



TRABALHO DE CONCLUSÃO DE CURSO

**SOFTWARE DE AUXÍLIO PARA A  
PRODUÇÃO DE CERVEJA ARTESANAL**

**Gabriel Pinheiro Said**

**Brasília, junho de 2019**

**UNIVERSIDADE DE BRASÍLIA**

**FACULDADE DE TECNOLOGIA**

UNIVERSIDADE DE BRASÍLIA  
Faculdade de Tecnologia

TRABALHO DE CONCLUSÃO DE CURSO

**SOFTWARE DE AUXÍLIO PARA A  
PRODUÇÃO DE CERVEJA ARTESANAL**

**Gabriel Pinheiro Said**

*Relatório submetido ao Departamento de Engenharia  
Elétrica como requisito parcial para obtenção  
do grau de Engenheiro Eletricista*

Banca Examinadora

Prof. Ricardo Zelenovsky, ENE/UnB  
*Orientador*

---

Prof. Alexandre Romariz, ENE/UnB  
*Examinador interno*

---

Prof. Daniel Café, ENE/UnB  
*Examinador interno*

---

## **Dedicatória**

*Dedico esse trabalho a todos os que se dedicam ao hobby de produzir cerveja artesanal.*

*Gabriel Pinheiro Said*

## **Agradecimentos**

*Gostaria de agradecer inicialmente a minha família, sem eles eu não estaria aqui hoje.  
À minha namorada, Carine Molz, por todo o suporte e apoio, mesmo nos momentos de mais angústia.  
Ao professor Ricardo Zelenovskypela paciência e ajuda nesta última etapa da graduação.  
À todos os meus amigos e colegas que me acompanharam na vida universitária.*

*Gabriel Pinheiro Said*

---

## RESUMO

As cervejas artesanais estão cada vez mais popularizadas no mercado brasileiro, incentivando muitos apreciadores a se tornarem produtores amadores, fabricando esta bebida em suas próprias casas. Isto se deve, principalmente, a facilidade de informações referentes aos métodos produtivos e a disponibilidade de compra fácil de equipamentos e insumos no mercado. Em contrapartida a estas facilidades, a produção ainda é processo muito manual e demorado. Buscando simplificar e tornar as etapas de fabricação artesanal menos onerosas, este trabalho propõe um software computacional capaz de controlar os equipamentos cervejeiros e acompanhar em tempo real cada fase de produção. Este sistema desenvolvido foi realizado através da confecção de simulador de produção de cerveja e de um software computacional para o controle do equipamento, o software foi programado em Python e o simulador foi construído utilizando Arduino. Por fim, o sistema computacional desenvolvido se mostrou satisfatório em relação ao objetivo inicial, sendo possível a complementação deste com a adição de novas funcionalidades em trabalhos futuros.

**Palavras-chave:** Cerveja Artesanal. Automação. Software Computacional.

---

## ABSTRACT

Handcrafted beers are becoming more popular each year in the Brazilian market, encouraging many of its consumers to become amateur producers and brewing this beverage by their own. This is mainly due to the amount of information regarding the production processes and the availability of equipment in the market. On the other hand, production is still a manual and a time-consuming process. In order to simplify the production and make it more comfortable, this work develop a computational software capable of controlling the brewing equipment and monitoring in real time each phase of production. This work developed a brewing production simulator and a computational software capable of controlling the simulator. The software was programmed in Python and the simulator was built using Arduino. Finally, the automatization system developed was satisfactory in relation to the initial objective, allowing the complementation of this work with the addition of new functions of the software in the future.

# SUMÁRIO

<b>1</b>	<b>Introdução .....</b>	<b>1</b>
1.1	Contextualização.....	1
1.2	Definição do problema.....	2
1.3	Objetivos do projeto .....	2
1.4	Apresentação do manuscrito.....	2
<b>2</b>	<b>Revisão Bibliográfica.....</b>	<b>3</b>
2.1	História da Cerveja e Mercado Atual .....	3
2.2	Cervejas artesanais e etapas de produção.....	4
2.2.1	Malteação.....	5
2.2.2	Moagem.....	5
2.2.3	Brassagem.....	6
2.2.4	Fervura .....	6
2.2.5	Fermentação.....	7
2.2.6	Envase .....	8
2.3	Produtos e trabalhos sobre automatização do processo de produção de cerveja artesanal .....	9
<b>3</b>	<b>Desenvolvimento .....</b>	<b>15</b>
3.1	Introdução .....	15
3.2	Simulador de Produção de Cerveja .....	16
3.2.1	Componentes do Hardware .....	16
3.2.2	Botões.....	18
3.2.3	Software do simulador.....	21
3.3	Software do computador.....	25
3.3.1	Linguagem Python .....	25
<b>4</b>	<b>Resultados Experimentais .....</b>	<b>30</b>
4.1	Introdução .....	30
4.2	Resultados .....	31
<b>5</b>	<b>Conclusões.....</b>	<b>37</b>
	<b>REFERÊNCIAS BIBLIOGRÁFICAS.....</b>	<b>39</b>

<b>Anexos.....</b>	<b>40</b>
<b>I Diagramas Esquemáticos.....</b>	<b>41</b>

# LISTA DE FIGURAS

2.1	Mercado brasileiro de cerveja em números (Sindicerv).....	4
2.2	Moedor de malte manual.....	5
2.3	Exemplo de rampa de temperatura de produção de cerveja.....	6
2.4	Adição de lúpulo na fervura.....	7
2.5	Processo de Fermentação.....	8
2.6	Envase da cerveja.....	8
2.7	Cigarra Elétrica. Fonte: Casa O.L.E.C.....	9
2.8	Sistema HERMS (Campos).....	10
2.9	Esquemático de automação do sistema HERMS.....	11
2.10	Diagrama de blocos de hardware proposto. (Da Rosa, 2015).....	11
2.11	Fluxograma do programa proposto[1].....	12
2.12	Formato da receita lida pelo projeto. (Da Rosa, 2015).....	13
2.13	Formato dos dados gerados (Da Rosa, 2015).....	13
3.1	Diagrama de blocos do hardware proposto.....	15
3.2	Arduino 2560 e a funcionalidade dos pinos.....	16
3.3	Diagrama de Blocos de comunicação.....	17
3.4	Esquemático de pinagem do LCD.....	17
3.5	LCD em funcionamento no simulador.....	18
3.6	Esquemático dos botões.....	19
3.7	Gravação na memo WQ.....	20
3.8	Modos de operação do simulador.....	21
3.9	Tabela de comandos de comunicação.....	24
3.10	Interface gráfica construída com o Tkinter.....	26
3.11	Exemplos de gráficos com Matplotlib.....	27
3.12	Relatório de fervura gerado mediante a leitura dos dados.....	28
3.13	Relatório de fermentação gerado mediante a leitura dos dados.....	29
4.1	Tela inicial do programa.....	31
4.2	Tela de ingredientes da receita.....	31
4.3	Arquivo ReceitaJunho.txt gerado pela tela de ingredientes.....	32
4.4	Tela de parâmetros de brassagem.....	32
4.5	Arquivo BrasJunho.txt gerado pela tela de Brassagem.....	33
4.6	Tela de parâmetros de fervura.....	33



4.7	Arquivo FervJunho.txt gerado pelos parâmetros de fervura .....	33
4.8	Tela de parâmetros de fermentação .....	34
4.9	Arquivo FermJunho.txt gerado pelos parâmetros de fermentação.....	34
4.10	Tela de monitoramento de Brassagem.....	35
4.11	Tela de monitoramento de Brassagem durante o experimento.....	35
4.12	Arquivo BrassagemJunho.txt gerado ao final da brassagem.....	36
I.1	Esquemático Controlador.....	41
I.2	Esquemático Fonte.....	42
I.3	Esquemático Projeto.....	43
I.4	Esquemático Memoria.....	44



# Capítulo 1

## Introdução

### 1.1 Contextualização

Em um mundo cada vez mais industrializado, em que grandes empresas acabam se tornando líderes de seus segmentos, principalmente devido ao seu grande portfólio de produtos e aquisição de novas marcas, uma nova tendência vem surgindo: a de produtos artesanais. Tal situação também ocorre no mercado de cervejas, em que multinacionais, como a Anheuser-Busch Inbev (ABInbev), uma das maiores cervejarias do mundo, vem assistindo o crescimento acelerado da produção desta bebida de forma caseira e artesanal. No mercado brasileiro não poderia ser diferente. Novas marcas de cervejas estão surgindo nos últimos anos, de forma que entre 2008 e 2018, o número de cervejarias regulamentadas foi de 70 a 700, segundo dados do ano 2019 da Associação Brasileira de Cervejas Artesanais (Abracerva).

Este crescimento exponencial pode ser explicado, em grande parte, pela mudança de gosto dos consumidores e, também, pela facilidade de acesso à informação e equipamentos de produção de cerveja. Assim, muitas cervejarias começam com pequenas fabricações caseiras e, só depois, ganham escala comercial. É estimado que somente no Distrito Federal (DF), eram mais de 2 mil produtores caseiros de cerveja em 2017 (Correio Braziliense, 2017)

Apesar de todo esse crescimento, o sistema de produção ainda é muito demorado e manual, o que acaba causando a desistência de alguns cervejeiros, seja por falta de tempo ou pela onerosidade e dificuldade na fabricação. Existem várias etapas processuais para a produção da cerveja, dentre as principais citam-se as fases de Brassagem, Fervura e Fermentação. Estas se relacionam diretamente com o controle de temperatura para definição da qualidade do produto final.

A etapa de Brassagem é definida como o processo de extração de açúcares do malte e aditivos. Esta ocorre a partir da mistura desses produtos com a água, realizando um controle de temperatura de forma a otimizar a extração das enzimas. Após esta fase ocorre a fervura, onde se realiza a adição de lúpulo, que serve como conservante natural e traz o amargor e aroma para a cerveja, e a caramelização do líquido. Por fim, realiza-se a fermentação que é considerada por muitos cervejeiros a etapa mais importante. Neste processo os açúcares são transformados em álcool por meio da ação das leveduras e as principais características da bebida são geradas.

## **1.2 Definição do problema**

Atualmente, com o aumento da produção artesanal de cerveja, muitos equipamentos automatizados de fabricação desta bebida vêm sendo projetados e utilizados no mercado. Porém, grande parte destas ferramentas produtivas ainda não possuem um ambiente de controle virtual, ou seja, um aplicativo ou software computacional. Com isso, este projeto visa desenvolver um software para cobrir esse *gap* do mercado e assim, reduzir ainda mais o esforço empregado pelo cervejeiro artesanal e aumentando o seu conforto durante a produção e possibilitando a repetição de procedimentos.

Em vista deste déficit, vários equipamentos e trabalhos para a automação da produção de cerveja estão surgindo para atender esta demanda. Porém, grande parte destes, não oferecem processos de integração com um ambiente computacional. Um ambiente virtual traz ao processo um melhor monitoramento e propicia um aprendizado acelerado sobre a fabricação desta bebida, dado que torna mais fácil a visualização dos principais indicadores e potenciais pontos de aprimoramento

## **1.3 Objetivos do projeto**

O objetivo desse projeto é programar um software que auxilie o produtor artesanal de cerveja na fabricação desta bebida, comandando os processos produtivos, além de monitorar em tempo real as etapas e gerar arquivos que contemplem toda a fase produtiva. Espera-se que com isso, seja possível aprimorar o controle de temperatura dos processos de produção ao integrar o software com equipamentos de automação de produção de cerveja e reduzir o esforço empregado na fabricação caseira desta bebida. Ainda, conseqüentemente, deseja-se simplificar as etapas de produção, otimizando o tempo do produtor cervejeiro.

## **1.4 Apresentação do manuscrito**

No capítulo 2 será apresentado um apanhado teórico sobre o mercado cervejeiro e o seu processo de fabricação, assim como são discutidos outros trabalhos que serviram como inspiração. Em seguida, o capítulo 3 descreverá a metodologia empregada no desenvolvimento deste projeto. Por fim, os resultados deste estudo serão discutidos no capítulo 4, seguido das conclusões no capítulo 5. Ainda, os anexos irão conter o material complementar relevante.

## Capítulo 2

# Revisão Bibliográfica

### 2.1 História da Cerveja e Mercado Atual

A cerveja é considerada uma das bebidas mais antigas do mundo. Segundo Silva et al (2016), existem alguns registros arqueológicos encontrados na região do Oriente Médio e Egito, datados de cerca de 6000 A.C., que comprovam a existência de uma bebida que seria semelhante a cerveja. Matos (2011) também menciona o consumo desta entre povos antigos, como babilônios e assírios, citando como base a fermentação de cereais. Mediante este fato, a história da cerveja também é associada a história do pão, que assim como esta bebida, compartilha os mesmos ingredientes e cumpria papel importante na base alimentar da população.

A produção cervejeira recebeu maior refinamento a padrões semelhantes aos que a caracterizam hoje a partir da Idade Média (Mega et al., 2011). Neste período, a fabricação desta passou a ser uma atividade especializada associada a um ofício, principalmente em mosteiros, com elaboração e documentação de técnicas específicas de produção. Assim, os monges foram fundamentais para a melhoria da qualidade da cerveja e são considerados os primeiros pesquisadores deste tema (Silva et al., 2016).

Conforme Campos (2017), na Renascença, com o surgimento do regime capitalista e o crescimento populacional, a cerveja passou a ser produzida em uma escala maior para atender a demanda. Devido a esta necessidade produtiva, a indústria cervejeira se consolidou e avançou tecnologicamente até o início do século XX. Quando, nos Estados Unidos, instalou-se uma proibição da bebida no país, provocou uma crise neste mercado. Porém, mesmo com a abolição desta em 1933 a produção de cerveja caseira ainda era proibida. Somente então, em 1979, com a legalização da produção caseira nos Estados Unidos e com a criação de movimentos cervejeiros no Reino Unido, a indústria da cerveja foi retomada e foram popularizados os pequenos produtores artesanais.

