



# **PROJETO DE AUTOMAÇÃO DE UMA COZINHA CERVEJEIRA CASEIRA UTILIZANDO A PLATAFORMA NODEMCU E CONCEITOS DE IOT**

**Trabalho de Conclusão de Curso**

**Engenharia da Computação**

**Rafael Lacerda Miranda**  
**Orientador: Prof. Dr. Sérgio Campello Oliveira**



UNIVERSIDADE  
DE PERNAMBUCO

**Universidade de Pernambuco  
Escola Politécnica de Pernambuco  
Graduação em Engenharia de Computação**

**RAFAEL LACERDA MIRANDA**

**PROJETO DE AUTOMAÇÃO DE UMA  
COZINHA CERVEJEIRA CASEIRA  
UTILIZANDO A PLATAFORMA  
NODEMCU E CONCEITOS DE IOT**

Monografia apresentada como requisito parcial para obtenção do diploma de Bacharel em Engenharia de Computação pela Escola Politécnica de Pernambuco – Universidade de Pernambuco.

**Recife, dezembro de 2018.**



## MONOGRAFIA DE FINAL DE CURSO

### Avaliação Final (para o presidente da banca)\*

No dia 19 de dezembro de 2018, às 17:00 horas, reuniu-se para deliberar a defesa da monografia de conclusão de curso do discente **RAFAEL LACERDA MIRANDA**, orientado pelo professor **Sérgio Campello Oliveira**, sob título **PROJETO DE AUTOMAÇÃO DE UMA COZINHA CERVEJEIRA CASEIRA UTILIZANDO A PLATAFORMA NODEMCU E CONCEITOS DE IOT**, a banca composta pelos professores:

**José Paulo G. de Oliveira**

**Sérgio Campello Oliveira**

Após a apresentação da monografia e discussão entre os membros da Banca, a mesma foi considerada:

Aprovada       Aprovada com Restrições\*       Reprovada

e foi-lhe atribuída nota: 8,0 ( oito )

\*(Obrigatório o preenchimento do campo abaixo com comentários para o autor)

O discente terá 07 dias para entrega da versão final da monografia a contar da data deste documento.

\_\_\_\_\_  
**JOSE PAULO G. DE OLIVEIRA**

\_\_\_\_\_  
**SÉRGIO CAMPELLO OLIVEIRA**

\* Este documento deverá ser encadernado juntamente com a monografia em versão final.

*A meus pais.*

# Agradecimentos

Agradeço a minha família, meu pai, minha mãe e irmão, que fazem de mim quem eu sou hoje. Também à minha namorada, pela paciência e apoio incondicional durante toda essa caminhada.

Agradeço ao meu orientador, professor Dr. Sérgio Campello, por ter sido a medida exata de apoio que eu precisei, nos momentos corretos, das maneiras mais oportunas, durante e além de toda a orientação.

# Resumo

Uma cozinha cervejeira é o coração pulsante da produção de cerveja, seja numa grande indústria, ou numa escala artesanal. Durante o cozimento dos insumos cervejeiros, que ocorrem na sua utilização, diversas etapas de controle de temperatura e circulação líquido acontecem. Quando executadas sem grande atenção variações gritantes são esperadas no produto final. Aliado a isto, existe na cena local uma crescente cultura cervejeira de produção artesanal. No cenário tecnológico a difusão dos conceitos de internet das coisas já é uma realidade, e dispositivos prontos para o uso em projetos de automação já estão disponíveis e acessíveis como nunca. Este trabalho propõe um sistema automatizado para controle de uma cozinha cervejeira caseira usando tais conceitos. Partiu-se da análise de todo processo de produção de cerveja para identificação dos requisitos necessários para a automação, para apresentação e escolha dos componentes. Ele propõe a utilização de dispositivos simples e arquitetura de algoritmo enxuta para a solução dos problemas. Apesar de nem todas as etapas propostas serem passíveis de automações com componentes disponíveis largamente no mercado, observou-se uma possibilidade de ganho significativo no processo como um todo, tornado sua implementação extremamente viável.

# Abstract

A brewery kitchen is the beating heart of the beer production process, be it in big industry, or in a small home brewery. During the cooking of the beer materials several steps of temperature adjustments and liquid circulation occur. When executed without attention, these steps may end up in huge variations in the final product. Together with that, there is a growing local scene of homebrewing culture. In the technologic area the concept of internet of thing is a reality already, and devices ready for use in automation projects are available and accessible like never. This project assembles a proposition for a automatized system to a brewery kitchen using such concepts. From the analysis of the whole process of beer production to the identification of the needs for automation and choosing of components. It proposes the utilization of simple devices and a lean coding architecture into solving problems. Even though not all steps were objects for automation by using market available components, the possibility for a noticeable gain was noted, making the implementation extremely viable.

# Sumário

<b>Capítulo 1 Introdução</b>	<b>1</b>
<b>Capítulo 2 Produção de Cerveja</b>	<b>5</b>
<b>2.1 Processo</b>	<b>7</b>
2.1.1 Receita	7
2.1.2 Moagem	9
2.1.3 Mostura e filtragem	9
2.1.4 Fervura e Resfriamento	13
2.1.5 Fermentação e Maturação	16
2.1.6 Carbonatação	16
<b>2.2 Tipos de Cozinhas</b>	<b>17</b>
2.2.1 Monobloco	17
2.2.2 Tri bloco	20
2.2.3 Demais Configurações	21
<b>Capítulo 3 IoT</b>	<b>22</b>
<b>2.3 Definição</b>	<b>23</b>
<b>Capítulo 4 NodeMCU</b>	<b>27</b>
<b>2.4 Hardware e Firmware</b>	<b>28</b>
<b>Capítulo 5 Desenvolvimento</b>	<b>31</b>
<b>3.1 Projeto Elétrico</b>	<b>35</b>
3.1.1 Arquitetura	35
3.1.2 Componentes	36
3.1.3 Elaboração do Projeto do Circuito Elétrico	39
3.1.4 Arquitetura do Algoritmo	42
3.1.5 Prova de Conceito	44
<b>Capítulo 6 Conclusão e Trabalhos Futuros</b>	<b>49</b>
<b>4.1 Conclusão</b>	<b>49</b>
<b>4.2 Trabalhos Futuros</b>	<b>50</b>
<b>Bibliografia</b>	<b>51</b>
<b>Apêndice A Trechos de códigos de referência do NodeMCU</b>	<b>54</b>
<b>Apêndice B Código do programa da prova de conceito</b>	<b>55</b>

# Índice de Figuras

<b>Figura 1.</b> Número de cervejarias no Brasil .....	1
<b>Figura 2.</b> Resumo do processo de malteação. ....	5
<b>Figura 3.</b> Algumas dentre as centenas de variedades de lúpulos. ....	6
<b>Figura 4.</b> Processo genérico de fabricação de cerveja .....	7
<b>Figura 5.</b> Mapa dos diversos estilos de cerveja. (The, 2018) .....	8
<b>Figura 6.</b> Diferentes moagens: (a) quase farinha, (b) ideal e (c) quase inteira.....	9
<b>Figura 7.</b> Mosto cervejeiro durante a mostura. ....	10
<b>Figura 8.</b> <i>Chiller</i> de imersão em mosto fervendo. ....	14
<b>Figura 9.</b> <i>Chiller</i> de contra fluxo com indicação de uso das entradas. ....	15
<b>Figura 10.</b> Chiller de contra fluxo de capacidade industrial.....	15
<b>Figura 11.</b> Painela e cesto.....	18
<b>Figura 12.</b> Cesto suspenso para a filtragem .....	18
<b>Figura 13.</b> Cesto de inox, guindaste com guincho e fogão de alta pressão.....	19
<b>Figura 14.</b> Cozinha Tri bloco .....	20
<b>Figura 15.</b> Painela de lavagem com tubo interno. ....	21
<b>Figura 16.</b> Evolução da IoT .....	23
<b>Figura 17.</b> Domínios da IoT (tecnológicos e negócios) .....	24
<b>Figura 18.</b> Funcionalidades do Objeto na Internet das Coisas .....	25
<b>Figura 19.</b> NodeMCU .....	27
<b>Figura 20.</b> Diagrama de bloco do ESP8266.....	29

