesmo. Assim houve a necessidade de substituição da mesma e a utilização do regulador de tensão LM7805 junto com um dissipador de calor.

Outra dificuldade apresentada foi o mal funcionamento do modulo rele que após a inserção do regulador de tensão não funcionou corretamente, então quando inserido dois capacitores na entrada e saída do regulador, o sistema voltou a funcionar de maneira esperada.

Na confecção da IHM, a dificuldade foi referente a soldagem dos componentes e dos fios que tiveram a função de *jumpers*. Além disso pode-se observar que existem 5 bornes que não estão conectados, pois tinham o objetivo de conexão dos sensores, mas por conveniência e melhor distribuição, estes sensores foram conectados diretamente na plataforma Arduino Mega 2560 pela parte superior do painel (furos na carcaça com trava cabo).

Adaptações na estrutura física de sustentação tiveram que ser feitas, como a remoção da tinta na parte que entra em contato com a chama, pois esta em início de processo liberou gases não convenientes a prática. Além disso as conexões de água e gás tiveram que ser adaptadas pelos diferentes diâmetros das extremidades.

Por fim, com este trabalho concluiu-se que o protótipo desenvolvido apresentou resultados satisfatórios cumprindo os objetivos gerais e específicos, permitindo a facilidade da fabricação de cerveja caseira de maneira segura e controlada por parte do cervejeiro artesanal.

## 7. SUGESTÕES PARA FUTUROS TRABALHOS

Para futuros estudos e aperfeiçoamentos deste trabalho, existem inúmeras sugestões. Pode-se melhorar a programação em questão para que aumente o número de rampas de temperatura (caso a receita sugira) e lupulagem se necessário.

Além disso existe a opção de tornar o dia de batelada mais eficiente, elaborando um programa para que possa se iniciar uma nova mostura quando iniciar a fervura da batelada em atual. Assim é possível realizar duas bateladas em menos tempo aumentando a eficiência de produção.

É pertinente sugerir uma programação específica para uma rotina de limpeza dos caldeirões e tubulações para preparar o sistema antes de iniciar uma batelada.

Outra sugestão pertinente é a inserção de válvulas solenoides de esfera rotativa que basicamente proporcionariam toda a autonomia do processo. Assim iriam reduzir as atividades por parte do operador.

Para finalização das sugestões, a troca do display LCD 16X2 por um display de maior de 16X4 ou um display do tipo *touchscreen* ou até mesmo um módulo wifi para controle remoto. Dessa forma seria possível visualizar um maior número de dados e um melhor controle e monitoramento.

## REFERÊNCIAS

- ARTWINE. **Da Mesopotâmia à Modernidade, a Cerveja tem sua História**. Página da Internet. 33ª Edição, 2011. Disponível em <<http://www.artwine.com.br/edicoes/wine-style-33-da-mesopotamia-a-modernidade-a-cerveja-tem-historia.pdf>>. Acesso em 04 de junho de 2017.
- BAMFORTH, C. W. **Beer: a Quality Perspective**. Burlington: Academis Press, 2009.
- BASTOS, D. H. M.; SHIBAO, J.; FERREIRA, E. L.; BOMBO, A. J. **Produtos da reação de Maillard em alimentos**. Nutrire: Revista Sociedade Brasileira de Alimentos e Nutrição, São Paulo, SP, v. 36, n. 3, p. 63-78, dez. 2011.
- BRAGA, N. C. **Relés: Conceitos e Aplicações**. 1ª Edição. Instituto Newton C. Braga. São Paulo. 2012.
- BRASIL. **Dispõe sobre a proteção de informação não divulgada submetida para aprovação da comercialização de produtos e dá outras providências**. Decreto nº 10.603, de 17 de dezembro de 2009. Diário Oficial da República Federativa do Brasil, Brasília, DF, 17 dezembro, 2002.
- BRASIL. **Regulamenta a Lei nº 8.918, de 14 julho de 1994. Dispõe sobre a padronização, a classificação, o registro, a inspeção, a produção e a fiscalização de bebidas**. Decreto nº 6.871, de 04 de junho de 2009. Diário Oficial da República Federativa do Brasil, Brasília, DF, 4 junho, 2009.
- BRIGIDO, R. V. **Produção de Cerveja**. Trabalho de Conclusão de Curso. Universidade Federal de Santa Catarina. Florianópolis, 2006.
- BROUWLAND. **The Brew Bag/Kaggle Bag**. Technical Manual. Baverio, 2017.
- CERVBRASIL. **Anuário 2015**. Associação Brasileira da Indústria da Cerveja, 2015.
- CERVESIA. **A Cerveja e sua História**. Página da Internet, 2017. Disponível em <<https://www.cervesia.com.br/artigos-tecnicos/cerveja/historia-da-cerveja/1-a-cerveja-e-sua-hist%C3%B3ria.html>>. Acesso em 15 de maio de 2017.
- COOPER, T. **DS1307 Real Time Clock Breakout Board Kit**. Guia de Utilização, 2016.
- FERMENTIS. **Safale US-05**. Manual de Utilização, 2017. Disponível em <[http://www.fermentis.com/wp-content/uploads/2013/07/SFA\\_US05\\_PT.pdf](http://www.fermentis.com/wp-content/uploads/2013/07/SFA_US05_PT.pdf)>. Acesso em 20 de agosto de 2017.
- GRAINFATHER. **The Grain Father All Grain Brewing**. Página da Internet. Manual Técnico, 2017. Disponível em <[https://www.grainfather.com/media/wysiwyg/PDFs/Grainfather\\_Connect\\_Instructions.pdf](https://www.grainfather.com/media/wysiwyg/PDFs/Grainfather_Connect_Instructions.pdf)>. Acesso em 15 de julho de 2017.
- HITACHI. **HD44780U Dot Matrix Liquid Cristal Display Controlled/Driver**. Manual Técnico, 1998.