Atualmente, a cerveja é a bebida alcoólica mais popular do mundo (Kindersley, 2010). O Brasil é o seu terceiro maior produtor, atrás somente da China e dos Estados Unidos (Sindicerv), e grande parte dessa produção da Ambev, cervejaria brasileira considerada a maior do mundo que possui mais de 60% do mercado de cervejas do Brasil (Banco do Nordeste, 2016).



Figura 2.1: Mercado brasileiro de cerveja em números (Sindicerv)

## 2.2 Cervejas artesanais e etapas de produção

Apesar de todo o poder de mercado das grandes empresas popularmente consolidadas, uma modalidade de produção desta bebida vem ganhando espaço: as cervejas artesanais. Esse segmento está em forte expansão. Se em 2008 existiam aproximadamente 100 cervejarias artesanais no Brasil, atualmente, existem quase 900, além do grande número de cervejeiros caseiros, que é difícil de estimar (Ministério da Agricultura, 2019).

Esse crescimento se deve, em grande parte, a procura dos consumidores por produtos diferenciados e de melhor qualidade. Da mesma forma, a popularização das cervejas artesanais também foi influenciada por outros fatores, como a facilidade de acesso à informação sobre produção de cerveja e aquisição de seus insumos.

O processo de fabricação de cerveja artesanal pode ser dividido nas seguintes etapas: malteação, brasagem, fervura e fermentação. Estas fases processuais serão descritas nos itens abaixo.

### 2.2.1 Malteação

É o processo de germinação de cereais, o qual é executado num curto período de tempo, realizado em instalações apropriadas. Esse processo tem como objetivo a ativação e formação enzimática, transformação das substâncias de reserva do grão de cevada e formação de substâncias corantes e aromatizantes.

### 2.2.2 Moagem

O processo de moagem é a primeira etapa de produção de cerveja em que existe a possibilidade de ser feita por cervejeiros caseiros. Nesse processo, o malte é colocado em um moedor para quebrar a casca da semente e, assim, promover a exposição do amido do endosperma. Como há a necessidade de vir a utilizar as cascas do malte para filtrar o mosto (líquido resultante da mistura de malte com água), o malte, ao passar entre rolos cilíndrico, deve ser esmagado sem que a casca seja totalmente desintegrada (Cruz et al., 2008).

Hoje é possível que o cervejeiro compre o malte moído ou adquira um equipamento de moagem, manual ou automático. Dada a diferença de preço e comodidade, grande parte dos cervejeiros opta pela compra do malte já moído, economizando uma etapa processual. Deste modo, a moagem não será abordada nesse projeto.



Figura 2.2: Moedor de malte manual

### 2.2.3 Brassagem

A etapa de Brassagem, também conhecida como mostura, é o processo em que os grãos de malte são misturados com água para que ocorra a extração enzimática. O nome do líquido açucarado formado ao final dessa mistura é chamado de mosto.

Nesse processo, a extração ocorre por meio do aquecimento da água e do mosto. Porém, para cada temperatura um tipo de enzima diferente atua, portanto rampas de temperatura são uma etapa importante do processo, de forma a controlar a quantidade de açúcares extraídos em cada etapa. Um exemplo desse tipo de rampa pode ser visto na Figura abaixo.

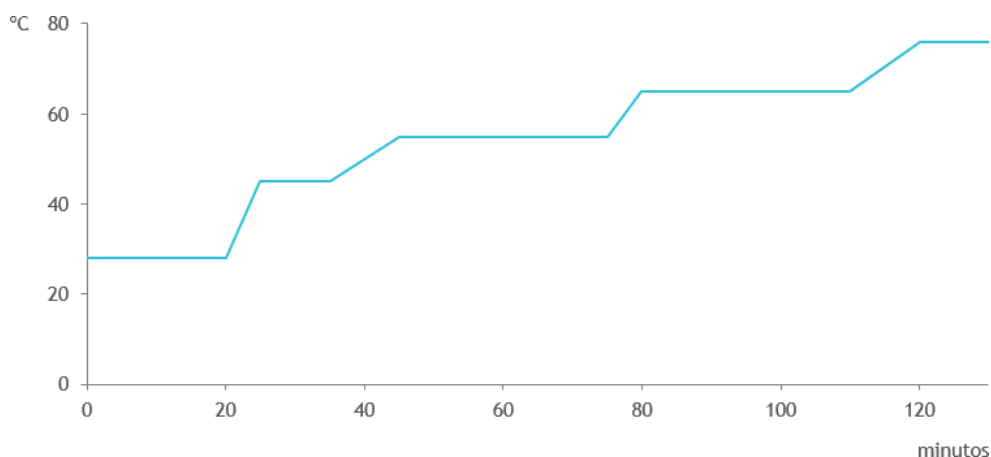


Figura 2.3: Exemplo de rampa de temperatura de produção de cerveja

Ao final dessa etapa ocorre o processo de clarificação, que consiste na separação do mosto e do bagaço (parte sólida), e tem como objetivo a obtenção do máximo em extrato de malte (Ghesti, 2013). Para que ocorra a obtenção do máximo de extrato é feita uma "lavagem" no bagaço, em que, água a 76°C é adicionada e o líquido dessa mistura é adicionado ao mosto.

### 2.2.4 Fervura

A fervura é a etapa em que o mosto é fervido durante um tempo determinado, usualmente, entre 50 e 70 minutos. Também, nesta fase o lúpulo é adicionado na mistura para gerar o amargor e o aroma da bebida assim como, para atuar como conservante natural por meio de seu poder bacteriostático e antioxidante.

Além da adição do lúpulo, essa etapa é responsável pela evaporação da água excedente, coagulação das proteínas, esterilização do mosto, inativação das enzimas, formação de substâncias redutoras e eliminação dos componentes aromáticos indesejados (Ghesti, 2013). Ao final da fervura, o líquido é resfriado por meio de uma serpentina com água gelada. Ao atingir a temperatura ambiente, o mosto está pronto para a etapa de fermentação.





Figura 2.4: Adição de lúpulo na fervura

### 2.2.5 Fermentação

A fermentação é considerada a fase mais importante da produção de cerveja. Esta ocorre após da adição do fermento (levedura) ao mosto em um recipiente esterelizado, quando então a levedura consome os açúcares para proliferar e sobreviver, gerando álcool e  $CO_2$ .

Um dos fatores mais importantes para o controle da fermentação é a temperatura, já que as leveduras são fortemente afetadas por esta variável. Temperaturas muito altas incentivam a produção de álcoois pesados, que podem ter sabores ruins que lembram à solventes, podendo dominar todo o sabor da cerveja. Ademais, o processo de produção de álcoois se dá pela liberação de calor, então é comum que o líquido em fermentação chegue a  $10^\circ C$  mais quente que a temperatura ambiente, por isso, é necessário um bom controle de temperatura. Um exemplo da etapa de fermentação pode ser visto na figura abaixo.



Figura 2.5: Processo de Fermentação

### 2.2.6 Envase

Ao final da etapa de fermentação é realizado o envase, ou seja, o engarrafamento do líquido. Primeiramente, ocorre a esterilização das garrafas, então a cerveja é misturada com o *priming*, que é a adição de uma solução de açúcar no recipiente junto com a cerveja. Essa solução deve ser de fácil consumo para a levedura, então o açúcar passa por processo de inversão, que é a transformação da sacarose em frutose e glicose, por meio da fervura. Assim, quando a garrafa estiver lacrada, a levedura restante na cerveja consome esse açúcar gerando como produto  $CO_2$ .



Figura 2.6: Envase da cerveja

## 2.3 Produtos e trabalhos sobre automatização do processo de produção de cerveja artesanal

Devido à complexidade das etapas de produção é possível encontrar inúmeros equipamentos automatizados que facilitam a produção, além de projetos de controladores de temperatura. Um dos equipamentos automatizados de produção de cerveja mais difundidos no mercado é a Cigarra Elétrica, um equipamento disponibilizado pela loja de produtos cervejeiros Casa OLEC, pensado em produtores caseiros.



Figura 2.7: Cigarra Elétrica. Fonte: Casa O.L.E.C

Segundo a loja, é um equipamento com alto grau de controle no processo e que foi produzido para locais com espaço reduzido, possibilitando sua utilização em espaços exíguos como apartamentos. O controle de temperatura é feito de maneira automática e digital, com até 4 rampas programáveis, mas sem integração com um ambiente computacional. O custo desse produto é de mais de R\$ 6.000,00 para uma produção de 20L, o que acaba se tornando caro para muitos cervejeiros.

Também existem vários projetos de automação do processo de produção de cerveja como o "Projeto e automatização de um sistema HERMS artesanal"(Campos, 2017). Este projeto se baseia no sistema HERMS (*Heat Exchange Recirculating Mash System*) ou sistema de troca de calor por recirculação do mosto. Ele é composto por duas panelas, uma de mosto e outra de água quente que possui uma serpentina submersa. Esse método funciona recirculando o líquido do mosto pela serpentina de forma a ter o controle sobre a temperatura de brassagem (Projeto HERMS, 2011). Esse sistema é considerado bastante seguro, dado que as rampas de temperatura são elevadas bem devagar, por volta de 1°C/min e ele oferece um grau de automação no processo.

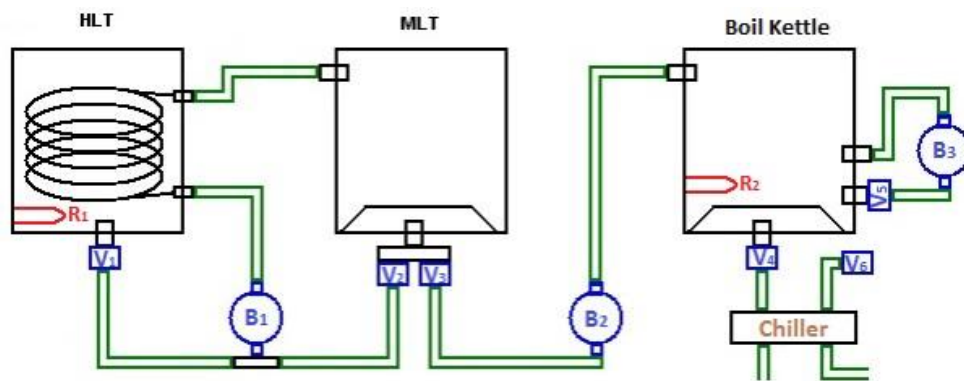


Figura 2.8: Sistema HERMS (Campos, 2017)

Os componentes dispostos na figura 2.8 são:

- HLT (*Hot Liquor Tank*): Panea de água quente com a serpentina
- MLT (*Mash/Lauter Tank*): Panea de brassagem com fundo falso
- *Boil Kettle*: Panea de fervura com fundo falso
- $R_i$ : Resistências
- $V_i$ : Válvulas
- $B_i$ : Bombas elétricas
- Chiller: *Chiller* de placas para o resfriamento

O Sistema HERMS funciona colocando água na panea HLT e e os grãos na panea MLT, a água é então aquecida pela resistência  $R_1$  e controlada por um PID. Então, na temperatura desejada, a água é transferida do HLT para o MLT por meio do acionamento da bomba B1, a partir desse momento ocorre a recirculação do mosto pela panea HLT para que ocorra o controle de temperatura da rampa. Ao fim da brassagem, o mosto é transferido para a panea de *Boil Kettle* por meio da bomba B2 para que ocorra a fervura. Ao fim da fervura, o mosto passa pelo *chiller* de placas para que ocorra o resfriamento antes de iniciar a etapa de fermentação. (Campos, 2017)

O trabalho de Campos (2017) ainda contempla de um sistema com controladores PID (Proporcional - Integral - Derivativo) para a automação do sistema HERMS, como pode ser visto na Figura 2.9. Assim como a Cigarra Elétrica, esse projeto de automação não possui integração com um ambiente virtual e nem um monitoramento de temperatura, sendo necessário o ajuste manual dos controladores a cada produção. Outro ponto a ser notado é o custo do equipamento. Segundo Campos, o valor desse projeto seria de R\$ 5.177,67, o que ainda é um valor bastante próximo da Cigarra Elétrica e pode ser alto para produtores caseiros



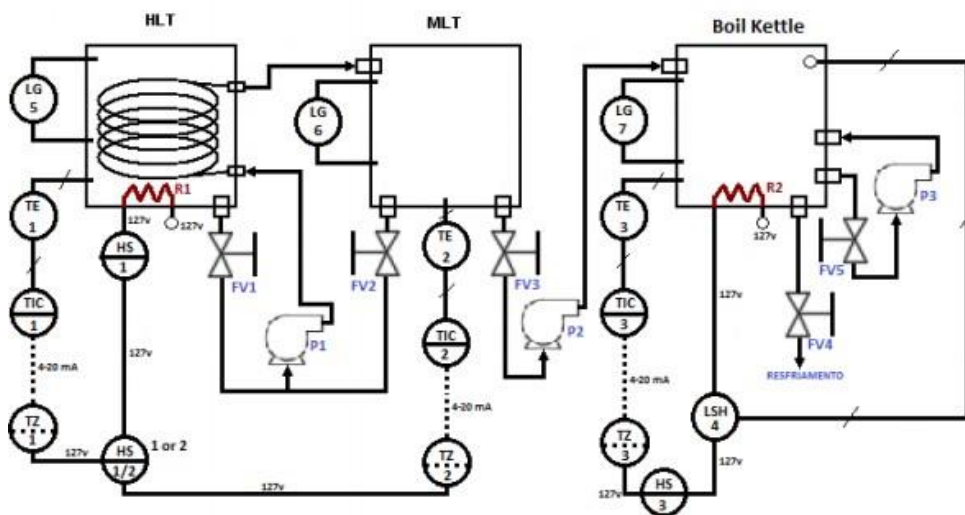


Figura 2.9: Esquemático de automação do sistema HERMS (Campos, 2017)

Por fim, no trabalho de Da Rosa (2015) intitulado "Sistema eletrônico de aquecimento para produção de cerveja artesanal" é proposto um sistema de aquecimento de cerveja de baixo custo, utilizando um Arduino 2560, conforme pode ser visualizado na Figura 2.10. Apesar do projeto possuir um módulo bluetooth, não existe um ambiente computacional para realizar o controle do processo, esse módulo foi utilizado para monitoramento da brassagem via Matlab e para integração com projetos futuros.