<b>Figura 21.</b> Cozinha Tribloco com fonte de calor por resistência elétrica.....	32
<b>Figura 22.</b> Diagrama de cozinha Tribloco.....	32
<b>Figura 23.</b> Sensores e Atuadores na Panela de Lavagem.....	33
<b>Figura 24.</b> Sensores e Atuadores na Panela de Mosturação.....	33
<b>Figura 25.</b> Fluxograma de Funcionamento do Sistema.....	34
<b>Figura 26.</b> Fluxo de Controle de Temperatura .....	35
<b>Figura 27.</b> Resistencia Elétrica de Baixa Densidade .....	36
<b>Figura 28.</b> DS18B20 .....	36
<b>Figura 29.</b> Relé de Estado Sólido .....	37
<b>Figura 30.</b> Válvula Solenoide .....	38
<b>Figura 31.</b> SHT10.....	38
<b>Figura 32.</b> Bomba Magnética .....	39
<b>Figura 33.</b> Protótipo .....	41
<b>Figura 34.</b> Arquitetura do Algoritmo .....	43
<b>Figura 35.</b> Diagrama do Circuito Elétrico.....	44
<b>Figura 36.</b> Protótipo do Circuito Elétrico Implementado .....	45
<b>Figura 37.</b> Panela Montada .....	46
<b>Figura 38.</b> Programa em Execução .....	47
<b>Figura 39.</b> Gráfico para 10 litros.....	48
<b>Figura 40.</b> Gráfico para 15 litros.....	48

# Índice de Tabelas

**Tabela 1.** Enzimas presentes nos grãos e suas condições de atividade. ....12

**Tabela 2.** Componentes do ESP8266-12E (ESP8266EX, 2018).....28

# Tabela de Símbolos e Siglas

Acerva - Associação de Cervejeiros Caseiros

BIAB – *Brew in a Bag*

BJCP – *Beer Judge Certification Program*

GPIO – *General Purpose Input/Output* (Saída/Entrada de Propósito Geral)

I2C – *Inter-Integrated Circuit* (Circuito Inter Integrado)

I2S – *Inter-IC Sound* (Som Inter-IC)

IoT – Internet of Things (internet das coisas)

LDR – *Light Dependent Resistor* (Resistor Dependente de Luz)

LED – *Light Emitting Diode* (Diodo Emissor de Luz)

MAPA – Ministério da Agricultura, Pecuária e Abastecimento

PWM – *Pulse Width Modulation* (Modulação de Largura de Pulso)

SSR – *Solid State Relay* (Relé de Estado Sólido)

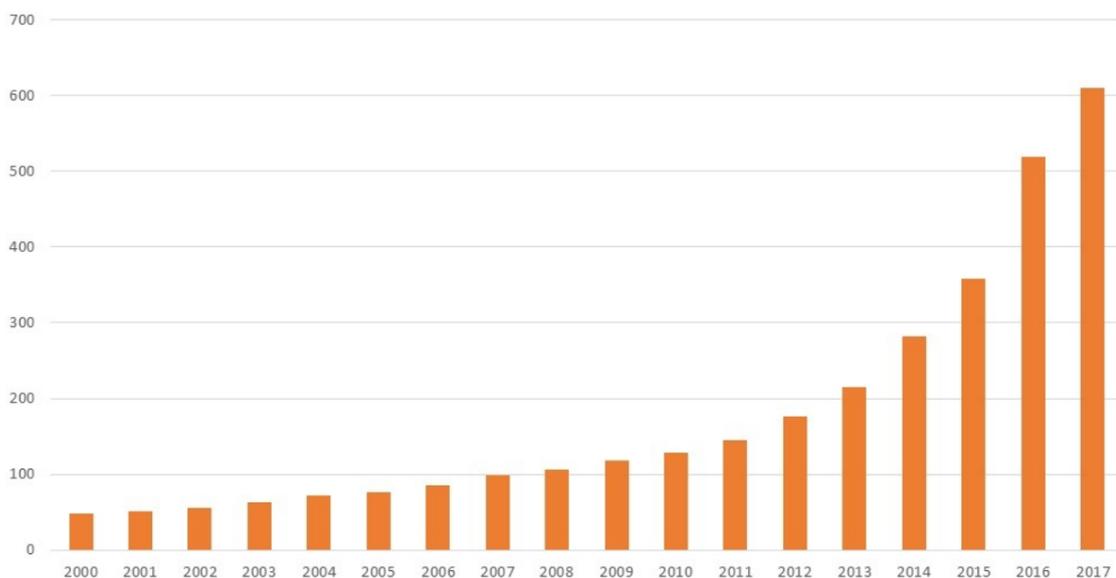
UART – *Universal Asynchronous Receiver/Transmitter* (Receptor/Transmissor Assíncrono Universal)

# Capítulo 1

## Introdução

Devido a seu conjunto de valores culturais, a cerveja é uma das bebidas mais apreciadas no mundo. Sua trajetória se confunde com a da história da humanidade. Há registros dos primeiros campos de cultura de cereal antes até do surgimento da escrita, no ano 9.000 a.C. na Ásia. A roda não havia sido inventada e já se encontravam registros de fabricação de cerveja no Antigo Egito em 3.400 a.C. (MORADO, 2009).

Segundo artigo publicado pelo Ministério da Agricultura, Pecuária e Abastecimento (MAPA), o mercado produtor de cervejas brasileiro vem passando por um aumento acelerado no número de estabelecimentos. No final de 2017 já são mais de 610 estabelecimentos com 7.540 produtos registrados no Ministério da Agricultura (MARCUSO, 2017). O crescimento observado é de mais de 500% desde o início da década, conforme ilustrado na Figura 1.



**Figura 1.** Número de cervejarias no Brasil

A prática de se fabricar cerveja artesanalmente em casa vem se tornando cada vez mais comum, pois a popularidade das cervejas artesanais está crescendo

(SHETH, SISODIA, WOLFE, 2007). O movimento cervejeiro caseiro, de panela ou *homebrewer* (todos termos para descrever quem faz cerveja em casa) vem acompanhando o crescimento do setor. O aumento no número, por todo país, de lojas de insumos específicos à atividade; a maior quantidade de informações e grupos de discussão; o aumento na quantidade de fornecedores de equipamentos para as necessidades do cervejeiro caseiro são alguns indicadores que comprovam esta tendência.

A cerveja típica europeia chegou ao Brasil pelas colônias holandesas que possuíam tradição de produção familiar. Após a industrialização brasileira a cerveja artesanal desaparece dos registros oficiais e só volta a aparecer no início do século XXI (SANTOS, 2004). A indústria cervejeira se impôs como único produtor. A história da cerveja brasileira se moldou a partir dela e as produções artesanais perderam foco.

O renascimento da cerveja artesanal se associa com movimentos do fim do século XX. O movimento *slowfood*, em 1989 na Itália, levou à difusão do alimento como fonte não apenas nutritiva e comercial, mas cultural, ambiental e política. Esse resgate de relações abriu portas para diversas outras vertentes no mundo gastronômico, dentre elas o *slowbeer* (VARGAS, 2015).

No Brasil, o movimento do *slowbeer* foi acolhido pelas associações de cervejeiros caseiros (Acerva) regionais e as cervejarias artesanais. A busca do movimento é pelo resgate do conceito da cerveja artesanal: disseminar a cultura cervejeira, acolhendo pessoas interessadas, estreitando laços de amizade entre os membros e os apoiadores dessa cultura; colocar a cerveja como objeto de devoção e estudo cultural e de análise sensorial. Neste movimento existe o cervejeiro caseiro.

A regulamentação, inspeção e fiscalização de estabelecimentos produtores e de bebidas em território nacional compete ao MAPA (MARCUSO, 2017). Uma série de restrições são colocadas aos produtores visando segurança e controle sanitário para que se consiga a permissão de produção e venda. De modo simples, a única diferença entre um cervejeiro caseiro e um comercial é o registro no MAPA.

O processo de fabricação de cerveja passa por diversas etapas e quando artesanal requer maior atenção, demandando um acompanhamento contínuo principalmente das variações de temperaturas e tempos de mudanças de etapas, tornando o processo bastante exaustivo (NACHEL, 2013). Além disso, seja caseiro ou não, depende largamente de um bom controle de temperatura por todas as etapas: da cozinha à fermentação. Equipamentos comerciais que permitam o controle desejado na cozinha costumam ter o preço muito elevado, impossibilitando ao *homebrewer* saltar de um processo rústico para algo mais refinado. Tal salto aumentaria a qualidade de sua cerveja e diminuiria o trabalho total necessário, uma vez que o tempo médio na cozinha varia entre 5h e 8h (HOW, 2018).