HT-ELEKTRONIC. **Ultraschall Messmodul HC-SR04**. Manual Técnico, Janeiro, 2012.

HUANG, Y.; TIPPMANN J.; BECKER, T. **Kinetic Modeling of Hop Acids during Wort Boiling**. International Journal of Bioscience, Biochemistry and Bioinformatics, Vol. 3, No. 1, Janeiro, 2013.

INCROPERA, F. P. et al. **Fundamentals of Heat and Mass Transfer**. 7. ed. Nova Iorque: John Wiley & Sons, 2007.

JEFFERSON. **Válvulas Solenoides: Informação de Engenharia**. Manual Técnico, 2017.

MANTECH. **Arduino Mega 2560**. Manual Técnico, 2017. Disponível em <<http://www.mantech.co.za/datasheets/products/A000047.pdf>>. Acesso em 22 de agosto de 2017.

MAXIM INTEGRATED PRODUCTS. **DS1307**. Manual Técnico, 2015.

MAXIM INTEGRATED PRODUCTS. **Programmable Resolution 1-Wire Digital Thermometer**. Manual Técnico, 2015.

MCROBERT, M.; [tradução. Rafael Zanolli]. **Arduino Básico**. Editora Novatec. São Paulo, 2011.

MSBEER. **Faça sua Própria Cerva: Curso de Cerveja Artesanal**. Ebook, Curso de Cerveja Artesanal, 2017. Disponível em <<http://msbeerartesanal.com.br/wp-content/uploads/2017/01/E-BOOK-CURSO-DE-CERVEJA-ARTESANAL-MSBEER.pdf>>. Acesso em 22 de julho de 2017.

NBREWCLUB. **Como Fazer Cerveja**. 2017. Imagem da Internet. Disponível em <[http://www.comofazercerveja.com.br/conteudo/view?ID\\_CONTEUDO=11](http://www.comofazercerveja.com.br/conteudo/view?ID_CONTEUDO=11)>. Acesso em 16 de julho de 2017.

OZOYOO. **2 Channel DC 5V Relay Module**. Imagem da Internet, 2017. Disponível em <<http://osoyoo.com/2017/05/27/2-channel-dc-5v-relay-module/>>. Acesso em 25 de agosto de 2017.

PALMER, John J. **Fluid Dynamics: A Simple Key to the Mastery of Efficient Lautering**. 1ª Edição. Natl Book Network, 1995.

PALMER, John J. **How to Brew**. 3ª Edição. Nova Iorque: Brewer Association, 2006.

PAREDE, I. M.; GOMES, L. E. L. **Eletrônica: Automação Industrial**. Manual Técnico. São Paulo. Fundação Padre Anchieta, 2011.

PICCINI, A. R.; Moresco C.; Munhoz L. **Cerveja**. 2002. Página da Internet. Disponível em <<http://www.ufrgs.br/alimentus1/feira/prcerea/ceveja/>>. Acesso em 16 de julho de 2017.

PROCEL. **Bombas: Guia Básico**. Procel Indústria. Edição Seriada, 2009.

SEBRAE. **Potencial de Consumo de Cervejas no Brasil**. Resposta Técnica, 2014.

SERIAL LINK. **Módulo de Relê 4 canais**. Manual Técnico, 2011.

SHENZHEN EONE ELECTRONICS. **Specification for LCD Module 1602A-1**. Manual Técnico, 2005.

SMITH, BRAD. **Brew in a Bag (BIAB) All Grain Beer Brewing**. Imagem da Internet, 2009. Disponível em < <http://beersmith.com/blog/2009/04/14/brew-in-a-bag-biab-all-grain-beer-brewing/>>. Acesso em 12 maio de 2017.

SUPERCOLEÇÃO. **Eisenbahn Pale Ale**. Imagem da Internet, 2014. Disponível em <<https://supercolegao.com/colecoes/embalagens/cerveja/eisenbahn-pale-ale/7062>>. Acesso em 04 de junho de 2017.

THEORYCIRCUIT. **Arduino Flame Sensor Interface**. Página da Internet, 2017. Disponível em <<http://www.theorycircuit.com/arduino-flame-sensor-interface/>>. Acesso em 28 de agosto de 2017.

VENTURI FILHO, W. G. **Tecnologia de Cerveja**. Funep, Jaboticabal, 2000.