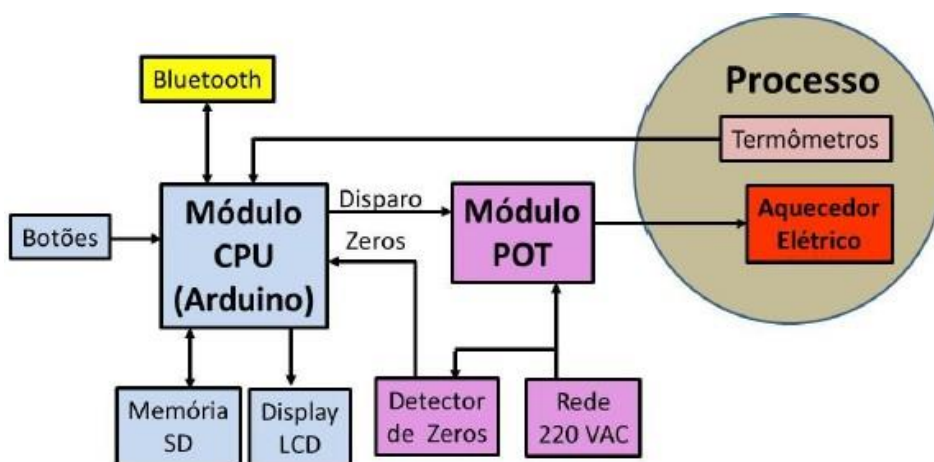


Figura 2.10: Diagrama de blocos de hardware proposto. (Da Rosa, 2015)

Neste trabalho de Da Rosa (2015), o projeto foi executado apenas para a etapa de brasagem. Porém, o autor já apresentava em sua programação a sugestão de sua integração para as etapas de fervura e fermentação, conforme o fluxograma proposto e apresentado na Figura 2.11

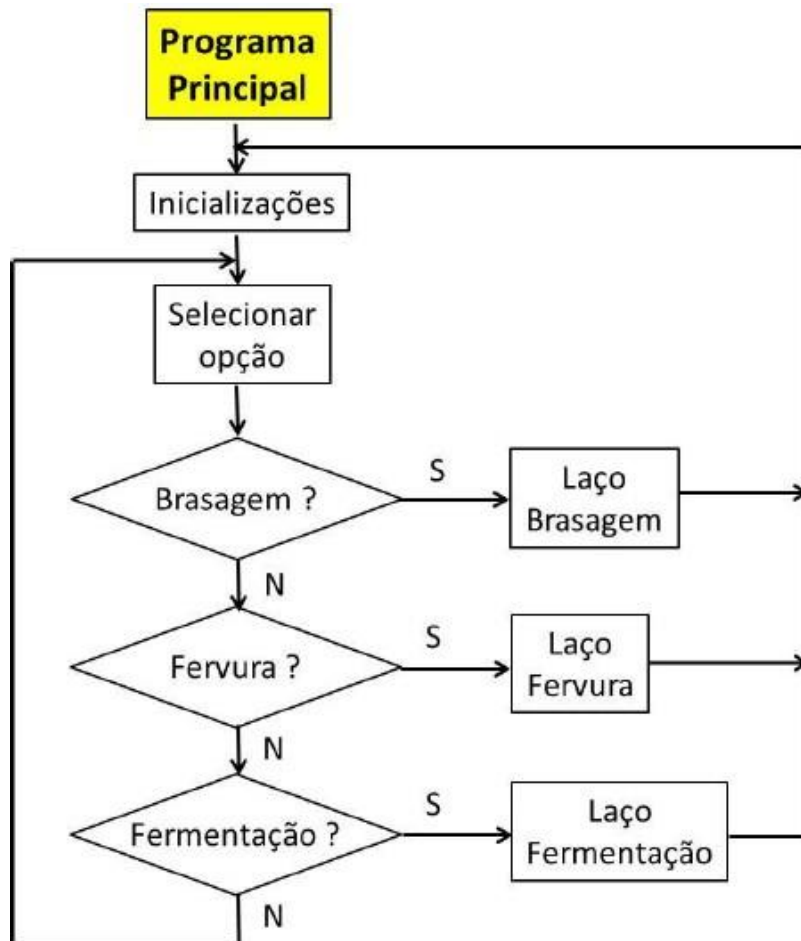


Figura 2.11: Fluxograma do programa proposto (Da Rosa, 2015)

Conforme citado por Da Rosa (2015) em seu trabalho, o laço de brassagem foi a única etapa preparada. Este laço é inicializado lendo um arquivo chamado BRAS.txt e então o sistema inicia a operação realizando o aquecimento do líquido. Durante o aquecimento, o LCD mostra as informações de temperatura e, a cada 1 minuto, é gravado na memória SD um conjunto de informações a ser enviado via Bluetooth para o computador.

```

; 0012 - 10 Dez 2015 - Blonde Ale
; Controlador 11
; Temperatura Tempo
      54      30      ;fase 1 colocar malte
      67      45      ;fase 2
      76      10      ;fase 3 inativ. enzimática
      99      99      ;FIM

```

Figura 2.12: Formato da receita lida pelo projeto. (Da Rosa, 2015)

```

BRASAGEM
dddddd ddd dddd dddd dddd dddd dddd XXXX XXXX XXXX XXXX XXXX XXXX ddd
MIN TOP -SP- MEDIA -T1- -T2- -T3- -KP- ERRO -KI- -IE- -KD- -dE- POT
00002 000 5500 2879 2881 2881 2875 0060 0000 0080 0000 0000 0000 000
00003 000 5500 2879 2881 2881 2875 0060 01A4 0080 0D20 0000 0000 100
00004 000 5500 2896 2888 2881 2919 0060 01A1 0080 0D0F 0000 0000 100
00005 000 5500 3035 3006 3075 3025 0060 018B 0080 0C61 0000 0000 100
00006 000 5500 3173 3150 3219 3150 0060 0175 0080 0BAF 0000 0000 100

```

Figura 2.13: Formato dos dados gerados (Da Rosa, 2015)

Os dados impressos na figura 2.13 são:

- MIN: tempo em minutos contado desde o início da brassagem
- TOP: marca um evento importante (Ex: chegada no *Set Point*)
- SP: indica a meta de temperatura
- MEDIA: média dos termômetros utilizados
- T1: Valor do termômetro 1
- T2: Valor do termômetro 2

- T3: Valor do termômetro 3
- KP: ganho proporcional do regulador PID
- ERRO: erro instantâneo do processo
- KI: ganho integral do regulador PID
- IE: integral do erro
- KD: ganho derivativo do PID
- dE: derivada do erro
- POT: potência calculado pelo PID



## Capítulo 3

# Desenvolvimento

### 3.1 Introdução

O projeto foi realizado em 2 etapas (i) construção de um simulador de produção de cerveja e (ii) construção do software de controle de temperatura da produção.

Levando em conta variáveis como custo e facilidade de reprodução do equipamento cervejeiro, o software desse projeto visa comunicar e monitorar o sistema eletrônico de aquecimento para produção de cerveja artesanal proposto por Da Rosa em 2015.

Conforme citado anteriormente, o projeto de Da Rosa (2015) aborda somente a etapa de brassagem e, além disso, as etapas de produção de cerveja são ainda bastante demoradas, com a duração média da fase de brassagem em cerca de 2 a 3 horas, a fervura cerca de 1 hora e a fermentação dias. Portanto, a construção de um simulador se mostra essencial, já que torna possível o teste do software a qualquer momento e permite a simulação das principais etapas de produção, a saber, brassagem, fervura e fermentação.

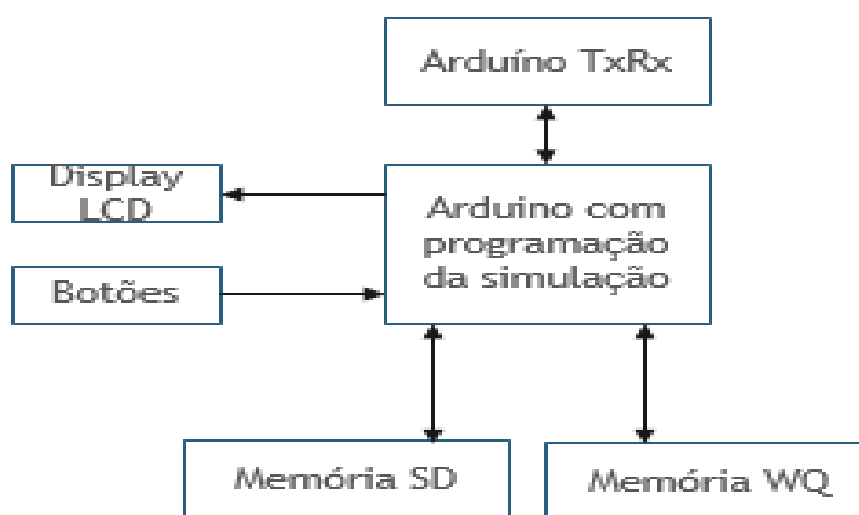


Figura 3.1: Diagrama de blocos do hardware proposto

A Figura 3.1 mostra o diagrama de blocos do simulador deste trabalho. Para a construção do simulador, os principais componentes utilizados foram: dois Arduinos MEGA 2560, um para realizar a comunicação com o computador e outro para funcionar como CPU do programa do simulador, um display LCD, memória SD e memória WQ. Já o software foi construído por meio de linguagem Python, e foram usadas as seguintes bibliotecas: Tkinter, Matplotlib, PySerial.

## 3.2 Simulador de Produção de Cerveja

Conforme discutido no início deste capítulo, este trabalho visa criar um software para o modelo proposto no Sistema Eletrônico de Aquecimento para Produção de Cerveja (Da Rosa, 2015). Com isso, foi construído um hardware utilizando Arduino que simule o comportamento de um equipamento de produção de cerveja automatizado para as principais etapas de produção: brasagem, fervura e fermentação. Esse projeto tem o objetivo refletir os dados e resultados gerados pelo sistema eletrônico de aquecimento. Ainda mais, o sistema tem o intuito de facilitar o teste e o desenvolvimento do software, de forma a permitir uma checagem de suas funcionalidades sem a necessidade de produzir cerveja em todos os testes.

O detalhamento do simulador de automatização do processo de produção de cerveja artesanal, assim como seus componentes, serão explicados detalhadamente nos itens posteriores.

### 3.2.1 Componentes do Hardware

#### 3.2.1.1 Arduino Mega 2560

O Arduino foi criado em 2005 por um grupo de pesquisadores da *Ivrea Interaction Design Institute*. Este é uma placa eletrônica *open source* que consiste de um circuito lógico programável, facilitando a implementação de softwares. Dentre os diversos tipos de produtos Arduino reconhecidos no mercado, optou-se para a execução deste projeto o Arduino Mega 2560.

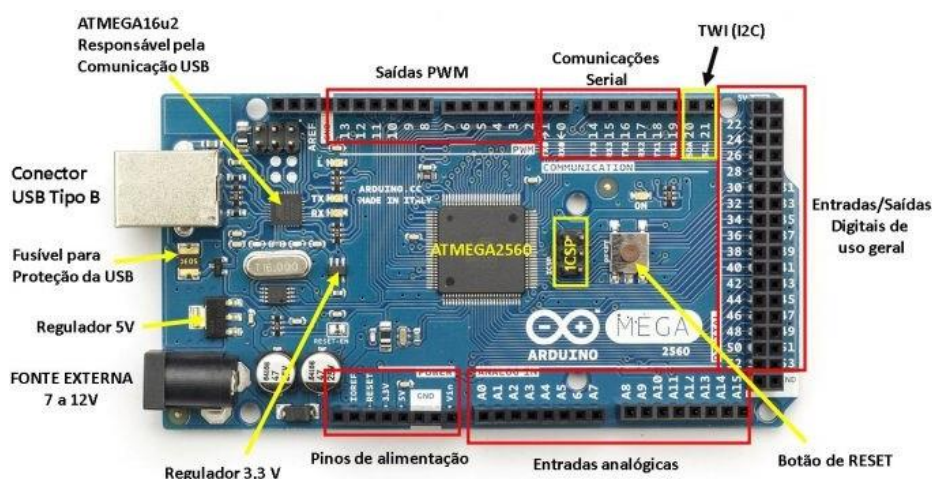


Figura 3.2: Arduino 2560 e a funcionalidade dos pinos

O Arduino Mega 2560 é uma placa baseada no microcontrolador ATmega2560 da ATMEL,

possuindo uma maior quantidade de pinos e de memória FLASH. Este também apresenta recursos como comunicação USB e serial e saídas PWM.

Para o desenvolvimento deste trabalho, um Arduino (TxRx) foi utilizado para simular um módulo de comunicação. O Arduino TxRx recebe por USB os comandos digitados no PC e os envia pela USART2. Tudo o que Arduino TxRx recebe pela USART2 do Arduino controlador, envia por USB para o PC, conforme Figura 3.3. Para pesquisas futuras, sugere-se a troca do Arduino TxRx um módulo bluetooth ou wi-fi.



Figura 3.3: Diagrama de Blocos de comunicação

### 3.2.1.2 Display LCD

O LCD (*Liquid Crystal Display*) é um componente muito útil para projetos, pois permite a visualização dos dados em tempo real e facilita a detecção de erros do programa. Neste estudo utilizou-se o display de 4x20, e durante a simulação da produção, mostrou-se os valores de temperatura de produção para as etapas de brassagem, fervura e fermentação. A conexão de seus pinos pode ser vista na Figura 3.4.