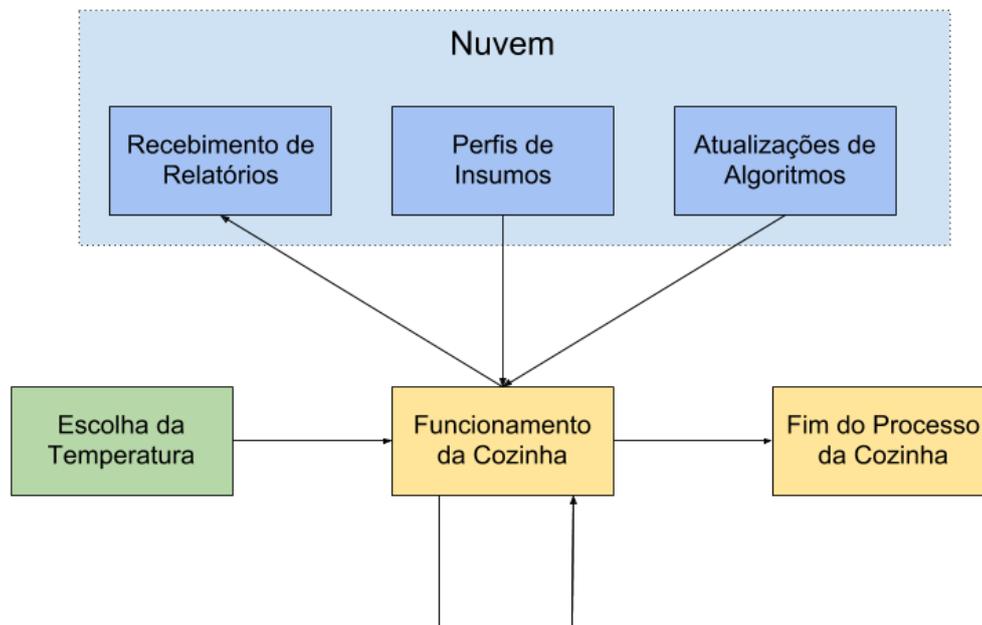
Recentemente, diversas soluções para controle e automação vêm surgindo nos últimos anos. O custo, a facilidade de uso e implementação destas vêm diminuindo fortemente (FACCIONI FILHO, 2016), colocando-as numa posição perfeita para atender as necessidades do cervejeiro de panela. Neste universo de soluções, a NodeMCU é uma plataforma completa para controle, automação e comunicação, com fácil programação e código aberto (PRASAD, 2017), se destacando ainda para a aplicação desejada.

Por sua vez, a Internet das Coisas (IoT) é um conceito relativamente novo, em formação e de definição difícil, pois cada entidade que o tenta desenvolver tende a influenciar sua definição para englobar características que ela julgue mais relevantes (MINERVA, BIRU e ROTONDI, 2015). Mas algumas definições apontadas como interconexão das coisas, conexão das coisas com a internet, inteligência embarcada, sensores e atuadores, capacidade de comunicação interoperável, autoconfiguração e outros são extremamente interessantes à aplicação em uma cozinha cervejeira caseira automatizada.

Este trabalho descreve o desenvolvimento da proposta de um sistema para integrar uma plataforma de IoT na cozinha de uma cervejaria artesanal caseira visando proporcionar ao cervejeiro maior eficiência, qualidade e controle sem precisar arcar com os elevados custos de equipamentos prontos. O sistema permite ao cervejeiro: extração de dados para detecção de melhorias, telemetria, auto ajuste e outras configurações.

Tal sistema possibilitaria a melhoria do processo de produção, uma vez que a temperatura do processo é melhor controlada, bem como, a circulação do líquido, permitindo uma distribuição de calor uniforme. Desta forma, a qualidade do produto final não dependeria da interação do usuário e nem ficaria sujeita a falhas de medidas de variáveis de maneiras diferentes, ou falhas no acionamento da fonte de calor ou circulação.

De maneira resumida a proposta sugere, conforme desenhado no diagrama da Figura 2 um sistema onde o usuário precise definir apenas a sua temperatura de funcionamento e ele seja capaz de se auto gerenciar e autorregular. Conectado a uma nuvem alimentada por um grande número de dispositivos semelhantes a ele, compartilhando relatórios de performance, sem que haja necessidade de nenhuma interação humana até o final do processo. A nuvem pode servir também para compartilhamento de perfis de insumos e atualizações nos algoritmos de controle dentre inúmeras funcionalidades.



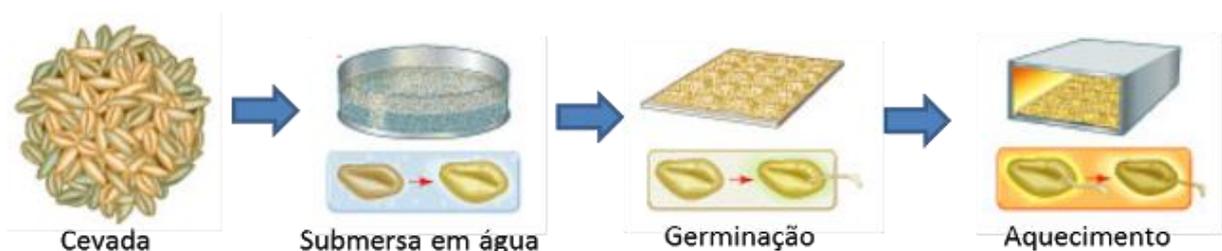
**Figura 2.** Diagrama Resumido do Sistema

## Capítulo 2

# Produção de Cerveja

De maneira simplificada, a cerveja é uma bebida de origem agrícola produzida a partir de cereais maltados e/ou não maltados (milho, trigo e arroz, por exemplo). Trata-se de uma bebida moderadamente alcoólica que se obtém da ação de leveduras selecionadas sobre um mosto resultante da mistura da farinha da moagem dos cereais junto com água, ao qual foi adicionado lúpulo e/ou derivados (Processo, 2018).

O Malte obtém-se a partir da cevada, sujeita a um processo de germinação sob condições controladas, processo exposto na Figura 3. Variando temperatura e humidade é possível conferir propriedades específicas à cerveja. O Malte é responsável pelo fornecimento dos açúcares fermentáveis que contribuem para o condicionamento (carbonatação), conteúdo de álcool e sabores fermentados da cerveja (PAPAZIAN, 2003).



**Figura 3.** Resumo do processo de malteação.

Na parte do cereais não maltados é frequente o uso de milho, arroz ou trigo. A utilização destes cereais tem como finalidade diminuir a percentagem de proteínas existentes no mosto. Estes conferem à cerveja cor mais clara e características específicas consoantes com o cereal escolhido.

Quem confere à cerveja o aroma e amargor específicos é o Lúpulo, que pode ser observado na Figura 4. É uma planta aromática que contribui para a formação de

uma boa espuma e protege a cerveja contra contaminações microbiológicas (PAPAZIAN, 2003).



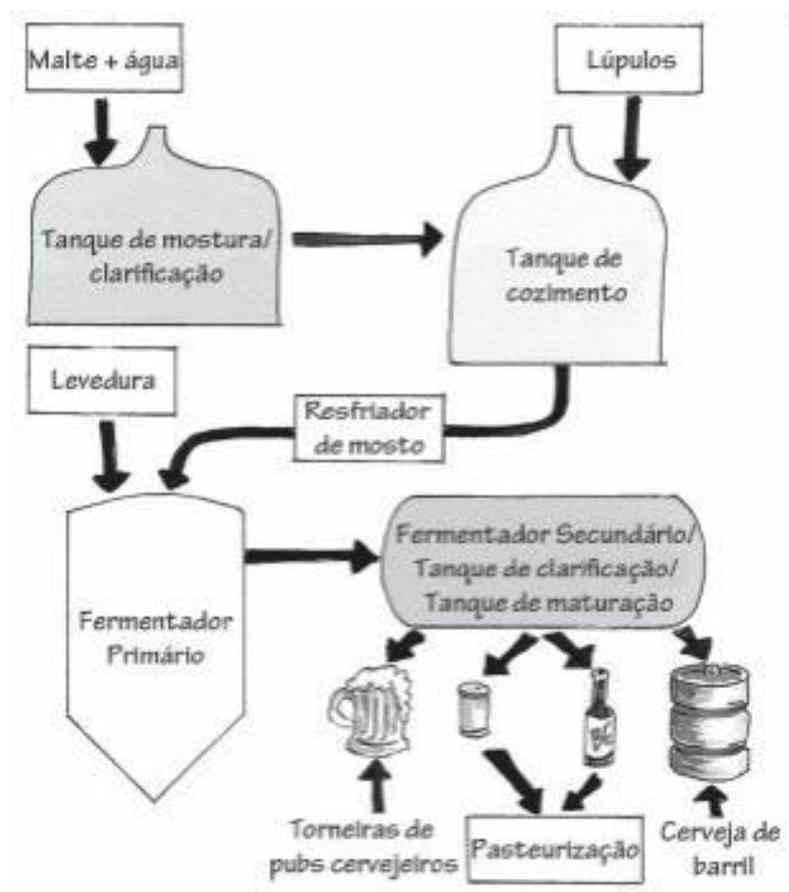
**Figura 4.** Algumas dentre as centenas de variedades de lúpulos.