## **APÊNDICE A – Programação**

```
//*****  
  
//Carregabibliotecas*****  
  
#include <LiquidCrystal.h>  
#include <RTC1307.h>  
  
//*****  
//Definição dos pinos do display LCD*****  
LiquidCrystal lcd(44, 45, 42, 43, 40, 41);  
//*****  
//Define os pinos *****  
int botao_enter = 36;  
int botao_esc = 35;  
int botao_cima = 34;  
int botao_baixo = 37;  
//int led = 0;  
int buzzer = 38;  
int solenoide_1 = 22;  
int solenoide_2 = 23;  
int bomba = 24;  
int queimador_tina_mostura = 25;  
int queimador_tina_enxague = 26;  
int queimador_tina_fervura = 27;  
//*****
```

```
//Definição das variáveis de entrada e leitura
int valor_botao_enter = 0;
int valor_botao_esc = 0;
int valor_botao_cima = 0;
int valor_botao_baixo = 0;
int valor_geral_tecla = 0;
int sensor_temperatura_tina_1 = 0;
int sensor_temperatura_tina_2 = 0;
int sensor_temperatura_tina_3 = 0;
int hidrometro_eletronico = 0;
//*****
//Definição das variáveis de escolha*****
int litros_mostura = 36;
int litros_enxague = 10;
int numero_rampas = 2;
int temperatura_rampa1 = 65;
int tempo_rampa1 = 60;
int temperatura_rampa2 = 65;
int tempo_rampa2 = 60;
int temperatura_rampa3 = 65;
int tempo_rampa3 = 60;
int temperatura_rampa4 = 65;
int tempo_rampa4 = 60;
int tempo_fervura = 60;
int numero_lupulagem = 2;
int tempo_lupulo1 = 60;
int tempo_lupulo2 = 45;
int tempo_lupulo3 = 30;
int tempo_lupulo4 = 15;
int tempo_lupulo5 = 10;
```

```

int tempo_lupulo6 = 5;

//*****

void setup() {

Serial.begin(9600);

lcd.begin(16, 2); //Define tamanho do display de 16X2

pinMode(botao_enter, INPUT); //Define como entrada

pinMode(botao_esc, INPUT); //Define como entrada pinMode(botao_cima, INPUT); //Define como
entrada pinMode(botao_baixo, INPUT); //Define como entrada //pinMode(led, OUTPUT); //Define
como saída pinMode(buzzer, OUTPUT); //Define pino de saída para buzzer pinMode(solenoide_1,
OUTPUT); //Define como saída pinMode(solenoide_2, OUTPUT); //Define como saída
pinMode(bomba, OUTPUT); //Define como saída pinMode(queimador_tina_mostura,
OUTPUT); //Define como saída pinMode(queimador_tina_enxague, OUTPUT); //Define como saída
pinMode(queimador_tina_fervura, OUTPUT); //Define como saída

}

void buzzer_curto()

{

digitalWrite(buzzer, HIGH);

delay(10);

digitalWrite(buzzer, LOW);

}

void dados_invalidos()

{

digitalWrite(buzzer, HIGH);

delay(500);

digitalWrite(buzzer, LOW);

//Limpa a tela

lcd.clear();

//Posiciona o cursor na coluna 3, linha 0; lcd.setCursor(0, 0);

//Envia o texto entre aspas para o LCD

lcd.print("    DADOS ");

lcd.setCursor(0, 1);

lcd.print("    INVALIDOS    ");

delay(1000);

```

```
}  
  
void tela_inicio_processo()  
{  
    digitalWrite(buzzer, HIGH);  
    delay(500);  
    digitalWrite(buzzer, LOW);  
    //Limpa a tela  
    lcd.clear();  
    //Posiciona o cursor na coluna 3, linha 0; lcd.setCursor(0, 0);  
    //envia o texto entre aspas para o LCD lcd.print(" INICIANDO "); lcd.setCursor(0, 1);  
    lcd.print("    PROCESSO....");  
    delay(3000);  
}  
  
void tela_inicial()  
{  
    //Limpa a tela  
    lcd.clear();  
    //Posiciona o cursor na coluna 3, linha 0; lcd.setCursor(0, 0);  
    //Envia o texto entre aspas para o LCD lcd.print("***CERVEJARIA***"); lcd.setCursor(0, 1); lcd.print(" UTFPR-PG ");  
    delay(1000);  
}  
  
void tela_continua()  
{  
    //Limpa a tela  
    lcd.clear();  
    //Posiciona o cursor na coluna 3, linha 0; lcd.setCursor(0, 0);  