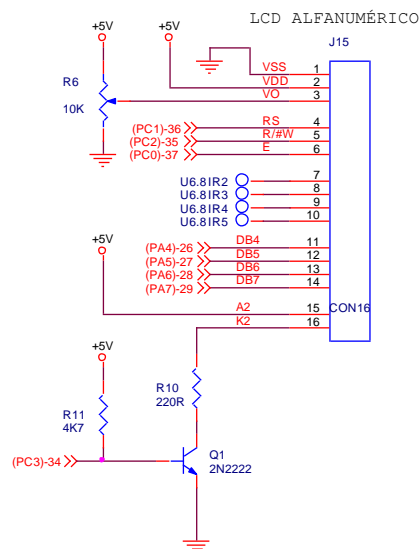


Figura 3.4: Esquemático de pinagem do LCD

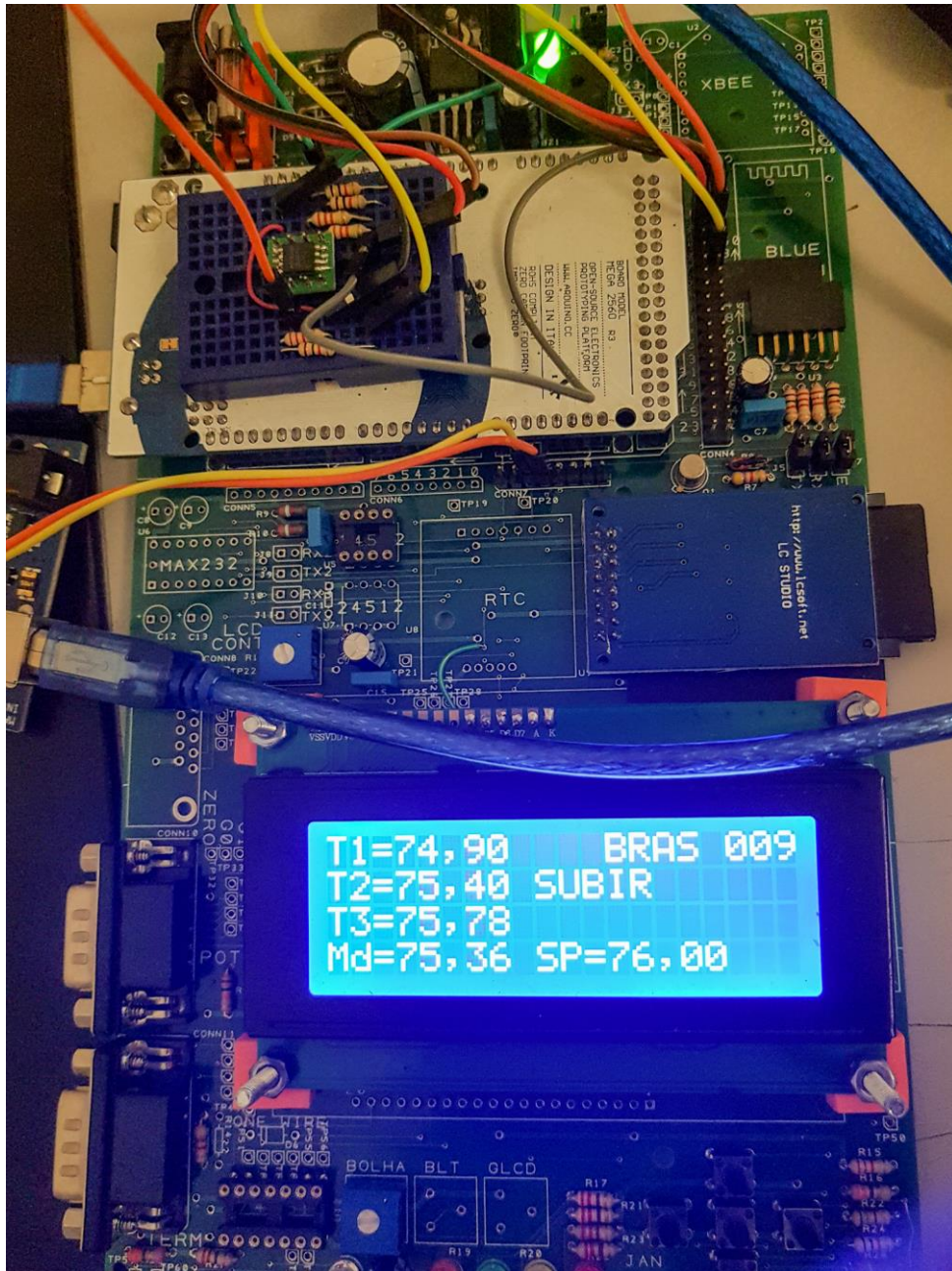


Figura 3.5: LCD em funcionamento no simulador

### 3.2.2 Botões

Neste projeto foram disponibilizados 5 botões para na placa CPU em formato de cruz. Entretanto, para o estudo fez-se necessário apenas a utilização do botão central, funcionando como uma espécie de *enter*, confirmando o recebimento do comando e dos dados, iniciando assim a etapa escolhida. Inicialmente os demais botões eram utilizados para navegar pelos comandos do simulador sem a necessidade de um computador, de forma que, após a integração destes com o simulador, a função desses botões foi desativada. Em um estudo futuro, caso surjam outras necessidades de utilizações do simulador estes botões podem ser necessários.

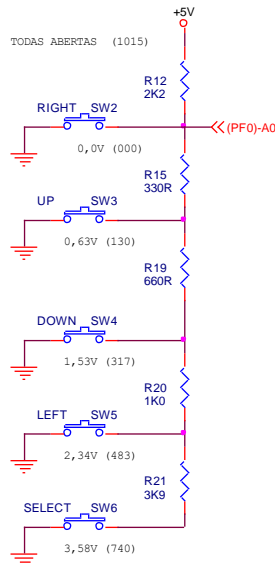


Figura 3.6: Esquemático dos botões

### 3.2.2.1 Memória

No simulador, a memória SD serve como um dispositivo de armazenamento das receitas e resultados das etapas. Neste trabalho, além da memória SD, foi adicionada a memória Flash Winbond W25Q128FV de 16 MBytes, já que ela oferece uma melhor flexibilidade e performance, possui um preço e tamanho acessível para o projeto e alta disponibilidade no mercado nacional quando comparada com outras memórias Flash. A memória foi utilizada para guardar os dados da etapa em execução em binário, tornando assim, mais rápida a comunicação entre o PC e o simulador.

Durante a execução do programa, os dados representativos de temperatura das etapas de brassagem, fervura e fermentação são gravados apenas na memória WQ (Figura 3.7). Ao final de cada etapa, estas informações armazenadas são copiadas para a memória SD, gerando arquivos Bras\_DT.txt, Ferv\_DT.txt e Ferm\_DT.txt (ou arquivos de brassagem, fervura e fermentação).

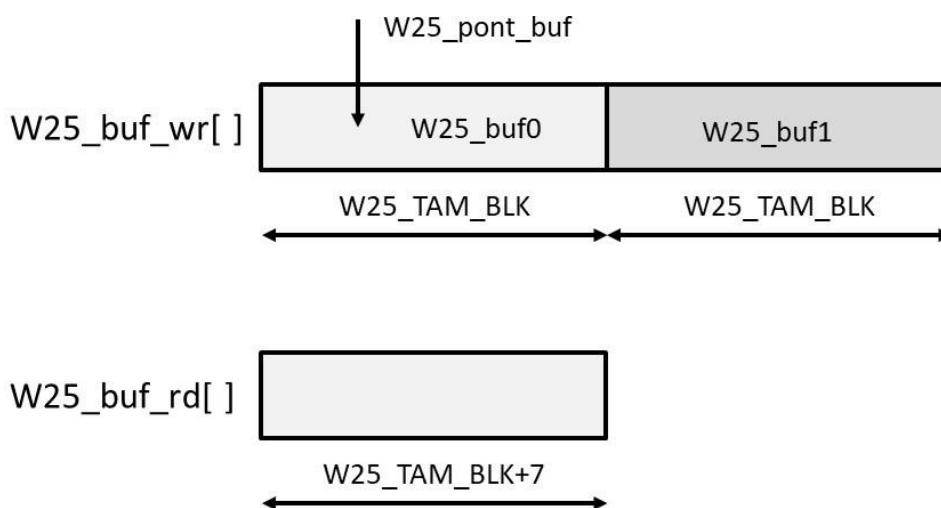


Figura 3.7: Gravação na memória WQ

A gravação na memória WQ usa dois blocos alternadamente, e cada bloco possui o tamanho `W25_TAM_BLK`.

- **char w25\_buf\_wr[2\*W25\_TAM\_BLK]**: Buffer para gravar dados W25Q32, tem o dobro do tamanho
- **int w25\_pont\_buf**: Ponteiro para dentro do buffer.
  - Se `w25_pont_buf == W25_TAM_BLK`, grava a primeira metade do `w25_buf_wr`,
  - Se `w25_pont_buf == 2*W25_TAM_BLK`, grava a segunda metade do `w25_buf_wr`
- **void w25\_wr\_buf(char \*str)**: grava a string terminada em `'\0'` nos `w25_buf0` ou `w25_buf1`, alternadamente. Quando um buffer enche, ele é habilitado para a gravação enquanto o seguinte passa a ser preenchido. Usa um ponteiro para saber onde continuar a gravação.

O envio dos dados segue o empacotamento de tamanho `W25_TAM_BLK+7`. Esses pacotes são criados à medida que `w25_pout` retira dados da memória WQ e escreve a partir da posição 5, primeiro '-'

```
BRAS: #[51 —W25_TAM_BLK—]#
FERV: #[52 —W25_TAM_BLK—]#
FERM: #[53 —W25_TAM_BLK—]#
```

- **char w25\_buf\_rd[W25\_TAM\_BLK+7]**: Buffer para ler W25Q32
- **long w25\_pin**: ponteiro para indicar até onde foi gravado na memo WQ. Sempre aponta para uma posição livre.
- **long w25\_pout**: ponteiro para indicar até onde foi a leitura da memo WQ. Sempre após a última posição lida.
- **w25\_pin**: escreve e depois incrementa, é o endereço da memo WQ.



- **w25\_pout**: lê e depois incrementa, é o endereço da memo WQ. Se  $w25\_pout = w25\_pin$ : nada há para ser lido Se  $(w25\_pin - w25\_pout) > W25\_TAM\_BLK$ : tem um bloco a ser lido e depois transmitido. Na gravação, rotina `w25_wr_buf()` chama `w25_wr()` para gravar um bloco e depois faz  $w25\_pin += W25\_TAM\_BLK$ .
- **Loop BRAS/FERV/FERM**, testa se `ser_dado==FALSE` caso positivo, testa se  $(w25\_pin - w25\_pout) > W25\_TAM\_BLK$ , caso positivo, chama `w25_rd()`, que prepara o buffer e faz `ser_dado=TRUE`; Timer 20 mseg, faz `ser_em_uso=TRUE` e transmite o primeiro byte. ISR serial envia os demais dados e quando chega no último, faz  $w25\_pout += W25\_TAM\_BLK$ , `ser_dado=FALSE` e `ser_em_uso=FALSE`.
- **char bt\_flag\_tx**: fica em 1 enquanto se transmite o bloco lido.
- **int w25\_cont\_rd\_blk**: Contador de blocos lidos para saber o endereço da W25Q32 onde se deve ler.
- **void w25\_rd(int w25\_cont\_rd\_buf)**: usando o contador `w25_cont_rd_buf` calcula o endereço e faz a leitura da WQ e depois dispara a transmissão pela porta serial.

### 3.2.3 Software do simulador

Como o projeto base de Da Rosa (2015) abordava somente a etapa de brassagem, o simulador proposto neste projeto reproduziu rampas de temperatura para as principais etapas de produção de cerveja, ou seja, brassagem, fervura e fermentação. Assim, o software apresentou três modos de operação, cada modo de operação pode ser acessado por meio de um comando específico com uma variável P adicional, sendo esta variável referente a velocidade do passo de incremento ou decrescimento de temperatura.

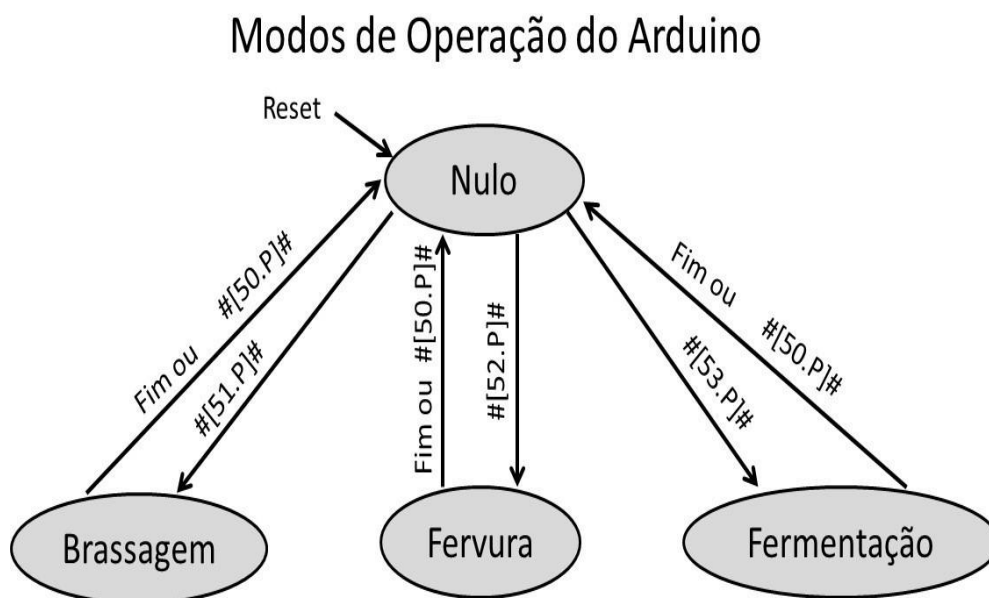


Figura 3.8: Modos de operação do simulador

Para que se possa reproduzir as rampas foi necessário definir uma temperatura inicial. Os valores de temperatura de cada termômetro foram definidos arbitrariamente, conforme descrito:

- Termômetro 1: 20,00 °C
- Termômetro 2: 20,50 °C
- Termômetro 3: 20,88 °C

Vale notar que no programa, os valores de temperatura foram variáveis int, então, o valor de 20,88 °C foi representado em centésimos de graus, ou seja, 2088 de modo a facilitar a transmissão e leitura dos dados. Com isso, as temperaturas foram acrescidas ou decrescidas a depender do passo.