A Levedura é um fungo que produz etanol, dióxido de carbono e outros componentes a partir dos açúcares, presentes no mosto. Conferem à cerveja características próprias de cada cepa (variedade) de fungo. São encontradas em inúmeros meios como “leveduras selvagens”. As que são utilizadas na fabricação de cerveja foram, em algum momento, coletadas do ambiente, isoladas e cultivadas para serem analisadas e terem seu impacto na cerveja mapeado. Isto garante a padronização e repetibilidade em seu uso.

A todo momento novas cepas são descobertas e não são necessariamente úteis a produção cervejeira. Estilos que se beneficiam de leveduras selvagens são exceção, pois elas têm comportamento imprevisível, podendo introduzir características indesejáveis à cerveja, como aroma desagradável e sabor avinagrado. Leveduras cultivadas são normalmente utilizadas, pois, seu largo uso e conhecimento da cepa evitam estas surpresas (PAPAZIAN, 2003).

## 2.1 Processo

O processo de fabricação de uma cerveja pode acontecer de maneiras diferentes, dependendo da disposição de equipamentos adotada pelo produtor. Apesar de ter essa peculiaridade, passa sempre por um mesmo conjunto mínimo de etapas: elaboração da receita, moagem do malte, mostura, filtragem, fervura, resfriamento, aeração e inoculação do fermento, fermentação, maturação e engarrafamento, que podem ser observados, de maneira geral, na Figura 5 (MÜLLER, 2002).



**Figura 5.** Processo genérico de fabricação de cerveja

De forma generalizada, o processo é dividido entre a parte quente, que ocorre na cozinha cervejeira, compreendendo mostura, filtragem ou clarificação e fervura e a fria, que comprime a fermentação, carbonatação e envase.

### 2.1.1 Receita

A elaboração da cerveja consiste em personalizar e adaptar uma receita base de cerveja artesanal ao paladar desejado, definir o estilo e estabelecer os parâmetros e características a serem alcançados na fabricação.

Dentre os 103 estilos catalogados (BEER, 2015) deve-se escolher um. Observando-se as restrições de perfis de malte, lúpulo e leveduras, bem como os descritores de aroma, sabor e amargor, se elabora uma receita que obedecerá ao que determinam as normas do estilo escolhido. O catálogo de estilos mais difundido é elaborado pelo *Beer Judge Certification Program* (BJCP), um órgão que busca regulamentar as diretrizes de estilo e regras nos concursos cervejeiros por todo o mundo. Embora existam estilos desconhecidos ou apenas não enquadrados, esse catálogo é usado como base por praticamente toda indústria e produtor hobbista. Para se ter uma ideia da imensidade de estilos disponíveis a Figura 6 mostra de maneira geral os estilos e subdivisões.

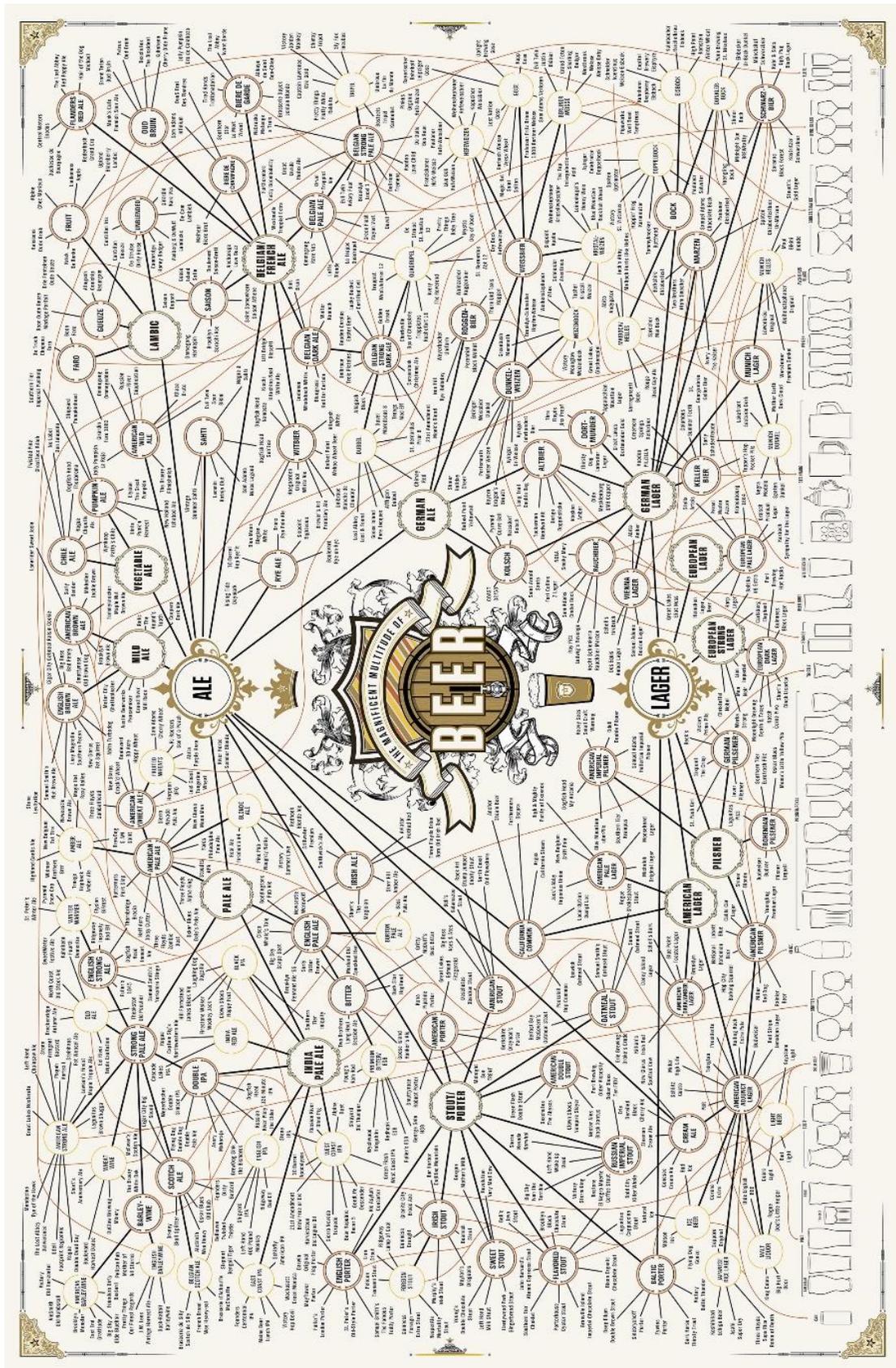


Figura 6. Mapa dos diversos estilos de cerveja (Magnificent, 2018).

As duas principais classificações de tipos de cerveja são *ale* e *lager*, a partir delas que se classificam os demais estilos. As *ales* são fermentadas em temperaturas relativamente quentes e são fermentadas com leveduras de alta fermentação que flutuam na superfície durante o processo. Já as *lagers* são fermentadas a baixas temperaturas com leveduras de baixa fermentação que afundam na cerveja durante o processo (NACHEL, 2013).

### 2.1.2 Moagem

A moagem do malte é a quebra do grão para que o amido contido no seu interior fique exposto. Depois de moído, o malte não pode ser estocado por longo período, por esta razão a moagem deve ser realizada pouco antes do início da mostura (MORADO, 2009). A moagem é um termo relativo aqui; a intenção não é fazer farinha —apenas quebrar a casca do núcleo da cevada para expor os amidos de dentro (NACHEL, 2013), embora se fique com uma quantidade considerável de farinha no fim do processo.

As cascas são importantes, pois auxiliarão na filtragem e clarificação do mosto, tornando-o mais límpido. Se a moagem resultasse apenas em farinha, isso não seria possível. O ideal é controlar o moedor para obter uma quantidade razoável de cascas inteiras, um pouco de farelo e praticamente nenhum grão inteiro, como mostra a Figura 7.



**Figura 7.** Diferentes moagens: (a) quase farinha, (b) ideal e (c) quase inteira.

### 2.1.3 Mostura e filtragem.

O grão moído passa para a mosturação, onde será infundido em água já pré-aquecida em temperatura determinada pela receita, como mostra a Figura 8. Muito frequentemente, o cervejeiro mistura diferentes tipos de maltes ou grãos especiais para conseguir cores e sabores específicos (NACHEL, 2013). Esse líquido fica submetido a variadas temperaturas por períodos determinados resultando em uma solução adocicada, chamada de mosto. Controles rigorosos de tempo e temperatura ajudam a converter com eficácia os amidos de dentro do núcleo do grão de malte em açúcares naturais.