    //Envia o texto entre aspas para o LCD  
    lcd.print("CONTINUAR?");
```

```
lcd.setCursor(0, 1);
lcd.print("ESC  ENTER");
delay(10);
}
void tela_litros_mostura()
{
  BLOCO2:
  //Limpa a tela
  lcd.clear();
  //Posiciona o cursor na coluna 3, linha 0; lcd.setCursor(0, 0);
  //Envia o texto entre aspas para o LCD lcd.print("LITROS MOSTURA?"); lcd.setCursor(7, 1);
  lcd.print(litros_mostura);
  delay(200);
  valor_geral_tecla = 0;
  testa_tecla();
  if(valor_geral_tecla == 1)
  {
    valor_geral_tecla = 0;
    tela_litros_enxague();
    goto CONTINUA2;
  }
  if(valor_geral_tecla == 2)
  {
    valor_geral_tecla = 0;
    tela_continua();
  }
  if(valor_geral_tecla == 3)
  {
    valor_geral_tecla = 0;
    litros_mostura++;
    delay(10);
```



```
goto BLOCO2;
}
if(valor_geral_tecla == 4)
{
valor_geral_tecla = 0;
litros_mostura--;
delay(10);
goto BLOCO2;
}
CONTINUA2:
Serial.println(valor_geral_tecla);
delay(100);
}
void tela_litros_enxague()
{
BLOCO3:
//Limpa a tela
lcd.clear();
//Posiciona o cursor na coluna 3, linha 0; lcd.setCursor(0, 0);
//Envia o texto entre aspas para o LCD lcd.print("LITROS ENXAGUE?"); lcd.setCursor(7, 1);
lcd.print(litros_enxague);
delay(200);
valor_geral_tecla = 0;
testa_tecla();
if(valor_geral_tecla == 1)
{
valor_geral_tecla = 0;
quantidade_rampas();
goto CONTINUA3;
```

```
}  
if(valor_geral_tecla == 2)  
{  
valor_geral_tecla = 0;  
tela_litros_mostura();  
}  
if(valor_geral_tecla == 3)  
{  
valor_geral_tecla = 0;  
litros_enxague++;  
delay(10);  
goto BLOCO3;  
}  
if(valor_geral_tecla == 4)  
{  
valor_geral_tecla = 0;  
litros_enxague--;  
delay(10);  
goto BLOCO3;  
}  
CONTINUA3:  
Serial.println(valor_geral_tecla);  
delay(100);  
}  
void quantidade_rampas()  
{  
BLOCO4:  
//Limpa a tela  
lcd.clear();  
//Posiciona o cursor na coluna 3, linha 0; lcd.setCursor(0, 0);
```

```
//Envia o texto entre aspas para o LCD lcd.print("NUMERO RAMPAS?"); lcd.setCursor(7, 1);  
lcd.print(numero_rampas);  
  
delay(200);  
  
valor_geral_tecla = 0;  
  
testa_tecla();  
  
if(valor_geral_tecla == 1)  
{  
valor_geral_tecla = 0;  
if(numero_rampas < 2)  
{  
//mensagem de erro  
dados_invalidos();  
delay(10);  
goto BLOCO4;  
}  
if(numero_rampas > 4)  
{  
//mensagem de erro  
dados_invalidos();  
delay(10);  
goto BLOCO4;  
}  
temperatura_r1();  
goto CONTINUA4;  
}  
  
if(valor_geral_tecla == 2)  
{  
valor_geral_tecla = 0;  
tela_litros_enxague();  
}  
}
```

```
if(valor_geral_tecla == 3)
{
valor_geral_tecla = 0;
numero_rampas++;
delay(10);
goto BLOCO4;
}
if(valor_geral_tecla == 4)
{
valor_geral_tecla = 0;
numero_rampas--;
delay(10);
goto BLOCO4;
}
CONTINUA4:
Serial.println(valor_geral_tecla);
delay(100);
}
void temperatura_r1()
{
BLOCO5:
//Limpa a tela
lcd.clear();
//Posiciona o cursor na coluna 3, linha 0; lcd.setCursor(0, 0);
//Envia o texto entre aspas para o LCD lcd.print("TEMPERATURA R1?"); lcd.setCursor(4, 1);
lcd.print(temperatura_rampa1); lcd.setCursor(8, 1); lcd.print("GRAUS C");

delay(200);
```

```
valor_geral_tecla = 0;
testa_tecla();
if(valor_geral_tecla == 1)
{
valor_geral_tecla = 0;
tempo_r1();
goto CONTINUA5;
}
if(valor_geral_tecla == 2)
{
valor_geral_tecla = 0;
quantidade_rampas();
}
if(valor_geral_tecla == 3)
{
valor_geral_tecla = 0;
temperatura_rampa1++;
delay(10);
goto BLOCO5;
}
if(valor_geral_tecla == 4)
{
valor_geral_tecla = 0;
temperatura_rampa1--;
delay(10);
goto BLOCO5;
}
```