As receitas foram geradas pelo software computacional e apresentaram o nome de BrasData.txt, FervData.txt e FermData.txt, sendo o Data um dado de *input* do software computacional. O formato dos arquivos pode ser visto no Capítulo 4.

### 3.2.3.1 Protocolo de Comunicação

Para a comunicação entre o simulador e o computador foi desenvolvido o protocolo de comunicação abaixo. Nesta grade de comunicação a sequência "#[" serve para alertar ao programa que uma mensagem começou a ser enviada e a sequência "]"# sinaliza o final da mensagem, já os 2 primeiros caracteres após o início sinalizam qual o comando.

Cmdo	PC-Simu	Nome	Descrição	A/R/S
#[00]#	←	NAK	Not Acknowledge. Não entendi a mensagem ou tem erro.	-
#[01]#	←	ACK	Acknowledge. Comando recebido e executado. Fim da execução.	-
#[02]#	→	ALO	Comando 02 (ALO). Não tem ACK. Tem como resposta o comando OLA	-
#[03]#	←	OLA	Comando OLA. É a resposta ao comando 02 (ALO).	-
#[10.LED.ACAO]#	→	LED	LED=(0, 1, 2, 3) qual led e ACAO = (0, 1, 2) apaga, acende, inverte	A
#[11.K]#	→	KQ	Qual o ganho PID, proporcional (K=0), integral (K=1) derivativo (K=2)	R
#[12.NNNNN]#	←	KR	Resposta com valor (NNNNN) atual do Ganho PID, proporcional (K=0), integral (K=1) derivativo (K=2)	-
#[13.K.NNNNN]#	→	KN	Altera valor (NNNNN) para o Ganho PID, proporcional (K=0), integral (K=1) derivativo (K=2)	A
#[14]#	→	SPQ	Qual o valor do Set Point?	R
#[15.NNNN]#	←	SPR	Resposta com o valor do Set Point em Celsius, precisão de 0,01	-
#[16.NNNN]#	→	SPN	Especifica o novo valor (NNNN) para o Set Point, precisão de 0,01	A
#[17]#	→	TPQ	Qual o valor do TOP	R
#[18.NNNNN]#	←	TPR	Resposta com o valor do TOP, sempre com 5 dígitos	-
#[19]#	→	TOP	Incrementa o contador do TOP	A
#[20.L.CC.str]#	→	LCD	Escrever string na linha L (0,1,2,3) coluna CC (0,...,19) do LCD (str=máximo 20 char)	A
#[21.CC.NNN]#	→	BIG	Escrever Big Number, coluna CC (0,...,19) e NNN número → NN,N	A
#[27]#	→	BR_FF_WQ	Apagar arquivo de brasagem da memo WQ. Escreve 0xFF. Ainda não feito.	A



#[30]#	←	BR_RD_SD	Transmitir o arquivo de brasagem "BRAS.txt" da memória SD. Termina com '\0'. Usa função <b>sd_tx()</b> . <b>BR</b> assagem <b>RD</b> ad <b>SD</b> ,)	SA
#[31]#	←	BR_RD_WQ	Transmitir o arquivo de brasagem da memória W25Q32. Termina com '\0'. Usa função <b>wq_tx()</b> . <b>BR</b> assagem <b>R</b> Dead <b>WQ</b>	SA
#[32]#	→	BR_SD_WQ	Copiar o arquivo de brasagem "BRAS.txt" da memória SD → W25Q32. Usa função <b>sd_wq()</b> <b>BR</b> assagem de <b>SD</b> para <b>WQ</b>	A
#[33]#	→	BR_WQ_SD	Copiar o arquivo de brasagem da memória W25Q32 → SD "BRAS.txt". Usa função <b>wq_sd()</b> <b>BR</b> assagem de <b>WQ</b> para <b>SD</b>	A
#[34]#	→	BR_WR_SD	Receber receita brasagem e gravar na memo SD com o nome "BRAS.txt". PC Transmite string. terminada com '\0'. Usa função <b>sd_rx()</b> . <b>BR</b> assagem <b>WR</b> ite <b>SD</b>	SA
#[35]#	→	BR_WR_WQ	Receber arquivo de brasagem e gravar na memória WQ. PC Transmite string terminada com '\0'. Usa função <b>wq_rx()</b> . <b>BR</b> assagem <b>WR</b> ite <b>WQ</b>	SA
#[28]#	→	FV_FF_WQ	Apagar arquivo de Fervura da memo WQ. Escreve 0xFF. Ainda não feito.	A
#[36]#	←	FV_RD_SD	Transmitir o arquivo de fervura "FERV.txt" da memória SD. Termina com '\0'. Usa função <b>sd_tx()</b> . <b>Fer</b> Vura <b>R</b> Dead <b>SD</b>	SA
#[37]#	←	FV_RD_WQ	Transmitir o arquivo de fervura da memória W25Q32. Termina com '\0'. Usa função <b>wq_tx()</b> . <b>Fer</b> Vura <b>R</b> Dead <b>WQ</b>	SA
#[38]#	→	FV_SD_WQ	Copiar o arquivo de fervura "FERV.txt" da memória SD → W25Q32. Usa função <b>sd_wq()</b> <b>Fer</b> Vura de <b>SD</b> para <b>WQ</b>	A
#[39]#	→	FV_WQ_SD	Copiar o arquivo de fervura da memória W25Q32 → SD "FERV.txt". Usa função <b>wq_sd()</b> <b>Fer</b> Vura de <b>WQ</b> para <b>SD</b>	A
#[40]#	→	FV_WR_SD	Receber arquivo de fervura e gravar na memória SD com o nome "FERV.txt". Termina com '\0'. Usa função <b>sd_rx()</b> . <b>Fer</b> Vura <b>WR</b> ite <b>SD</b>	A
#[41]#	→	FV_WR_WQ	Receber arquivo de fervura e gravar na memória WQ. Termina com '\0'. Usa função <b>wq_rx()</b> . <b>Fer</b> Vura <b>WR</b> ite <b>WQ</b>	A
#[29]#	→	FM_FF_WQ	Apagar arquivo de Fermentação da memo WQ. Escreve 0xFF. Ainda não feito.	A
#[42]#	←	FM_RD_SD	Transmitir o arquivo de fermentação "FERM.txt" da memória SD. Termina com '\0'. <b>Fer</b> Mentação <b>R</b> Dead <b>SD</b>	SA
#[43]#	←	FM_RD_WQ	Transmitir o arquivo de fermentação da memória W25Q32. Termina com '\0'. Usa função <b>wq_tx()</b> . <b>Fer</b> Mentação <b>R</b> Dead <b>WQ</b>	SA
#[44]#	→	FM_SD_WQ	Copiar o arquivo de fermentação "FERM.txt" da memória SD → W25Q32. Usa função <b>sd_wq()</b> <b>Fer</b> Mentação de <b>SD</b> para <b>WQ</b>	A
#[45]#	→	FM_WQ_SD	Copiar o arquivo de fermentação da memória W25Q32 → SD "FERM.txt". Usa função <b>wq_sd()</b> <b>Fer</b> Mentação de <b>WQ</b> para <b>SD</b>	A
#[46]#	→	FM_WR_SD	Receber arquivo de fermentação e gravar na memória SD com o nome "FERM.txt". Termina com '\0'. Usa função <b>sd_rx()</b> . <b>Fer</b> Mentação <b>WR</b> ite <b>SD</b>	A
#[47]#	→	FM_WR_WQ	Receber arquivo de fermentação e gravar na memória WQ. Termina com '\0'. Usa função <b>wq_rx()</b> . <b>Fer</b> Mentação <b>WR</b> ite <b>WQ</b>	A
#[50.P]#	→	NULO	De (Bras, Ferv ou Ferv) → Nulo. Envia o parâmetro P (1 dígito) que ainda não sei para que serve.	A
#[51.P]#	→	BRAS	Do Nulo → Brassagem. Envia o parâmetro P (1 dígito) que ainda não sei para que serve.	A

#[51.xxxxxxx]#	←	BRAS	Pacote com W25_TAM_BLK dados da brasagem.	R
#[52.P]#	→	FERV	Do Nulo → Fervura. Envia o parâmetro P (1 dígito) que ainda não sei para que serve.	A
#[52.xxxxxxx]#	←	FERV	Pacote com W25_TAM_BLK dados da fervura.	A
#[53.P]#	→	FERM	Do Nulo → Fermentação. Envia o parâmetro P (1 dígito) que ainda não sei para que serve.	A
#[53.xxxxxxx]#	←	FERM	Pacote com W25_TAM_BLK dados da fermentação.	A
#[54 p]#	→	TESTE	Do Nulo → Teste. Parâmetro P indica qual o teste	
#[55]#	→	QUAL	Qual modo está ativo e qual é o valor do parâmetro?	R
#[56.M.P]#	←	QM	Resposta indicando o modo (0=Nulo, 1=Bras, 2=Ferv e 3=Ferm) e o parâmetro P	-
#[57.P]#	→	DT	Receber os dados (P=1 → só temperatura média e P=0 → tudo)	A
#[70.EEEEE.E. QQQQQQ]#	→	WQRD	Transmite QQQQQQ bytes da memo WQ2532 a partir do endereço EEEEE. Tudo em hexadecimal. <b>WQ ReaD</b> – Usado para fazer testes.	RA
→	→	Reset	Interrompe o modo atual e volta para o modo inicial (RESET)	
←	←	Fim	Fim de uma etapa (por tempo ou porque o usuário interrompeu)	

Figura 3.9: Tabela de comandos de comunicação

- Na Figura acima, na coluna Cmdo, "." significa um espaço. Na coluna PC–Simu, a seta para a direita representa que o PC enviou comando para o simulador e a seta para a esquerda mostra que PC recebeu comando do simulador. Já na coluna A/R/S, os significados são os seguintes:
  - A: responde com ACK
  - R: responde com outro comando
  - S: *string*

A transmissão pela porta serial é compartilhada entre a resposta aos comandos e envio de dados. Um timer de 20 mseg é responsável por iniciar a transmissão. Existem 3 tipos de função de transmissão:

1. Resposta a um comando simples (SER\_RESP\_1): a rotina prepara a resposta no buffer de comando, acerta o tamanho da mensagem e faz ser\_resp\_cmd=SER\_RESP1. Se a porta serial estiver livre, o timer detecta o pedido para resposta a um comando, envia do primeiro dado e tranca a porta serial em uso. O resto acontece por interrupção serial.
2. Resposta a um comando extenso (SER\_RESP\_2) (como a transmissão de uma receita): a rotina envia a receita com o uso de vários pacotes colocados no buffer de comando, e de tamanho no comando de tamanho. A rotina tranca a porta serial em uso e envia do primeiro dado. O timer não participa. A interrupção serial envia o resto dos dados e ao final, não mexe nos *flags*. A rotina acompanha o envio de cada pacote. Ao final de um pacote, inicia o próximo. Após o envio do último byte, a rotina destranca a porta serial. Note que a porta serial fica dedicada ao envio da receita. Nem comando, nem dado são transmitidos.

3. Envio de dados (SER\_DADO): a rotina prepara a resposta no `w25_buf_rd [ ]`, o tamanho é `W25_TAM_BLK +7` e faz `ser_dado=TRUE`. O timer detecta o pedido de transmissão de dado, envia do primeiro dado e tranca a porta serial em uso. O resto acontece por interrupção serial. Por ocasião do último byte, a interrupção serial destranca a porta serial;

Os pacotes de envios de dados possuem o seguinte formato:

```
#[NN ddddd ddd dddd dddd dddd dddd dddd dddd ddd]#
```

Sendo NN o comando principal e "d" é o valor daquela posição para cada uma das colunas.

### 3.3 Software do computador

O software computacional foi construído utilizando a linguagem Python. Esta foi escolhida devido a simplicidade de programação e também por proporcionar uma gama de bibliotecas que auxiliam a implementação dos objetivos propostos neste projeto. A saber, os principais objetivos do software são:

- Proporcionar uma interface gráfica
- Gerar arquivos
- Enviar e receber comandos
- Plotar gráficos em tempo real
- Calcular parâmetros da produção

Devido ao tamanho do código gerado neste trabalho, serão apresentados somente as principais funcionalidades do código e nesta seção serão descritas a linguagem de programação e as principais bibliotecas utilizadas no projeto.

#### 3.3.1 Linguagem Python

Segundo a Python Software Foundation, Python é uma linguagem de programação interpretada, orientada a objetos, de alto nível com semântica dinâmica. Essa linguagem foi introduzida pela primeira vez em 1990 por Guido von Rossum e tem sofrido inúmeras otimizações desde então. Atualmente, Python conta com uma extensa biblioteca desenvolvida e divulgada por seus usuários. Neste projeto optou-se pela utilização das bibliotecas Tkinter, Matplotlib e PySerial, as quais são detalhadas nos itens posteriores.

### 3.3.1.1 Tkinter

O Tkinter é a principal biblioteca em Python voltada para aplicações GUI (*Graphical User Interface*). Tkinter é o script e aplicação gráfica desenvolvido por John Ousterhout, inicialmente na Universidade da Califórnia em Berkeley e posteriormente na Sun Microsystems.

O Tkinter cria uma interface orientada a objetos, ou seja, as referências do *widget* são objetos, e a estruturação do *widget* é feita utilizando os métodos e atributos dos objetos. Com isso, apesar de sua interface gráfica ter a aparência de programas antigos, os programas feitos com o Tkinter propiciam um melhor entendimento da programação da interface (Figura 3.10.).