**Figura 8.** Mosto cervejeiro durante a mostura.

As diferentes temperaturas a que é submetido o mosto têm o propósito de estimular diferentes atividades enzimáticas das enzimas presentes no grão, quebrando amidos e aminoácidos que vão influenciar o corpo (densidade), graduação alcoólica, estabilidade da espuma e turvação, por exemplo, da cerveja pronta. O controle de pH também influencia a atividade enzimática, porém atualmente os maltes são bastante modificados em sua produção, dispensando variações complexas de temperatura e controle de pH (contanto que a água usada seja potável) (MEDINA, 2015).

Via de regra, as temperaturas na mosturação são as seguintes (MEDINA, 2015):

**Beta-Glucanos** (35 a 45°C por 15 min a 30 min) – É a temperatura de trabalho da beta-glucanase responsável pela quebra de beta-glucanos:

carboidratos encontrados na camada proteica dos grãos. Apenas necessário quando se está utilizando uma boa proporção (>25%) de cereais que não foram maltados. Esta etapa confere uma viscosidade extra para o mosto, elevando o corpo da cerveja. Além disso, quando se está produzindo uma cerveja do tipo Weiss. Nessa temperatura acontece a formação do ácido ferúlico que proporcionará os aromas de cravo, típico dela.

**Sacarificação** (60 e 65°C ou 67°C e 71°C por 60 min) – Conversão dos amidos em açúcares. Quando mais baixa, a temperatura trabalha a beta-amilase, convertendo o amido em açúcares fermentáveis, resultando em uma cerveja mais leve e seca com maior teor alcoólico. Na faixa mais alta trabalha a alfa-amilase, enzima onde as grandes moléculas de amido são quebradas em açúcares não fermentáveis, resultando numa cerveja mais densa e doce, de teor alcoólico menor. A escolha depende exclusivamente do estilo;

**Mash out** (75 a 79°C por 15 min) – Terminada a conversão do amido em açúcares, esta etapa funciona para inativar as enzimas, evitando que continuem a trabalhar durante as próximas etapa.

Todas as etapas dependem diretamente da exposição dos grãos imersos em água a uma determinada faixa de temperatura e de uma variação de temperatura controlada para a mudança de fases. Deve-se levar em consideração também o tempo necessário para a temperatura se estabilizar por todo o volume na panela. A abordagem mais coerente para este processo é a de manter o líquido sempre em movimento durante as variações de temperaturas e, ao chegar no teto da etapa, desligar a fonte de calor, religando-a apenas se a temperatura baixar à proximidade do piso da mesma. Ainda assim, sem um controle fino da temperatura e da fonte de calor, o cervejeiro fica sujeito a variações no resultado final devido a inconsistências na ação das enzimas.

Conforme a Tabela 1 mostra, existem outras proteínas atuantes em outras faixas de temperatura. Elas ou entraram em desuso, devido a evolução do processo de malteação dos grãos, fazendo com que eles dispensem tais faixas, ou são necessárias apenas em casos bastante isolados.

**Tabela 1.** Enzimas presentes nos grãos e suas condições de atividade.

Enzima	Melhor temperatura	Escala de PH	Função
Fitase	30-52 °C	5.0-5.5	Reduz o pH da mostura.
Debranching (var.)	35-45 °C	5.0-5.8	Solubilização de amidos.
Beta-glucanase	35-45 °C	4.5-5.5	Facilita a quebra do amido. Indicado para utilização de cereais não maltados.
Peptidase	45-55 °C	4.6-5.3	Produz maior quantidade de proteínas solúveis no mosto.
Protease	45-55 °C	4.6-5.3	Quebra proteínas grandes que formam a turbidez na cerveja
Beta Amilase	55-65 °C	5.0-5.5	Produz maltose.
Alfa Amilase	67-72 °C	5.3-5.7	Produz uma variedade de açúcares, incluindo a maltose.

A etapa Fitase era usada para corrigir o pH da água utilizada, mas foi posta em desuso devido a evolução dos maltes. O Debranching, ou gelatinização, só se faz necessário no uso de grãos crus nas maltados, como milho, centeio e trigo, por exemplo. Ela se faz dispensável no uso de flocos destes mesmos produtos, o que é largamente aplicado. Peptidase e Protease auxiliam na quebra de proteínas, provendo mais nutrientes para as leveduras na fermentação e proporcionando uma cerveja ainda mais clara, mas também foram rendidas opcionais com a evolução das matérias primas.

A filtragem é a separação do bagaço do malte e do mosto. Durante a mostura, o mosto é mexido para ser homogeneizado toda vez que a temperatura precisa ser ajustada. No *mash out* ele é deixado descansando e vai gradualmente assentando no fundo do recipiente, formando um filtro natural, por isso a importância das cascas do malte inteiras na moagem. O mosto deve ser recirculado, para filtrar qualquer casca ou sedimento através da cama de grãos. Dependendo do equipamento usado, uma quantidade extra de água é aquecida na mesma temperatura do *mash out* e é adicionada por cima do mosto existente enquanto ele é retirado por baixo do malte

assentado e deslocado para o recipiente da fervura (MARTINS, 1991). Este procedimento varia conforme o equipamento usado, como descrito na Seção 2.2.

O grão, agora chamado de bagaço, não tem mais uso para o cervejeiro após o processo. Rico em celulose, mas com baixo conteúdo proteico, é amplamente destinado à alimentação de ruminantes ou à compostagem. Alguns cozinheiros o usam como fonte de fibras na produção de pães e massas diversas.

#### **2.1.4 Fervura e Resfriamento**

A fervura é responsável pela esterilização do mosto e qualquer adjunto adicionado a ele. Contribui diretamente para o resultado da cerveja, pois é nesta etapa que se definem sua cor e sabor. A fervura deve ser intensa, permitindo grande evasão de vapores, eliminando assim os odores indesejados (MORADO, 2009). É nela que o cervejeiro tenta equilibrar o sabor doce do mosto com os agradáveis efeitos amargos dos lúpulos. Os lúpulos que irão conferir o amargor devem ser adicionados no início, os que irão dar sabor, mais ao fim da fervura, e os lúpulos de aroma realmente no final (NACHEL, 2013).

Uma das características mais importantes do lúpulo na cerveja é o de estender sua durabilidade através de compostos que inibem o crescimento de bactérias causadores da deterioração da cerveja. A liberação e solubilização destes compostos são resultados da isomerização dos lúpulos, que acontece durante a fervura. Este processo também é responsável pela estabilização do amargor gerado pelos lúpulos (PAPAZIAN, 2003).

Certos tipos de proteínas podem ser coagulados durante a fervura do mosto. Estas proteínas coaguláveis são em grande parte responsáveis pela opacidade da cerveja. Após a fervura, há a união de proteínas e restos de lúpulos e adjuntos que se decantam no fundo da panela, o chamado *trub* (MORADO, 2009). Uma técnica difundida entre cervejeiros para aumentar aglutinação e depósito do *trub* é a do *whirlpool*. O mosto é submetido a uma força centrífuga, como uma colher mexendo um copo de suco, a fim de formar um redemoinho, que puxa as proteínas e partículas suspensas para o centro e fundo da panela, onde ficam depositadas.

Seguindo o fim da fervura e depósito do *trub* deve haver um rápido resfriamento do líquido para que ele atinja a temperatura necessária para a fermentação, e evitar a contaminação e a formação de novos odores indesejados. É necessário que o resfriamento seja rápido pois um líquido quente e doce é o meio perfeito para o desenvolvimento de bactérias e leveduras selvagens (presentes no ambiente) (NACHEL, 2013).

O resfriamento costuma ser feito através da imersão de um elemento refrigerante no mosto quente ou da passagem do mosto por uma unidade trocadora de calor. Quando no uso do elemento imerso é costume se usar um tubo de metal para passagem de água ou outra solução, o *chiller* de imersão, proporcionando a troca de calor, resfriando o mosto na panela, como exemplificado na Figura 9. Ele deve ser adicionado ao mosto fervendo à 15 min do fim da fervura, para ser sanitizado pela temperatura e pelo vapor.

Já na segunda situação, o mosto é transferido direto para o fermentador através de uma unidade que fará a troca de calor conforme o líquido passa por ela. Uma configuração simplificada é o *chiller* de contra fluxo, representado na Figura 10, onde um tubo similar ao do *chiller* de imersão é revestido por uma mangueira, por onde a solução ou água fria passará, trocando o calor. Equipamentos mais complexos podem existir se adequando a necessidade da produção como a Figura 11.