CONTINUA5:

```
Serial.println(valor_geral_tecla);
```

```
delay(100);
```

```
}
```

```
void tempo_r1()
```

```
{
```

BLOCO6:

```
//Limpa a tela
```

```
lcd.clear();
```

```
//Posiciona o cursor na coluna 3, linha 0; lcd.setCursor(0, 0);
```

```
//Envia o texto entre aspas para o LCD
```

```
lcd.print("TEMPO R1?");
```

```
lcd.setCursor(4, 1);
```

```
lcd.print(tempo_rampa1);
```

```
lcd.setCursor(8, 1);
```

```
lcd.print("MINUTOS");
```

```
delay(200);
```

```
valor_geral_tecla = 0;
```

```
testa_tecla();
```

```
if(valor_geral_tecla == 1)
```

```
{
```

```
valor_geral_tecla = 0;
```

```
temperatura_r2();
```

```
goto CONTINUA6;
```

```
}
```

```
if(valor_geral_tecla == 2)
```

```
{
```

```
valor_geral_tecla = 0;
```

```
temperatura_r1();
```

```
}  
if(valor_geral_tecla == 3)  
{  
valor_geral_tecla = 0;  
tempo_rampa1++;  
delay(10);  
goto BLOCO6;  
}  
if(valor_geral_tecla == 4)  
{  
valor_geral_tecla = 0;  
tempo_rampa1--;  
delay(10);  
goto BLOCO6;  
}  
CONTINUA6:  
Serial.println(valor_geral_tecla);  
delay(100);  
}  
void temperatura_r2()  
{  
BLOCO7:  
//Limpa a tela  
lcd.clear();  
//Posiciona o cursor na coluna 3, linha 0; lcd.setCursor(0, 0);  
//Envia o texto entre aspas para o LCD lcd.print("TEMPERATURA R2?"); lcd.setCursor(4, 1);  
lcd.print(temperatura_rampa2); lcd.setCursor(8, 1); lcd.print("GRAUS C");  
delay(200);  
valor_geral_tecla = 0;  
testa_tecla();  
if(valor_geral_tecla == 1)
```

```
{
valor_geral_tecla = 0;
tempo_r2();
goto CONTINUA7;
}
if(valor_geral_tecla == 2)
{
valor_geral_tecla = 0;
tempo_r1();
}
if(valor_geral_tecla == 3)
{
valor_geral_tecla = 0;
temperatura_rampa2++;
delay(10);
goto BLOCO7;
}
if(valor_geral_tecla == 4)
{
valor_geral_tecla = 0;
temperatura_rampa2--;
delay(10);
goto BLOCO7;
}
CONTINUA7:
Serial.println(valor_geral_tecla);
delay(100);
}
```



```
void tempo_r2()
{
  BLOCO8:
  //Limpa a tela
  lcd.clear();
  //Posiciona o cursor na coluna 3, linha 0; lcd.setCursor(0, 0);
  //Envia o texto entre aspas para o LCD
  lcd.print("TEMPO    R2?");
  lcd.setCursor(4, 1);
  lcd.print(tempo_rampa2);
  lcd.setCursor(8, 1);
  lcd.print("MINUTOS");
  delay(200);
  valor_geral_tecla = 0;
  testa_tecla();
  if(valor_geral_tecla == 1)
  {
    valor_geral_tecla = 0;
    if(numero_rampas >= 3)
    {
      temperatura_r3();
    }
    goto CONTINUA8;
  }
  if(valor_geral_tecla == 2)
  {
    valor_geral_tecla = 0;
    temperatura_r2();
  }
  if(valor_geral_tecla == 3)
  {
```

```
valor_geral_tecla = 0;
tempo_rampa2++;
delay(10);
goto BLOCO8;
}
if(valor_geral_tecla == 4)
{
valor_geral_tecla = 0;
tempo_rampa2--;
delay(10);
goto BLOCO8;
}
CONTINUA8:
Serial.println(valor_geral_tecla);
delay(100);
}
void temperatura_r3()
{
BLOCO9:
//Limpa a tela
lcd.clear();
//Posiciona o cursor na coluna 3, linha 0; lcd.setCursor(0, 0);
//Envia o texto entre aspas para o LCD lcd.print("TEMPERATURA R3?"); lcd.setCursor(4, 1);
lcd.print(temperatura_rampa3); lcd.setCursor(8, 1); lcd.print("GRAUS C");
delay(200);
valor_geral_tecla = 0;
testa_tecla();
if(valor_geral_tecla == 1)
{
```

```
valor_geral_tecla = 0;
tempo_r3();
goto CONTINUA9;
}
if(valor_geral_tecla == 2)
{
valor_geral_tecla = 0;
tempo_r2();
}
if(valor_geral_tecla == 3)
{
valor_geral_tecla = 0;
temperatura_rampa3++;
delay(10);
goto BLOCO9;
}
if(valor_geral_tecla == 4)
{
valor_geral_tecla = 0;
temperatura_rampa3--;
delay(10);
goto BLOCO9;
}
CONTINUA9:
Serial.println(valor_geral_tecla);
delay(100);
```

```
}  
  
void tempo_r3()  
{  
  BLOCO10:  
  //Limpa a tela  
  lcd.clear();  
  
  //Posiciona o cursor na coluna 3, linha 0; lcd.setCursor(0, 0);  
  
  //Envia o texto entre aspas para o LCD  
  
  lcd.print("TEMPO      R3?");  
  
  lcd.setCursor(4, 1);  
  
  lcd.print(tempo_rampa3);  
  
  lcd.setCursor(8, 1);  
  lcd.print("MINUTOS");  
  
  delay(200);  
  
  valor_geral_tecla = 0;  
  
  testa_tecla();  
  
  if(valor_geral_tecla == 1)  
  {  
    valor_geral_tecla = 0;  
    if(numero_rampas >= 4)  
    {  
      temperatura_r4();  
    }  
    goto CONTINUA10;  
  }  
  
  if(valor_geral_tecla == 2)  
  {  
    valor_geral_tecla = 0;
```

```
temperatura_r3();
}
if(valor_geral_tecla == 3)
{
valor_geral_tecla = 0;
tempo_rampa3++;
delay(10);
goto BLOCO10;
}
if(valor_geral_tecla == 4)
{
valor_geral_tecla = 0;
tempo_rampa3--;
delay(10);
goto BLOCO10;
}
CONTINUA10:
Serial.println(valor_geral_tecla);
delay(100);
}
void temperatura_r4()
{
BLOCO11:
//Limpa a tela
lcd.clear();
//Posiciona o cursor na coluna 3, linha 0; lcd.setCursor(0, 0);
//Envia o texto entre aspas para o LCD lcd.print("TEMPERATURA R4?"); lcd.setCursor(4, 1);
lcd.print(temperatura_rampa4); lcd.setCursor(8, 1); lcd.print("GRAUS C");
```

```
delay(200);
valor_geral_tecla = 0;
testa_tecla();
if(valor_geral_tecla == 1)
{
valor_geral_tecla = 0;
tempo_r4();
goto CONTINUA11;
}
if(valor_geral_tecla == 2)
{
valor_geral_tecla = 0;
tempo_r3();
}
if(valor_geral_tecla == 3)
{
valor_geral_tecla = 0;
temperatura_rampa4++;
delay(10);
goto BLOCO11;
}
if(valor_geral_tecla == 4)
{
valor_geral_tecla = 0;