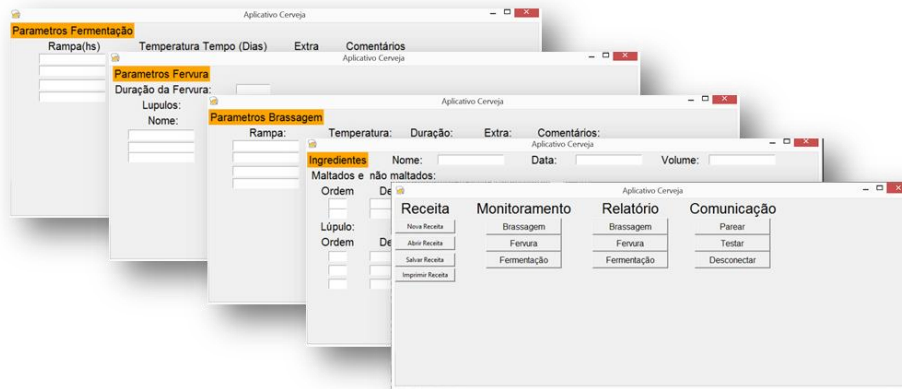


Figura 3.10: Interface gráfica construída com o Tkinter

Para este trabalho, essa biblioteca foi utilizada para gerar a interface gráfica em que o usuário insere como *input* os parâmetros da receita da produção de cerveja a serem enviados para o simulador.

### 3.3.1.2 Matplotlib

A biblioteca Matplotlib foi utilizada para monitorar graficamente as rampas de temperaturas em tempo real comparando-as com as teóricas. A Matplotlib serve para a plotagem de gráficos 2D (Figura 3.11.), trazendo para o programa Python inúmeras funções similares às utilizadas no Matlab.

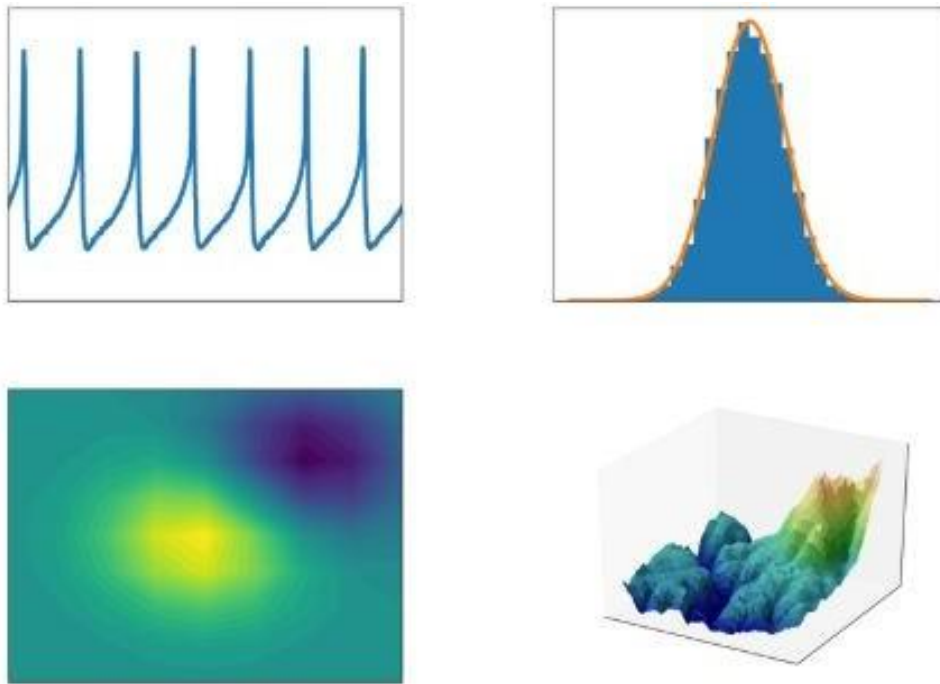


Figura 3.11: Exemplos de gráficos com Matplotlib (Matplotlib)

### 3.3.1.3 PySerial

A PySerial é a biblioteca responsável pela comunicação serial do software computacional. Com ela é possível realizar a comunicação serial para inúmeras aplicações, como comunicação Bluetooth, infra-vermelho e outras.

Para cada um dos modos de operação do simulador foi gerado um relatório da etapa a partir da leitura dos dados enviados na simulação. Devido a quantidade de caracteres disponíveis em cada pacote da mensagem, foi necessária uma função para tratar os pacotes de dados e gerar um relatório de cada uma das etapas. Os relatórios de cada uma das etapas estão ilustrados a seguir nas Figuras 3.12 e 3.13.



■ FervuraJunho - notepad

File	Edit	Format	View	Help
FERVURA				
dddddd	ddd	dddd	dddd	dddd ddd
-SEG-	TOP	-T1-	-T2-	-T3- POT
00001	004	2000	2050	2088 037
00002	004	2360	2410	2448 037
00003	004	2720	2770	2808 037
00004	004	3080	3130	3168 037
00005	004	3440	3490	3528 037
00006	004	3800	3850	3888 037
00007	004	4160	4210	4248 037
00008	004	4520	4570	4608 037
00009	004	4880	4930	4968 037
00010	004	5240	5290	5328 037
00011	004	5600	5650	5688 037
00012	004	5960	6010	6048 037
00013	004	6320	6370	6408 037
00014	004	6680	6730	6768 037
00015	004	7040	7090	7128 037
00016	004	7400	7450	7488 037
00017	004	7760	7810	7848 037
00018	004	8120	8170	8208 037
00019	004	8480	8530	8568 037
00020	004	8840	8890	8928 037
00021	004	9200	9250	9288 037
00022	004	9560	9610	9648 037

Figura 3.12: Relatório de fervura gerado mediante a leitura dos dados

FermentaçãoJunho - Notepad

File Edit Format View Help

FERMENTAÇÃO

ddddd	ddd	ddd	ddd	ddd	ddd	XXX	XXX	XXXX
-MIN-	TOP	-SP-	-T1-	-T2-	-T3-	POT	BPM	EXTRA
00001	010	2000	2400	2440	2480	037	020	12336
00002	010	2000	2310	2350	2390	037	020	12336
00003	010	2000	2220	2260	2300	037	020	12336
00004	010	2000	2130	2170	2210	037	020	12336
00005	010	2000	2040	2080	2120	037	020	12336
00006	010	2000	1950	1990	2030	037	020	12336
00007	011	2000	1950	1990	2030	037	020	12336
00008	011	2000	1950	1990	2030	037	020	12336
00009	011	2000	1950	1990	2030	037	020	12336
00010	011	2000	1950	1990	2030	037	020	12336
00011	011	2000	1950	1990	2030	037	020	12336
00012	011	2000	1950	1990	2030	037	020	12336
00013	012	1600	1860	1900	1940	037	020	12336
00014	012	1600	1770	1810	1850	037	020	12336
00015	012	1600	1680	1720	1760	037	020	12336
00016	012	1600	1590	1630	1670	037	020	12336
00017	012	1600	1500	1540	1580	037	020	12336
00018	013	1600	1500	1540	1580	037	020	12336
00019	013	1600	1500	1540	1580	037	020	12336
00020	013	1600	1500	1540	1580	037	020	12336
00021	014	1200	1410	1450	1490	037	020	12336
00022	014	1200	1320	1360	1400	037	020	12336

Figura 3.13: Relatório de fermentação gerado mediante a leitura dos dados

## Capítulo 4

# Resultados Experimentais

### 4.1 Introdução

Finalizada a construção do simulador e do software computacional, testou-se a utilização do software computacional para uma aplicação de produção de cerveja, de forma a analisar as suas funcionalidades.

Dado isso, foi criada uma receita fictícia que produz 30 litros de cerveja de trigo utilizando os seguintes ingredientes e respectivas proporções:

- Malte de Trigo: 3,5 Kg
- Malte Pilsen: 1,5 Kg
- Malte Vienna: 1,5 Kg
- Lúpulo Saphir: 48g
- Fermento Safbrew Wheat: 2 pacotes

Adicionalmente, os parâmetros utilizados para as etapas estão descritos a seguir:

- Brassagem:
  1. Iniciar em 55 °C e manter por 5 minutos
  2. Subir para 65 °C em 10 minutos e manter por 10 minutos
  3. Subir para 70 °C em 5 minutos e manter por 8 minutos
  4. Subir para 76 °C em 5 minutos e manter por 5 minutos
- Fervura:
  1. Ferver por 70 minutos



- Fermentação:
  1. Manter em 18 °C por 7 dias
  2. Descer para 16 °C em 12 horas e manter por 4 dias
  3. Descer para 12 °C em 10 horas e manter por 10 dias
  4. Descer para 10 °C em 6 horas e manter por 5 dias

## 4.2 Resultados

Inicialmente, foram inicializados o programa computacional e os Arduínos, que foram conectados ao computador e ficam à espera de comandos do computador. Já o programa foi aberto por meio do *prompt* de comando do computador, dando acesso assim, a sua página inicial, conforme pode ser visto na Figura 4.1.

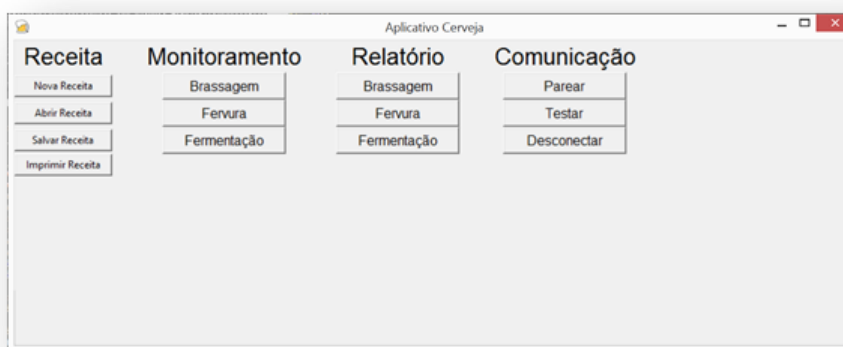


Figura 4.1: Tela inicial do programa

Para este caso, o usuário irá criar uma nova receita para colocar os parâmetros citados anteriormente. Portanto, clicando em "Nova Receita", a página do programa segue para a página de receita, conforme Figura 4.2.

Ordem	Descrição	Kg	EBC
1	Trigo	3.5	4
2	Pilsen	1.5	4

Ordem	Descrição	Kg	EBC
3	Vienna	1.5	8
4	Carahell	0.4	25

Ordem	Descrição	g	Ácido Alfa (%)
1	Saphir	48	3.9
0	0	0	0
0	0	0	0

Ordem	Descrição	Pct
1	Safbrew Wheat	4

Figura 4.2: Tela de ingredientes da receita

Na tela de receita, os parâmetros analisados são os ingredientes e suas quantidades, o volume da receita e a coloração dos maltes e amargos dos lúpulos. Então, os campos do programa são preenchidos e, ao clicar em salvar, é gerado um arquivo .txt com os dados da receita e valores referentes a coloração e ao amargor da receita escolhido, Figura 4.3. Por padrão, o arquivo é salvo na pasta onde o programa está localizado com o nome de "ReceitaData".

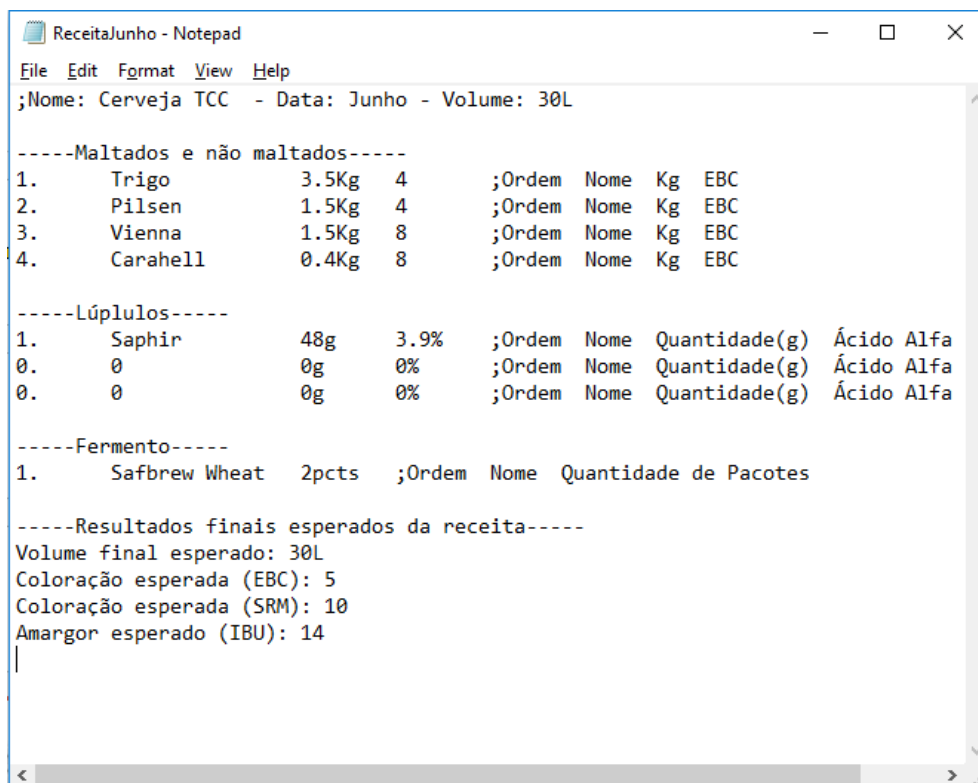


Figura 4.3: Arquivo ReceitaJunho.txt gerado pela tela de ingredientes

A seguir, na página de Brassagem (Figura 4.4), os parâmetros da receita são preenchidos, como as temperaturas, o tempo estimado do processo e o tempo para se chegar na temperatura escolhida. Após inserir os dados e ao clicar em salvar é gerado um arquivo .txt no formato padrão de leitura do controlador de temperatura (Figura 4.5).