**Figura 9.** *Chiller* de imersão em mosto fervendo.



**Figura 10.** Chiller de contra fluxo com indicação de uso das entradas.



**Figura 11.** Chiller de contra fluxo de capacidade industrial.

### 2.1.5 Fermentação e Maturação

Ao ser transferido para o fermentador, o mosto frio precisa ser aerado (oxigenado), o que pode ser feito de diversas formas: desde a introdução de uma queda do mosto para o fermentador, onde ela ocorre naturalmente, até a injeção de oxigênio armazenado em um cilindro. As leveduras consomem esse oxigênio para se multiplicarem antes de iniciarem a conversão dos açúcares.

A fermentação é a transformação dos açúcares em CO<sub>2</sub> e álcool, ou seja, é a transformação do mosto em cerveja. O CO<sub>2</sub> é eliminado durante este processo. Ao final desta etapa, o fermento flocula e decanta no recipiente fermentador (NACHEL, 2013). O processo bioquímico realizado pelas leveduras gera compostos essenciais para a determinação dos sabores da cerveja. Simplesmente adicionar álcool ao mosto não seria o suficiente para gerar o produto como é conhecido.

A maturação é a etapa posterior ao final da fermentação, em que a cerveja permanece armazenada em temperatura mais baixa que anteriormente. É nela onde todos os sabores vindos dos processos anteriores se incorporam e suavizam, atingindo um patamar sensorial estável que constitui a cerveja finalizada.

### 2.1.6 Carbonatação

Finalizada, a cerveja pode ser embarrilada ou engarrafada. Caso o fermentador utilizado seja capaz de aguentar pressão interna, ele pode ser utilizado para regular a saída de CO<sub>2</sub> produzido pelas leveduras e fermentar a cerveja pressurizada, assim ela estará pronta para consumo ou envase. Não sendo capaz disso, ela precisa ser carbonatada, o que pode ser feito de duas formas:

**Priming:** Consiste na adição de açúcar nas próprias garrafas lacradas (ou barris). Esse açúcar é fermentado pelas leveduras remanescentes no próprio recipiente envasado, gerando CO<sub>2</sub>.

**Carbonatação forçada:** Ela é submetida a uma carga controlada de pressão de gás carbônico e é deixada em repouso para que o CO<sub>2</sub> tenha tempo o suficiente para se solubilizar na cerveja.

## 2.2 Tipos de Cozinhas

Diversas composições de equipamentos são utilizadas dentre as cervejarias espalhadas pelo mundo. As diferenças significativas entre elas são: a fonte térmica para aquecer as panelas, que pode ser uma chama a gás, resistência elétrica, caldeira de vapor, dentre outras e o número de panelas usadas no sistema. Preferências pessoais, entendimentos dos processos, restrições regionais e experimentação podem influenciar na escolha. Serão exploradas as duas configurações mais difundidas no mercado nacional e um pouco sobre outras possibilidades.

### 2.2.1 Monobloco

Conhecida também como *single vessel*, constitui-se do uso de apenas uma panela para toda operação. O malte fica dentro de um cesto que fica dentro da panela, dessa forma, quando a mostura acaba ele é suspenso, e a água pode ser recirculada através dele por gravidade, permitindo a filtragem. Em seguida é removido para o início da fervura (BRAUMEISTER, 2018).

É a forma mais barata de iniciar a produção de cerveja como um hobbista. Nesse caso é costume se usar um saco de *voil* como cesto, conforme as Figuras 12 e 13, *setup* chamado de *Brew In a Bag*, ou BIAB. Além disso, ocupa o menor espaço de todos tipos de cozinhas, reduzindo preocupações com espaço para uso e armazenamento.



**Figura 12.** Panela e cesto.



**Figura 13.** Cesto suspenso para a filtragem

Outras configurações desse sistema são permitidas desde que respeitado o uso de apenas uma panela. O cesto pode ser de material mais resistente, para permitir uma quantidade maior de grãos, pode haver um guincho para auxiliar na suspensão e retirada do cesto, a fonte de calor pode ser uma resistência elétrica dentro da panela, um fogão a gás ou qualquer outra comportada pela panela usada. Além dos materiais, a forma como a recirculação do mosto é feita, ou os cestos são estruturados varia muito entre fabricantes de versões mais sofisticadas.

As desvantagens desse sistema aparecem conforme o volume de malte aumenta. Tiragens muito grandes demandam uma forma mais sofisticada para mover os grãos no fim da mostura, como na Figura 14. Muitas vezes faz mais sentido optar por um sistema diferente do que arcar com os custos.



**Figura 14.** Cesto de inox, guindaste com guincho e fogão.

### 2.2.2 Tri bloco

Como o nome sugere, é a cozinha que faz o uso de três painelas para o preparo de cerveja, vide a Figura 15. É a mais utilizada por cervejarias que produzem em grande escala. Uma panela faz a mostura, a segunda serve para aquecer a água usada na lavagem dos grãos, e a terceira é usada para a fervura do mosto.

Tem um custo maior do que a monobloco, porém, permite ganho de tempo quando se precisa cozinhar diversas levas de cerveja em sequência. Assim que uma panela é desocupada já pode-se iniciar o passo da leva seguinte (DRAGON, 2018).



**Figura 15.** Cozinha Tri bloco

Existe uma série de configurações desse sistema, por exemplo, como as painelas são posicionadas, podendo ficar em cascata, para fazer transferências por gravidade, ou no mesmo nível, usando bombas para movimentar os líquidos. Outra forma é variando como o aquecimento do líquido é feito, podendo ter uma fonte de calor na panela de mostura, ou uma bomba recirculando o mosto através de um tubo dentro da panela com a água de lavagem, que detém a fonte de calor.

Todo o controle de transferências de líquidos e acionamentos dos sistemas de aquecimento podem ser feitos manualmente ou delegados a controladores automatizados, como a unidade representada na esquerda da Figura 15. A unidade possui mostradores de temperaturas nas painelas e botões para acionar a bomba para transferir os líquidos. Sendo possível ainda automatizar todo esse

funcionamento, deixando o cervejeiro responsável apenas por colocar os insumos nas panelas e ficar atento a possíveis falhas mecânicas.

### 2.2.3 Demais Configurações

O processo de produção cervejeira acontece conforme descrito no Capítulo 2 e não exige nenhum equipamento excessivamente complexo ou caro, como a opção do BIAB deixa claro. A simplicidade do material necessário permite diversas extrapolações e *layouts* que visem facilitar diferentes aspectos do processo geral, o modelo tri bloco, por exemplo, foca em volume e escalabilidade.

A partir do conhecimento do processo, diversas interpretações e ajustes podem surgir para se adequar a realidade do cervejeiro. Uma panela extra para aquecer mais água e fazer uma lavagem nos grãos suspensos em uma *single vessel*, ou para agilizar o aquecimento da água de mostura numa tri bloco. Uma caldeira a para aquecer a água de forma mais eficiente em sistemas de grande volume que precisem usar gás. Não existem restrições, apenas adequações às particularidades de quem usa as cozinhas.

# Capítulo 3

## IoT

A “internet das coisas” surgiu há pouco tempo como uma ideia de “rede” de comunicações e processamento entre equipamentos. Uma presença ubíqua de dispositivos eletrônicos conectados a Internet e entre si, capazes de interagir com o ambiente, gerar e compartilhar informações (ATZORI; IERA e MORABITO, 2010). A expressão serve para dar ao conceito mais abrangência além de compreensão imediata quanto a sua magnitude, tecnologia e perspectivas de futuro. Ela se encontra fora do âmbito das tecnologias, apenas as utiliza para cumprir um conjunto de funcionalidades, não necessariamente derivando delas (FACCIONI FILHO, 2016).

As funcionalidades abarcadas pela IoT já foram postas por organizações normativas diversas e pelo próprio mercado, e as tecnologias utilizadas continuam em constante desenvolvimento, como demonstra a Figura 16. É grande o número de indústrias da área de equipamentos eletrônicos e computacionais num esforço grande e constante para criar uma linha própria de produtos e dispositivos para IoT na busca de se firmarem como líderes deste nicho (BARNAGHI e SHETH, 2014).

O alto impacto em diversos aspectos do dia a dia e no comportamento de seus usuários é sem dúvida o ponto mais forte de toda a ideia por trás do conceito e o que alavanca o desenvolvimento acelerado da tecnologia. São muitos os desafios tecnológicos encarados, como a capacidade computacional e consumo energético dos dispositivos, a interoperabilidade, segurança, autonomia, privacidade e dentre outros. Ainda assim, igualmente extensos são os investimentos financeiros e esforços em trabalho pelos fabricantes e desenvolvedores na busca para se firmarem como detentores desse padrão tecnológico.