temperatura_rampa4--;
delay(10);
goto BLOCO11;
}
CONTINUA11:
```

```
Serial.println(valor_geral_tecla);
delay(100);
}
void tempo_r4()
{
BLOCO12:
//Limpa a tela
lcd.clear();
//Posiciona o cursor na coluna 3, linha 0; lcd.setCursor(0, 0);
//Envia o texto entre aspas para o LCD
lcd.print("TEMPO      R4?");
lcd.setCursor(4, 1);
lcd.print(tempo_rampa4);
lcd.setCursor(8, 1);
lcd.print("MINUTOS");
delay(200);
valor_geral_tecla = 0;
testa_tecla();
if(valor_geral_tecla == 1)
{
valor_geral_tecla = 0;
goto CONTINUA12;
}
if(valor_geral_tecla == 2)
{
valor_geral_tecla = 0;
temperatura_r4();
}
if(valor_geral_tecla == 3)
{
```

```
valor_geral_tecla = 0;
tempo_rampa4++;
delay(10);
goto BLOCO12;
}
if(valor_geral_tecla == 4)
{
valor_geral_tecla = 0;
tempo_rampa4--;
delay(10);
goto BLOCO12;
}
CONTINUA12:
Serial.println(valor_geral_tecla);
delay(100);
}
void config_1_completa()
{
lcd.clear();
//Posiciona o cursor na coluna 3, linha 0; lcd.setCursor(0, 0);
//Envia o texto entre aspas para o LCD
lcd.print("CONFIG_1  OK!");
lcd.setCursor(0,1);
lcd.print("ESC  ENTER");
delay(100);
}
```



```
void tempo_ferv()
{
    BLOCO14:
    //Limpa a tela
    lcd.clear();
    //Posiciona o cursor na coluna 3, linha 0; lcd.setCursor(0, 0);
    //Envia o texto entre aspas para o LCD lcd.print("TEMPO FERVURA?"); lcd.setCursor(4, 1);
    lcd.print(tempo_fervura); lcd.setCursor(8, 1); lcd.print("MINUTOS");
    delay(200);
    valor_geral_tecla = 0;
    testa_tecla();
    if(valor_geral_tecla == 1)
    {
        valor_geral_tecla = 0;
        num_lupulagens();
        goto CONTINUA14;
    }
    if(valor_geral_tecla == 2)
    {
        valor_geral_tecla = 0;
        goto BLOCO14;
    }
    if(valor_geral_tecla == 3)
    {
        valor_geral_tecla = 0;
        tempo_fervura++;
        delay(10);
        goto BLOCO14;
    }
    if(valor_geral_tecla == 4)
    {
```

```
valor_geral_tecla = 0;
tempo_fervura--;
delay(10);
goto BLOCO14;
}
CONTINUA14:
Serial.println(valor_geral_tecla);
delay(100);
}
void num_lupulagens()
{
BLOCO4:
//Limpa a tela
lcd.clear();
//Posiciona o cursor na coluna 3, linha 0; lcd.setCursor(0, 0);
//Envia o texto entre aspas para o LCD lcd.print("NUM LUPULAGENS?"); lcd.setCursor(7, 1);
lcd.print(numero_lupulagem);
delay(200);
valor_geral_tecla = 0;
testa_tecla();
if(valor_geral_tecla == 1)
{
valor_geral_tecla = 0;
if(numero_lupulagem < 1)
{
//mensagem de erro
dados_invalidos();
delay(10);
goto BLOCO4;
}
}
```

```
if(numero_lupulagem > 6)
{
//mensagem de erro
dados_invalidos();
delay(10);
goto BLOCO4;
}
tempo_lupulagem_1();
goto CONTINUA4;
}
if(valor_geral_tecla == 2)
{
valor_geral_tecla = 0;
tempo_ferv();
}
if(valor_geral_tecla == 3)
{
valor_geral_tecla = 0;
numero_lupulagem++;
delay(10);
goto BLOCO4;
}
if(valor_geral_tecla == 4)
{
valor_geral_tecla = 0;
numero_lupulagem--;
delay(10);
goto BLOCO4;
}
```

```
CONTINUA4:
Serial.println(valor_geral_tecla);
delay(100);
}
void tempo_lupulagem_1()
{
BLOCO8:
//Limpa a tela
  lcd.clear();
//Posiciona o cursor na coluna 3, linha 0; lcd.setCursor(0, 0);
//Envia o texto entre aspas para o LCD
  lcd.print("TEMPO    L1?");
  lcd.setCursor(4, 1);
  lcd.print(tempo_lupulo1);
  lcd.setCursor(8, 1);
  lcd.print("MINUTOS");
  delay(200);
  valor_geral_tecla = 0;
  testa_tecla();
  if(valor_geral_tecla == 1)
  {
    valor_geral_tecla = 0;
    if(numero_lupulagem >= 2)
    {
      tempo_lupulagem_2();
    }
    goto CONTINUA8;
  }
  if(valor_geral_tecla == 2)
  {
```

```
valor_geral_tecla = 0;
num_lupulagens();
}
if(valor_geral_tecla == 3)
{
valor_geral_tecla = 0;
tempo_lupulo1++;
delay(10);
goto BLOCO8;
}
if(valor_geral_tecla == 4)
{
valor_geral_tecla = 0;
tempo_lupulo1--;
delay(10);
goto BLOCO8;
}
CONTINUA8:
Serial.println(valor_geral_tecla);
delay(100);
}
void tempo_lupulagem_2()
{
BLOCO8:
//Limpa a tela
lcd.clear();
//Posiciona o cursor na coluna 3, linha 0; lcd.setCursor(0, 0);
//Envia o texto entre aspas para o LCD
lcd.print("TEMPO    L2?");
lcd.setCursor(4, 1);
lcd.print(tempo_lupulo2);
```

```
lcd.setCursor(8, 1);  
lcd.print("MINUTOS");  
delay(200);  
valor_geral_tecla = 0;  
testa_tecla();  
if(valor_geral_tecla == 1)  
{  
valor_geral_tecla = 0;  
if(numero_lupulagem >= 3)  
{  
tempo_lupulagem_3();  
}  
goto CONTINUA8;  
}  
if(valor_geral_tecla == 2)  
{  
valor_geral_tecla = 0;  
tempo_lupulagem_1();  
}  
if(valor_geral_tecla == 3)  
{  
valor_geral_tecla = 0;  
tempo_lupulo2++;  
delay(10);  
goto BLOCO8;  
}  
if(valor_geral_tecla == 4)  
  
{
```

```

valor_geral_tecla = 0;
tempo_lupulo2--;
delay(10);
goto BLOCO8;
}
CONTINUA8:
Serial.println(valor_geral_tecla);
delay(100);
}
void tempo_lupulagem_3()
{
BLOCO8:
//Limpa a tela
lcd.clear();
//Posiciona o cursor na coluna 3, linha 0; lcd.setCursor(0, 0);
//Envia o texto entre aspas para o LCD
lcd.print("TEMPO    L3?");
lcd.setCursor(4, 1);
lcd.print(tempo_lupulo3);
lcd.setCursor(8, 1);
lcd.print("MINUTOS");
delay(200);
valor_geral_tecla = 0;
testa_tecla();
if(valor_geral_tecla == 1)
{
valor_geral_tecla = 0;
if(numero_lupulagem >= 4)
{
tempo_lupulagem_4();