Figura 4.4: Tela de parâmetros de brassagem

```

BrasJunho - Notepad
File Edit Format View Help
;Nome: Cerveja TCC - Data: Junho - Volume: 30L
;
;
;Rampa      Temperatura  Tempo  Extra
0           55          5      0      ;Colocar maltes
10          65          10     0      ;
5           70          8      0      ;
5           76          5      0      ;Fim da Brassagem
99          99          99     9      ;      FIM

```

Figura 4.5: Arquivo BrasJunho.txt gerado pela tela de Brassagem

Para a fase de fervura (Figura 4.6), o processo é similar à etapa de brassagem. Porém, os campos a serem preenchidos são a duração da fervura, o momento em que cada lúpulo deve ser colocado na cerveja e a quantidade. Ao final, é gerado um arquivo chamado "FervData.txt" no formato padrão a ser lido pelo simulador (Figura 4.7).

Aplicativo Cerveja

**Parâmetros Fervura**

Duração da Fervura:

Lupulos:

Nome:	Min. de entrada:	Quantidade:	Comentários:
Saphir	10	31	
Saphir	55	12	
Saphir	64	5	

Figura 4.6: Tela de parâmetros de fervura

```

FervJunho - Notepad
File Edit Format View Help
;Nome: Cerveja TCC - Data: Junho - Volume: 30L
;
;
;Tempo      Lúpulo  Min. de Entrada  Quantidade
70          ;Tempo total de fervura
60          Saphir      ;dt=10      31g -
15          Saphir      ;dt=55     12g -
5           Saphir      ;dt=0       5g -
99          99          ;      FIM

```

Figura 4.7: Arquivo FervJunho.txt gerado pelos parâmetros de fervura

Por fim, os parâmetros de fermentação (Figura 4.8) são preenchidos com a rampa (unidade em horas), a temperatura desejada para cada fase e a duração delas. Além disso existe um campo extra, que foi inserido caso seja necessário adicionar algum outro parâmetro em projetos futuros. Assim como as outras duas etapas, ao clicar salvar, é gerado um arquivo de texto no formato de leitura do simulador, conforme Figura 4.9.

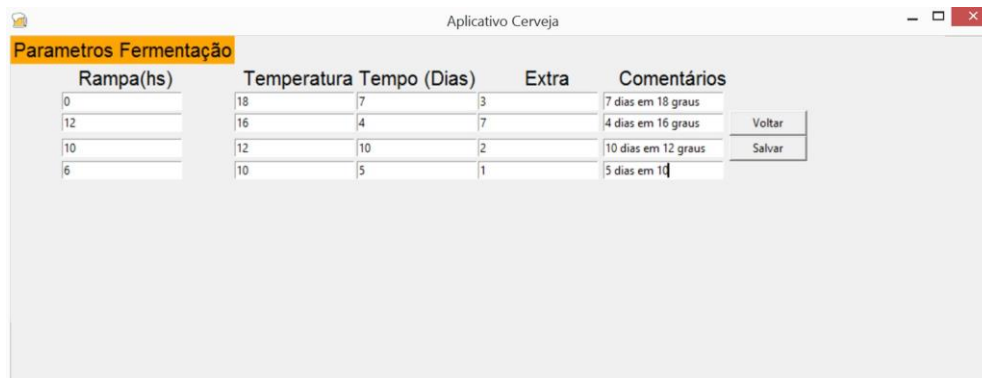


Figura 4.8: Tela de parâmetros de fermentação

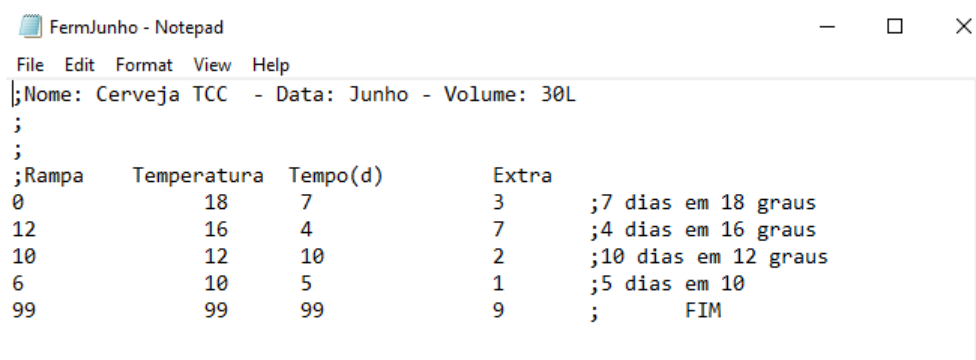


Figura 4.9: Arquivo FermJunho.txt gerado pelos parâmetros de fermentação

Com os parâmetros da receita todos ajustados, os arquivos de textos são salvos no cartão de memória para poderem ser simulados. O simulador fica em modo de espera, aguardando por algum comando do computador.

Portanto, para iniciar a produção, é necessário clicar no botão da etapa a ser realizada, em monitoramento, conforme Figura 4.1. Para esse caso foi simulada a etapa de Brassagem, dado que as outras etapas são análogas no ponto de vista da programação do software.

Assim, ao clicar em brassagem, uma página de gráfico foi aberta com o teórico plotado, para visualizar o experimental foi necessário clicar em "Iniciar", conforme Figura 4.10, para que fosse iniciada a simulação de produção de cerveja. Uma imagem do simulador em operação pode ser vista na Figura 3.5.

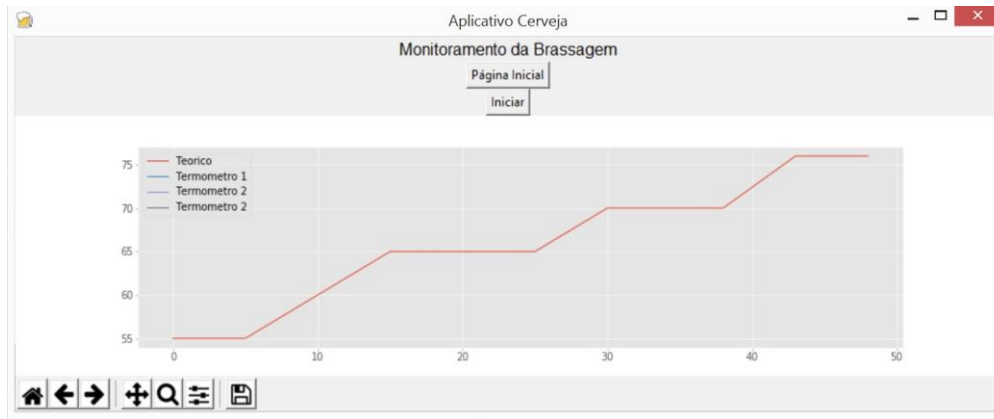


Figura 4.10: Tela de monitoramento de Brassagem

Com a produção em andamento, dois produtos foram sendo gerados pelo software. O primeiro foi um arquivo txt com os dados experimentais da brassagem, enquanto o segundo foi o gráfico de monitoramento atualizado em tempo real, como pode ser visto na Figura 4.11.

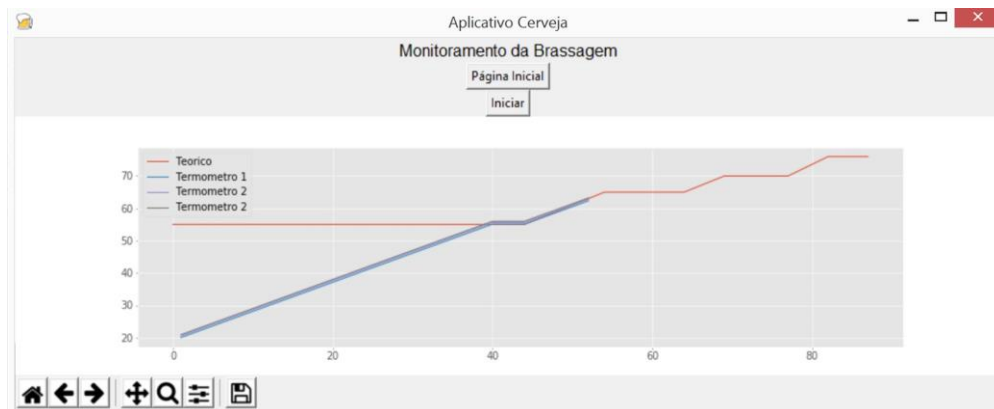


Figura 4.11: Tela de monitoramento de Brassagem durante o experimento

Um ponto interessante a se notar é que, em um primeiro momento, os gráficos das Figuras 4.10 e 4.11 parecem diferentes. Isso ocorre devido ao ajuste de eixo automático à medida que o programa vai coletando os dados da produção. Ao final, foi gerado um arquivo txt com o nome de "BrassagemData".

```

BrassagemJunho - Notepad
File Edit Format View Help
BRASSAGEM
dddddd ddd dddd dddd dddd dddd XXXX XXXX XXXX ddd
-SEG- TOP -SP- -T1- -T2- -T3- -KP- -KI- -KD- POT
00001 000 5500 2000 2050 2088 3039 D431 2BCB 037
00002 000 5500 2090 2140 2178 3039 D431 2BCB 037
00003 000 5500 2180 2230 2268 3039 D431 2BCB 037
00004 000 5500 2270 2320 2358 3039 D431 2BCB 037
00005 000 5500 2360 2410 2448 3039 D431 2BCB 037
00006 000 5500 2450 2500 2538 3039 D431 2BCB 037
00007 000 5500 2540 2590 2628 3039 D431 2BCB 037
00008 000 5500 2630 2680 2718 3039 D431 2BCB 037
00009 000 5500 2720 2770 2808 3039 D431 2BCB 037
00010 000 5500 2810 2860 2898 3039 D431 2BCB 037
00011 000 5500 2900 2950 2988 3039 D431 2BCB 037
00012 000 5500 2990 3040 3078 3039 D431 2BCB 037
00013 000 5500 3080 3130 3168 3039 D431 2BCB 037
00014 000 5500 3170 3220 3258 3039 D431 2BCB 037
00015 000 5500 3260 3310 3348 3039 D431 2BCB 037
00016 000 5500 3350 3400 3438 3039 D431 2BCB 037
00017 000 5500 3440 3490 3528 3039 D431 2BCB 037
00018 000 5500 3530 3580 3618 3039 D431 2BCB 037
00019 000 5500 3620 3670 3708 3039 D431 2BCB 037
00020 000 5500 3710 3760 3798 3039 D431 2BCB 037
00021 000 5500 3800 3850 3888 3039 D431 2BCB 037
00022 000 5500 3890 3940 3978 3039 D431 2BCB 037

```

Figura 4.12: Arquivo BrassagemJunho.txt gerado ao final da brassagem

De forma análoga são gerados os gráficos e os arquivos de produção para as etapas de Fervura e Fermentação.

# Capítulo 5

## Conclusões

Este projeto foi idealizado devido a dois principais fatores: o crescente número de produtores de cerveja artesanal e a possibilidade de otimizar equipamentos de controle de temperatura, os quais, em grande parte, ainda não possuem ambiente de controle virtual.

Esse projeto teve como foco programar um software em Python que auxilia no monitoramento da produção de cervejas artesanais, oferecendo ao cervejeiro mais conforto e comodidade, simplificando assim, o processo de produção. Assim, foi programado um aplicativo *desktop* que lê os dados da receita a ser realizada e gera os arquivos a serem utilizados na produção. Diminuindo assim possíveis erros humanos que possam ocorrer no processo e possibilitando a repetição de procedimentos de produção.

Tendo em vista estes objetivos iniciais, o sistema de automatização da produção de cerveja artesanal desenvolvido apresentou-se satisfatório, proporcionando um ambiente de monitoramento da produção, facilitando a visualização de eventuais falhas durante o processo e aumentando a segurança de quem produz essa bebida, uma vez que com o software, grande parte do controle da produção pode ser feito a distância.

Para facilitar o funcionamento do programa, foi construído um simulador utilizando dois Arduínos Mega 2560 para que fosse possível testar o projeto para todas as etapas de produção. Ainda, com os testes, é possível perceber que o programa funciona da forma desejada. Porém, ainda existem futuros aprimoramentos que podem ser implantados, como a substituição do arduino TxRx por um módulo bluetooth e a substituição do campo extra da etapa de fermentação por densidade, no momento em que for construído densímetros eletrônicos.

Outro ponto a ser notado é que a possibilidade de se controlar a produção à distância otimiza, principalmente, o processo de fermentação, que é mais demorado, e com a possível construção futura de densímetros eletrônicos, seria possível diminuir a quantidade de cerveja desperdiçada sem deixar de analisar as variações de densidade da mesma durante o processo.

Alguns outros pontos de aprimoramento do programa poderiam ser investigados como:

- Criação de interface compartilhada entre PC e Celular
- Implementação de envio de dados de produção para o controlador de temperatura, retirando assim, a única etapa ainda manual do processo

- Integração com outros tipos de controladores de temperatura

No geral, o software concebido neste projeto realiza grande parte das funcionalidades propostas originalmente e está preparado para receber dados futuros que possam vir a ser adicionados. Ademais, outra possibilidade não explorada é que, com alguns ajustes, o software pode ser adaptado ao controle de outros processos de produção e de outros tipos de equipamentos.