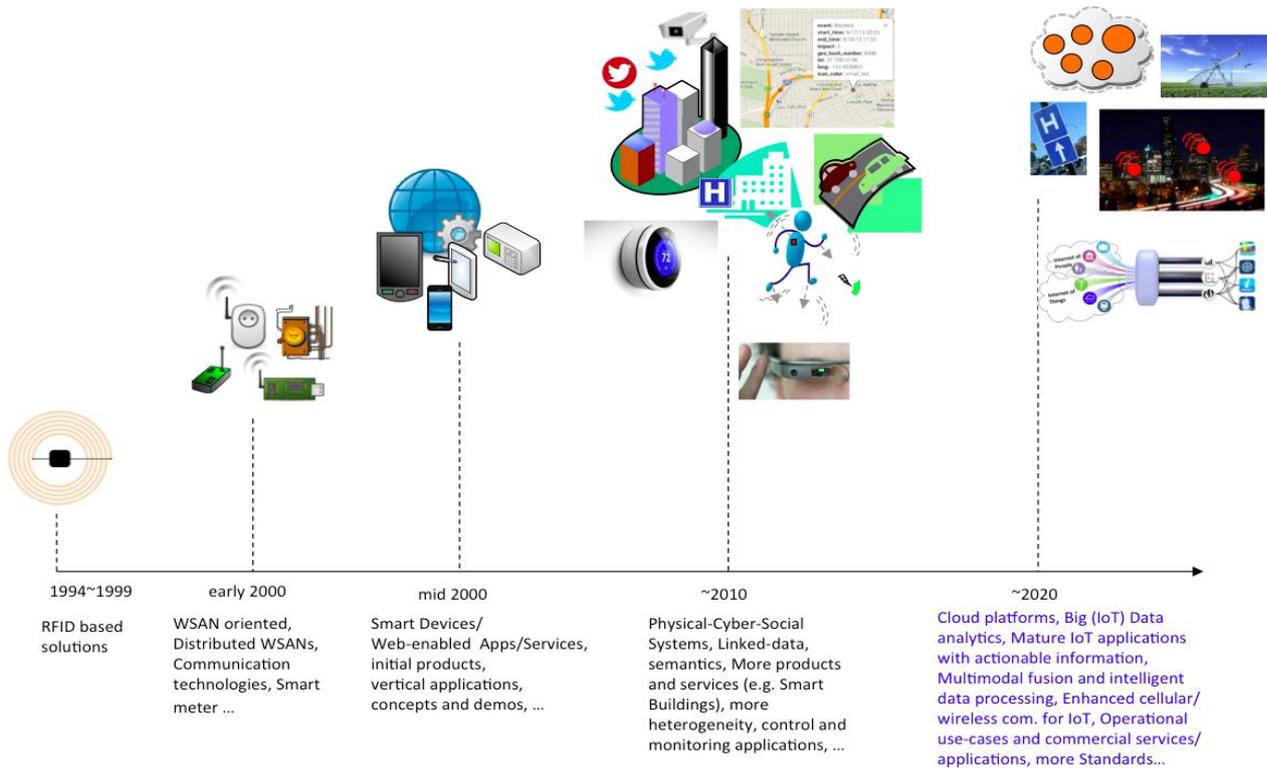


Figura 16. Evolução da IoT (BARNAGHI e SHETH, 2014).

## 2.3 Definição

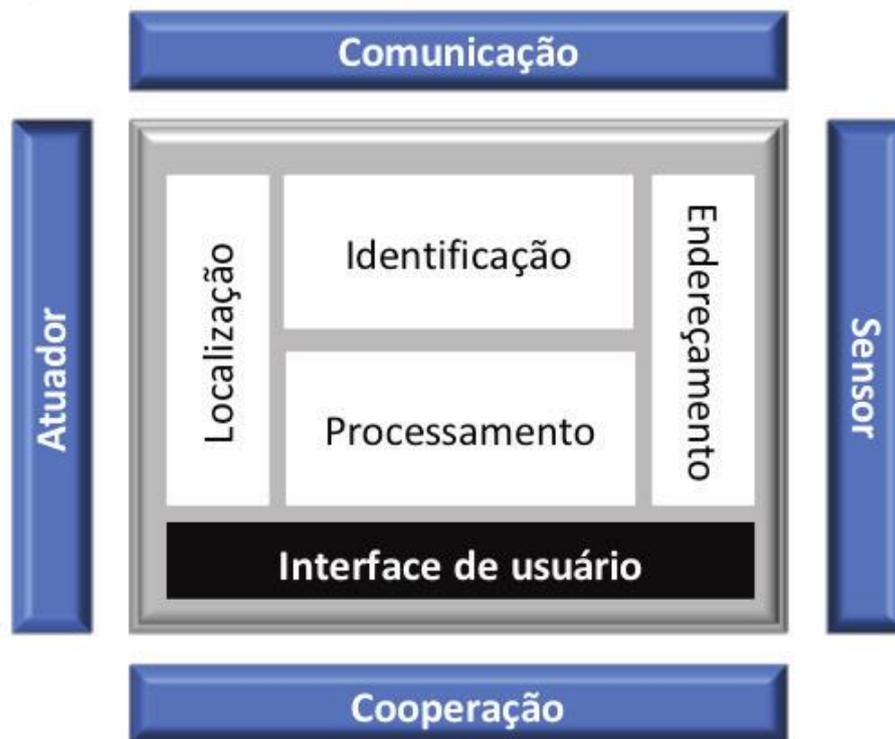
IoT é um domínio que integra diferentes tecnologias e campos sociais e de negócios. Devido a diversidade de elementos que a compõem, ainda não há uma definição consolidada para a Internet das Coisas (MINERVA, BIRU e ROTONDI, 2015). Na Figura 17 pode-se verificar o grande conjunto de áreas cobertas pela IoT.



**Figura 17.** Domínios da IoT (tecnológicos e negócios) (FACCIONI FILHO, 2016).

Em suma, componentes eletrônicos de baixo nível compõem uma camada base espalhada por diversos ambientes. Um conjunto de softwares integra tais componentes (sistemas operacionais, protocolos de comunicação, aplicações, interfaces, bancos de dados e sistemas em nuvem), nada muito diferente de outras redes de sistemas interconectados. Para diferenciar a Internet das Coisas deve-se partir do conjunto de funcionalidades e características do objeto (ou da “coisa”), que tanto pode ser físico quanto virtual.

IoT se define como um sistema complexo que suporta a criação e o *design* de objetos, subsistemas e seus próprios processos internos da IoT. Entretanto, nem todas essas funcionalidades, vistas na Figura 18, precisam estar presentes no objeto simultaneamente, dependendo de seu uso e das aplicações em que ele está inserido (FACCIONI FILHO, 2016).



**Figura 18.** Funcionalidades do Objeto na Internet das Coisas (FACCIONI FILHO, 2016).

Fazem parte das atribuições do próprio objeto:

**Processamento:** que se refere à capacidade de processamento do objeto, capaz de fazê-lo agir e responder às requisições da rede e suas aplicações;

**Identificação:** a capacidade de se identificar o objeto, fazendo-o único em toda a rede;

**Endereçamento:** o objeto pode ser localizado na rede por meio do roteamento;

**Localização:** relacionado ao local físico em que o objeto se encontra (posição no mapa geográfico);

Compõem as funcionalidades que definem como os objetos se relacionam com outros objetos:

**Comunicação:** o objeto é capaz receber/enviar mensagens de/para outros na IoT;

**Cooperação:** o objeto consegue realizar a atividades e aplicações conjuntas e de colaboração com outros objetos;

**Sensoriamento:** capacidade do objeto de captar dados do ambiente ou de outros objetos;

**Atuação:** que é a capacidade do objeto de agir sobre o ambiente.

A **interface de usuário**, é a capacidade do objeto em permitir a interação do usuário, visualizando informações do objeto, realizando configurações e modificando sua condição.

Partindo dessas características e funcionalidades, MINERVA, BIRU e ROTONDI (2015) definem:

“A IoT – *Internet of Things* – compreende uma rede complexa, adaptativa e autoconfigurável, que interconecta *coisas* à Internet por meio de protocolos de comunicação normatizados. As *coisas* interconectadas têm representação física ou virtual no mundo digital, capacidade de atuação/sensoriamento, funcionalidade de programação e identificação única. Tal representação contém informações da identidade, status, localização e informações privadas ou sociais relevantes da *coisa*. A *coisa* oferece serviços, com ou sem intervenção humana, por meio de identificação única, coleta de dados, comunicação e capacidade de atuação. A exploração dos seus serviços se dá pelo uso de interfaces inteligentes e pode ser feita de qualquer lugar, a qualquer tempo e com segurança.”