```

```
}  
goto CONTINUA8;  
}  
if(valor_geral_tecla == 2)  
{  
valor_geral_tecla = 0;  
tempo_lupulagem_2();  
}  
if(valor_geral_tecla == 3)  
{  
valor_geral_tecla = 0;  
tempo_lupulo3++;  
delay(10);  
goto BLOCO8;  
}  
if(valor_geral_tecla == 4)  
{  
valor_geral_tecla = 0;  
tempo_lupulo3--;  
delay(10);  
goto BLOCO8;  
}  
CONTINUA8:  
Serial.println(valor_geral_tecla);  
delay(100);  
}  
void tempo_lupulagem_4()  
{  
BLOCO8:  
//Limpa a tela
```



```
lcd.clear();
//Posiciona o cursor na coluna 3, linha 0; lcd.setCursor(0, 0);
//Envia o texto entre aspas para o LCD
lcd.print("TEMPO    L4?");
lcd.setCursor(4, 1);
lcd.print(tempo_lupulo4);
lcd.setCursor(8, 1);
lcd.print("MINUTOS");
delay(200);
valor_geral_tecla = 0;
testa_tecla();
if(valor_geral_tecla == 1)
{
valor_geral_tecla = 0;
if(numero_lupulagem >= 5)
{
tempo_lupulagem_5();
}
goto CONTINUA8;
}
if(valor_geral_tecla == 2)
{
valor_geral_tecla = 0;
tempo_lupulagem_3();
}
if(valor_geral_tecla == 3)
{
valor_geral_tecla = 0;
tempo_lupulo4++;
delay(10);
goto BLOCO8;
```

```
}  
if(valor_geral_tecla == 4)  
{  
valor_geral_tecla = 0;  
tempo_lupulo4--;  
delay(10);  
goto BLOCO8;  
}  
CONTINUA8:  
Serial.println(valor_geral_tecla);  
delay(100);  
}  
void tempo_lupulagem_5()  
{  
BLOCO8:  
//Limpa a tela  
lcd.clear();  
//Posiciona o cursor na coluna 3, linha 0; lcd.setCursor(0, 0);  
//Envia o texto entre aspas para o LCD  
lcd.print("TEMPO    L5?");  
lcd.setCursor(4, 1);  
lcd.print(tempo_lupulo5);  
lcd.setCursor(8, 1);  
lcd.print("MINUTOS");  
delay(200);  
valor_geral_tecla = 0;  
testa_tecla();  
if(valor_geral_tecla == 1)  
  
{
```

```
valor_geral_tecla = 0;
if(numero_lupulagem >= 6)
{
tempo_lupulagem_6();
}
goto CONTINUA8;
}
if(valor_geral_tecla == 2)
{
valor_geral_tecla = 0;
tempo_lupulagem_4();
}
if(valor_geral_tecla == 3)
{
valor_geral_tecla = 0;
tempo_lupulo5++;
delay(10);
goto BLOCO8;
}
if(valor_geral_tecla == 4)
{
valor_geral_tecla = 0;
tempo_lupulo5--;
delay(10);
goto BLOCO8;
}
```

```
CONTINUA8:
Serial.println(valor_geral_tecla);
delay(100);
}
void tempo_lupulagem_6()
{
BLOCO8:
//Limpa a tela
lcd.clear();
//Posiciona o cursor na coluna 3, linha 0; lcd.setCursor(0, 0);
//Envia o texto entre aspas para o LCD
lcd.print("TEMPO    L6?");
lcd.setCursor(4, 1);
lcd.print(tempo_lupulo6);
lcd.setCursor(8, 1);
lcd.print("MINUTOS");
delay(200);
valor_geral_tecla = 0;
testa_tecla();
if(valor_geral_tecla == 1)
{
valor_geral_tecla = 0;
goto CONTINUA8;
}
if(valor_geral_tecla == 2)
{
valor_geral_tecla = 0;
tempo_lupulagem_5();
}
if(valor_geral_tecla == 3)
```

```
{
valor_geral_tecla = 0;
tempo_lupulo6++;
delay(10);
goto BLOCO8;
}
if(valor_geral_tecla == 4)
{
valor_geral_tecla = 0;
tempo_lupulo6--;
delay(10);
goto BLOCO8;
}
CONTINUA8:
Serial.println(valor_geral_tecla);
delay(100);
}
void iniciar_processo()
{
BLOCO8:
//Limpa a tela
lcd.clear();
//Posiciona o cursor na coluna 3, linha 0; lcd.setCursor(0, 0);
//Envia o texto entre aspas para o LCD lcd.print("INICIAR PROCESS?"); lcd.setCursor(0, 1);
lcd.print("ESC  ENTER?");
delay(100);
}
```

```
void testa_tecla()
{
  delay(10);
  valor_geral_tecla = 0;
  while(valor_geral_tecla == 0)
  {
    valor_botao_enter = digitalRead(botao_enter);
    if(valor_botao_enter == HIGH)
    {
      buzzer_curto();
      valor_geral_tecla = 1;
    }
    valor_botao_esc = digitalRead(botao_esc);
    if(valor_botao_esc == HIGH)
    {
      buzzer_curto();
      valor_geral_tecla = 2;
    }
    valor_botao_esc = digitalRead(botao_cima);
    if(valor_botao_esc == HIGH)
    {
      buzzer_curto();
      valor_geral_tecla = 3;
    }
    valor_botao_esc = digitalRead(botao_baixo);