# REFERÊNCIAS BIBLIOGRÁFICAS

- [1] DA ROSA, M. M. *Sistema eletrônico de aquecimento para produção de cerveja artesanal*. Trabalho de conclusão de curso - Universidade de Brasília, 2015.
- [2] CAMPOS, R. M. *Projeto e automatização de um sistema HERMS artesanal*. Trabalho de conclusão de curso – Centro Federal de Educação Tecnológica de Minas Gerais, 2017.
- [3] PALMER, J. A. *How to Brew: Everything You Need to Know to Brew Beer Right the First Time*. Brewers Publications; 3a Edição, 2006
- [4] MATOS, R. A. G. *Como Fazer Cerveja em Casa*. Beer king, 2013
- [5] Arduino – Introdução. Disponível em: < <https://www.arduino.cc/en/Guide/Introduction> >. Acesso em 7 jun. 2019.
- [6] Arduino MEGA 2560 . Disponível em: < <https://www.embarcados.com.br/arduino-mega-2560/> >. Acesso em 7 jun. 2019.
- [7] Winbond Eletronics Corporation, *W25Q128FV Datasheet*, Winbond Eletronics Corporation, 2013
- [8] What is Python? Executive Summary. Disponível em: < <https://www.python.org/doc/essays/blurb/> >. Acesso em 16 jun. 2019.
- [9] GRAYSON, J. E. *Python and Tkinter Programming*. Manning Publications; First edition, 2000.
- [10] Matplotlib. Disponível em: < <https://matplotlib.org/> >. Acesso em 16 jun. 2019.
- [11] PySerial. Disponível em: < <https://wiki.python.org/moin/PySerial> >. Acesso em 16 jun. 2019.
- [12] Escola Superior de Cerveja e Malte, *O que o lúpulo faz na cerveja mesmo?*. Disponível em: < <http://cervejaemalte.com.br/blog/o-que-o-lupulo-faz-na-cerveja/> >. Acesso em 23 jun. 2019.
- [13] Afinal, o que o lúpulo faz realmente na cerveja? Disponível em: < <https://www.institutodacerveja.com.br/blog/n145/dicas/afinal-o-que-o-lupulo-faz-realmente-na-cerveja> >. Acesso em 23 jun. 2019.
- [14] Capital federal tem ao menos 2 mil produtores de cervejas. Disponível em: < <https://www.correiobraziliense.com.br/app/noticia/especiais/made-in-brasilia/2017/04/21/noticia-especial-madeinbrasilia,590019/capital-federal-tem-ao-menos-2-mil-produtores-de-cervejas.shtml> >. Acesso em: 30 jun. 2019.
- [15] Projeto HERMS. Disponível em: < <http://cervejaartesanalonline.blogspot.com/2011/06/projeto-herms.html> >. Acesso em 30 jun. 2019.
- [16] LAPOLLI, C. *Mercado da cerveja 2018*. ABRACERVA, abril 2019.

- [17] DE VARGAS GIORGI, V., “*Cultos em cerveja*”: *discursos sobre a cerveja artesanal no Brasil*. Sociedade e Cultura, 2015. Disponível em: <<http://www.redalyc.org/articulo.oa?id=70344885010>> . Acesso em 30 jun. 2019.
- [18] KINDESLEY, D. *Beer – visual reference guides series*. Londres e Nova York, 2010.
- [19] SILVA, H. A.; LEITE, M. A.; DE PAULA, A. R. V. *Cerveja e sociedade*. Universidade Federal de Juiz de Fora, 2016.
- [20] MEGA, J. F.; NEVES, E.; DE ANDRADE, C. J. *A produção de cerveja no Brasil*. Revista Citino, V. 1, 2011, P.34-42.
- [21] SANTOS, S. P. *Os primórdios de cerveja no Brasil*, 2.ed. Cotia: Ateliê Editorial, 2004.
- [22] ABRACERVA *Mercado cervejeiro – Panorama 2018/2019*. Abr. 2019.
- [23] MARCUSSO, E. F.; MUELLER, V. C. *Anuário de cerveja no Brasil 2018: Crescimento e Inovação*. Ministério da Agricultura, 2019.
- [24] CRUZ, I.; PINHEIRO, L. J.; AMORIM, S. M.; KUGLIN, V. B. *Produção de cerveja*. Universidade Federal de Santa Catarina, 2008. P.36.
- [25] GHESTI, G. F. *Fundamentos de Produção de Cerveja*. Notas de aula, Universidade de Brasília, 2013.
- [26] Cigarra Elétrica. Disponível em: < <http://casaolec.com.br/cigarra-eletrica/> >. Acesso em 30 jun. 2019.
- [27] Garrafa x Barril: conheça os principais tipos de envase de cerveja. Disponível em: < <http://mestrecervejeiro9.globo.com/faca-sua-cerveja/garrafa-barril.html> >. Acesso em 30 jun. 2019.

# **ANEXOS**

# I. DIAGRAMAS ESQUEMÁTICOS

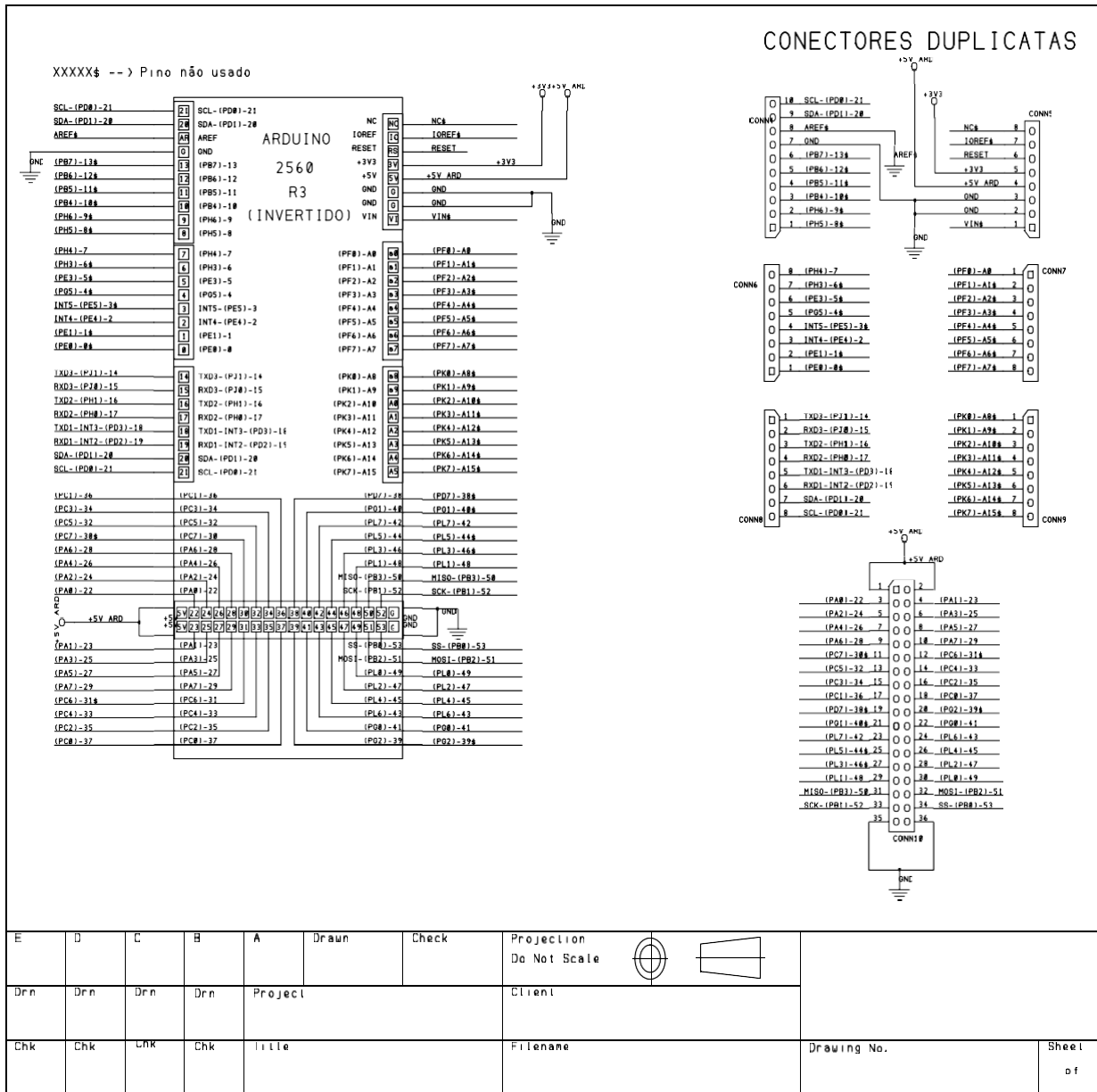
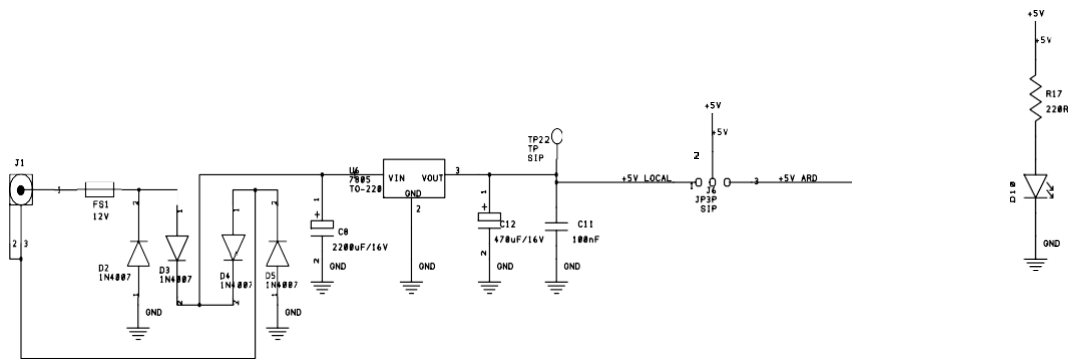
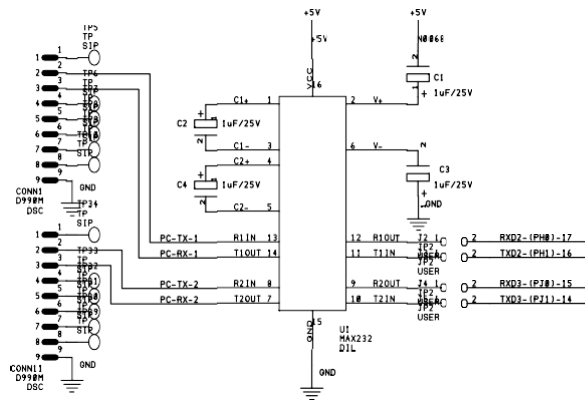


Figura I.1: Esquemático Controlador

PC-RX PINO 2  
 PC-TX PINO 3

Uso das seriais  
 RX0/TX0 --> Arduino  
 RX1/TX1 --> INT2 e INT3  
 RX2/TX2 --> Bluetooth  
 RX3/TX3 --> Zigbee



E	D	C	B	A	Drawn	Check	Projection De Not Scale		
Drn	Drn	Drn	Drn	Project			Client		
Chk	Chk	Chk	Chk	File			Filename	Drawing No.	Sheet df

Figura I.2: Esquemático Fonte

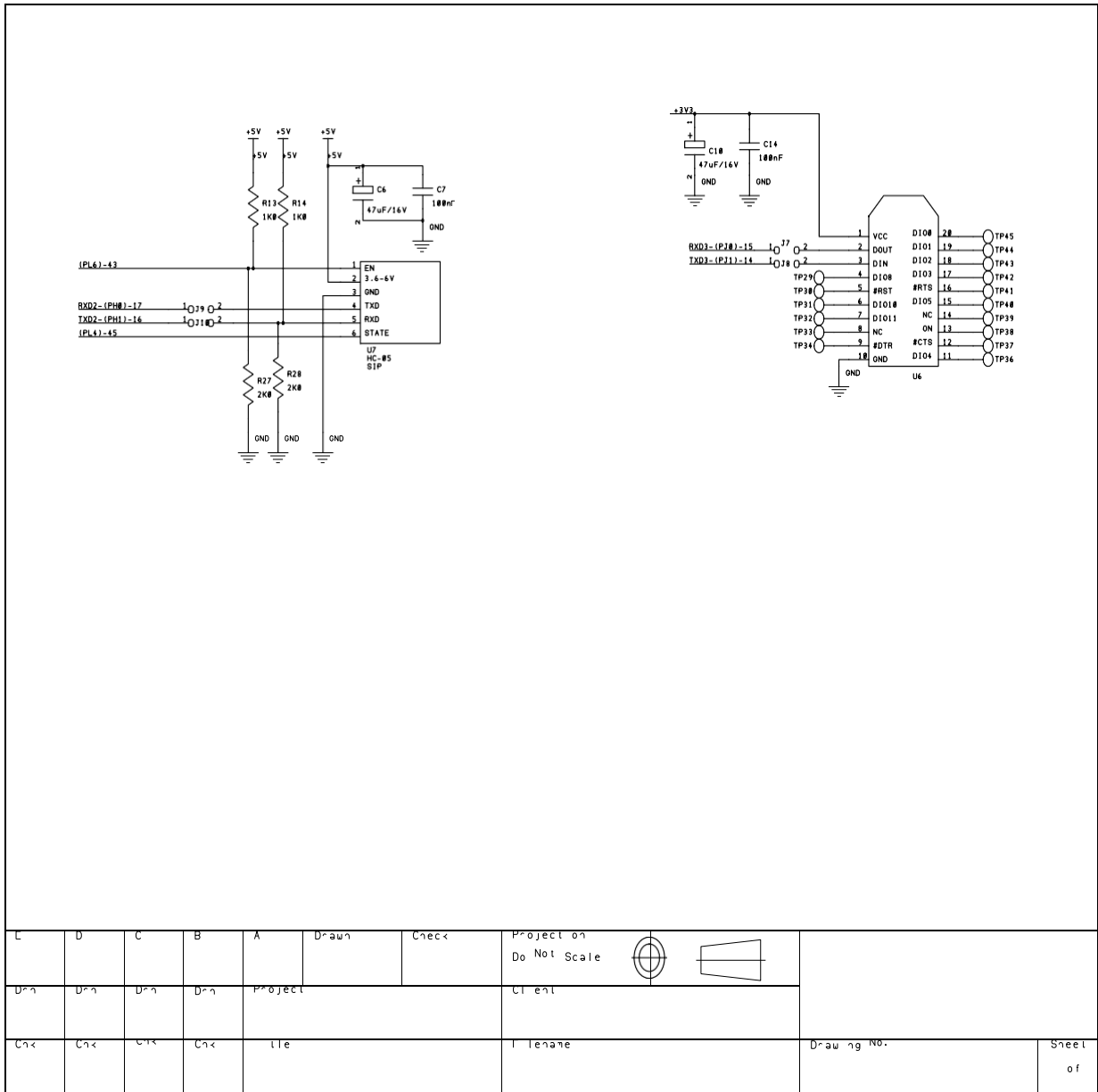


Figura I.3: Esquemático Projeto