    if(valor_botao_esc == HIGH)
    {
      buzzer_curto();
      valor_geral_tecla = 4;
    }
  }
}
```

```
}  
delay(10);  
}  
}  
void loop() {  
  INICIO:  
  tela_inicial();  
  digitalWrite(buzzer, HIGH);  
  delay(750);  
  digitalWrite(buzzer, LOW);  
  delay(750);  
  digitalWrite(buzzer, HIGH);  
  delay(750);  
  digitalWrite(buzzer, LOW);  
  delay(750);  
  
  //***** //Programação do menu para  
  inserir dados***** BLOCO2:  
  tela_continua();  
  delay(200);  
  BLOCO1:  
  testa_tecla();  
  delay(200);  
  if(valor_geral_tecla == 1)  
  {  
    tela_litros_mostura(); //Carrega tela de mostura  
    delay(200);  
    goto CONTINUA1;  
  }  
}
```

```
if(valor_geral_tecla == 2)
{
valor_geral_tecla = 0;
goto INICIO;
}
if(valor_geral_tecla == 3)
{
valor_geral_tecla = 0;
goto BLOCO1;
}
if(valor_geral_tecla == 4)
{
valor_geral_tecla = 0;
goto BLOCO1;
}
CONTINUA1:
config_1_completa();
BLOCO4:
testa_tecla();
delay(200);
if(valor_geral_tecla == 1)
{
tempo_ferv(); //Carrega tela de fervura
delay(200);
goto CONTINUA13;
}
if(valor_geral_tecla == 2)
{
valor_geral_tecla = 0;
goto BLOCO2;
```



```
}  
if(valor_geral_tecla == 3)  
{  
valor_geral_tecla = 0;  
goto BLOCO4;  
}  
if(valor_geral_tecla == 4)  
{  
valor_geral_tecla = 0;  
goto BLOCO4;  
}  
CONTINUA13:  
iniciar_processo();  
BLOCO5:  
testa_tecla();  
delay(200);  
if(valor_geral_tecla == 1)  
{  
delay(10);  
goto CONTINUA14;  
}  
if(valor_geral_tecla == 2)  
{  
valor_geral_tecla = 0;  
goto BLOCO2;  
}  
if(valor_geral_tecla == 3)  
{  
valor_geral_tecla = 0;  
goto BLOCO5;
```

```
}  
if(valor_geral_tecla == 4)  
{  
valor_geral_tecla = 0;  
goto BLOCO5;  
}  
CONTINUA14:  
tela_inicio_processo();  
delay(100);  
//goto INICIO;  
//*****  
digitalWrite(solenoide_1, HIGH);  
delay(500);  
digitalWrite(solenoide_1, LOW);  
delay(500);  
digitalWrite(solenoide_1, HIGH);  
delay(500);  
digitalWrite(solenoide_1, LOW);  
delay(500);  
digitalWrite(solenoide_1, HIGH);  
delay(500);  
digitalWrite(solenoide_1, LOW);  
delay(500);  
digitalWrite(solenoide_2, HIGH);  
delay(500);  
digitalWrite(solenoide_2, LOW);  
delay(500);  
digitalWrite(solenoide_2, HIGH);  
delay(500);  
digitalWrite(solenoide_2, LOW);
```

```
delay(500);
digitalWrite(solenoide_2, HIGH);
delay(500);
digitalWrite(solenoide_2, LOW);
delay(500);
digitalWrite(bomba, HIGH);
delay(500);
digitalWrite(bomba, LOW);
delay(500);
digitalWrite(bomba, HIGH);
delay(500);
digitalWrite(bomba, LOW);
delay(500);
digitalWrite(bomba, HIGH);
delay(500);
digitalWrite(bomba, LOW);
delay(500);
digitalWrite(queimador_tina_mostura, HIGH);
delay(500);
digitalWrite(queimador_tina_mostura, LOW);
delay(500);
digitalWrite(queimador_tina_mostura, HIGH);
delay(500);
digitalWrite(queimador_tina_mostura, LOW);
delay(500);
digitalWrite(queimador_tina_mostura, HIGH);
delay(500);
digitalWrite(queimador_tina_mostura, LOW);
delay(500);
```

```
digitalWrite(queimador_tina_enxague, HIGH);
delay(500);
digitalWrite(queimador_tina_enxague, LOW);
delay(500);
digitalWrite(queimador_tina_enxague, HIGH);
delay(500);
digitalWrite(queimador_tina_enxague, LOW);
delay(500);
digitalWrite(queimador_tina_enxague, HIGH);
delay(500);
digitalWrite(queimador_tina_enxague, LOW);
delay(500);
digitalWrite(queimador_tina_fervura, HIGH);
delay(500);
digitalWrite(queimador_tina_fervura, LOW);
delay(500);
digitalWrite(queimador_tina_fervura, HIGH);
delay(500);
digitalWrite(queimador_tina_fervura, LOW);
delay(500);
digitalWrite(queimador_tina_fervura, HIGH);
delay(500);
digitalWrite(queimador_tina_fervura, LOW);
delay(500);

}

//Fim do programa base de escolha e verificação*****
```