# Capítulo 4

## NodeMCU

O NodeMCU (NodeMCU, 2018), visto na Figura 19, é uma plataforma *open source* de IoT. Foi desenvolvido para proporcionar uma fácil programação e disponibilizar as funcionalidades de uma conexão. É ideal para soluções *Plug-n-Play*, permitindo fácil prototipação aos desenvolvedores.

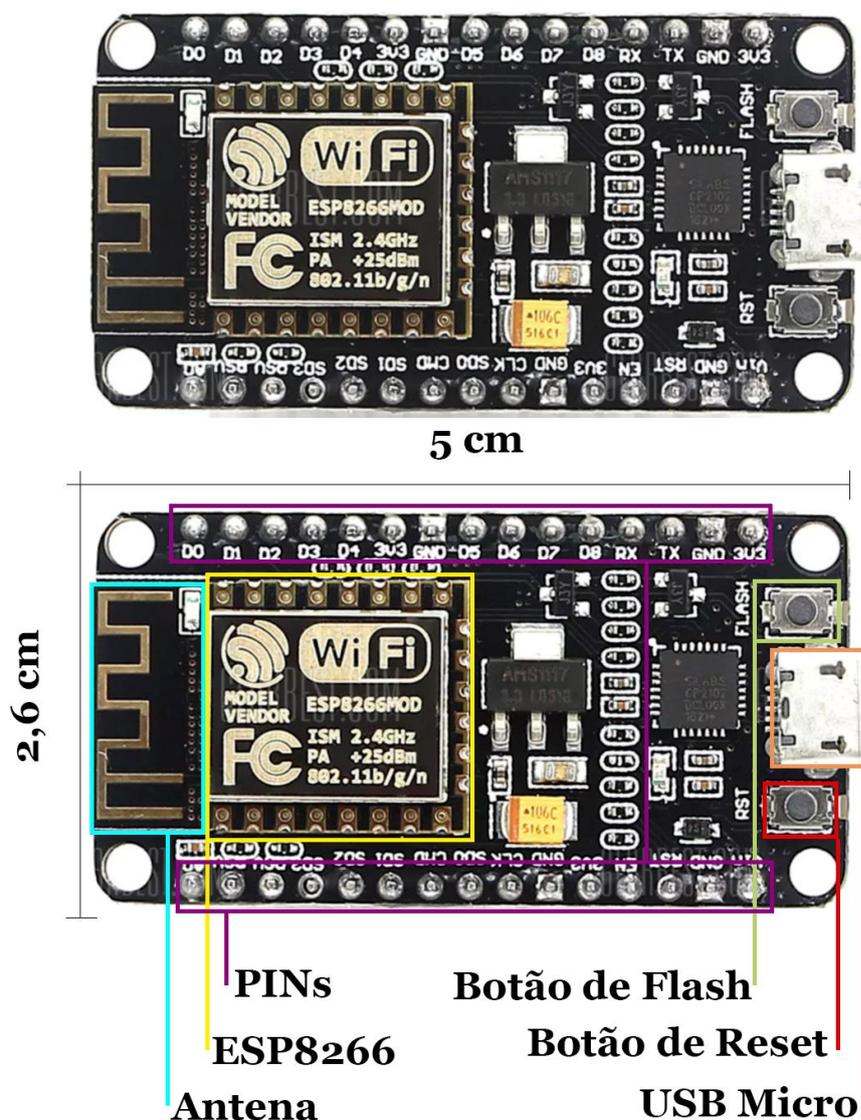


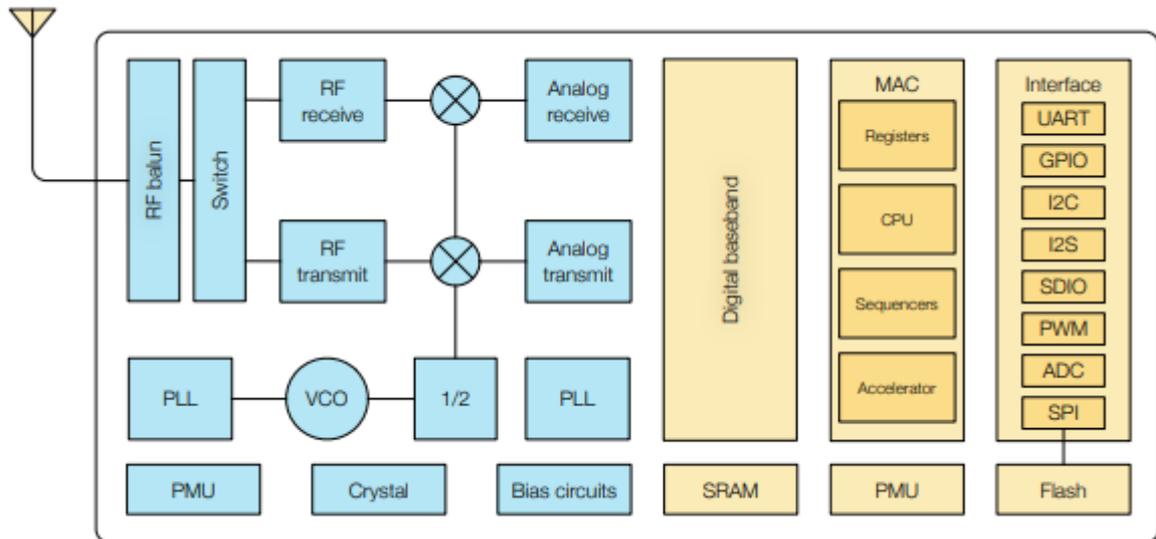
Figura 19. NodeMCU

## 2.4 Hardware e Firmware

O NodeMCU possui o chip ESP8266-12E (NodeMCU, 2018) como microcontrolador, um sistema embarcado projetado pela Espressif Systems (NodeMCU, 2018). Ele se define como uma solução de redes *Wi-Fi* autossuficiente capaz de executar aplicações de maneira independente ou como ponte entre um microcontrolador pré-existente e a rede *Wi-Fi*. Existem vários tipos de modelos de placas disponíveis, o que diferencia uma placa de outra é o número de pinos *GPIO* expostos, a quantidade de memória *flash* disponível, o estilo de pinos conectores e várias outras considerações relacionadas a sua construção (KOLBAN, 2015). O chip ESP8266-12E é composto conforme disposto na Tabela 2 e apresenta o diagrama de blocos demonstrado na Figura 20.

**Tabela 2.** Componentes do ESP8266-12E (ESP8266EX, 2018).

<b>Microcontrolador</b>	Tensilica L106 de 32 bits. (De baixo consumo de energia, <i>clock</i> de 80MHz e opera entre 3V e 3,6V)
<b>Módulo <i>Wi-Fi</i></b>	802.11 b/g/n
<b>Interfaces Periféricas</b>	Portas de entrada/saída digitais ( <i>GPIOs</i> )
	Comunicação serial ( <i>UART</i> )
	Conversor A/D (Analogico/Digital)
	Comunicação serial I2C ( <i>Inter-Integrated Circuit</i> )
	Comunicação serial I2S ( <i>Inter-IC Sound</i> )
	Módulo <i>PWM</i> ( <i>Pulse-Width Modulation</i> )
	Controladora para expansão de memória SPI Flash



**Figura 20.** Diagrama de bloco do ESP8266.

O NodeMCU possui 30 pinos. Memória RAM de 128k bytes. Integra todas as funcionalidades do ESP8266 numa mesma placa adicionando uma porta USB-TTL, *plug & play*, 10 *GPIO* (toda *GPIO* pode ser *PWM*, *I2C* ou *1-wire*), e uma antena no circuito (NodeMCU, 2018). Tem grande semelhança em dimensão e disposição dos *GPIOs* com a placa de desenvolvimento Arduino, com características particulares como WiFi.

Este módulo trabalha com um firmware baseado em Lua para o ESP8266 WifiSOC da Espressif. O *firmware* pode ser customizado em *builds*, que podem ser adquiridos no site do fabricante (Build, 2018) seguindo as funcionalidades do projeto e visando otimizar o espaço de memória. O modelo de programação do NodeMCU é bastante semelhante ao do Node.js, mas em Lua. É assíncrono e orientado a eventos. Diante disso, suas funções têm amplo acesso a *call-backs*. (NodeMCU, 2018), como podem ser observados nos trechos de códigos de referência no Apêndice A.

Dentre as ferramentas disponíveis para o desenvolvimento de aplicações estão:

Um serviço de *builds* na nuvem, onde o desenvolvedor pode escolher os módulos a serem carregados no *firmware* para usar nas aplicações, permitindo versões rápidas e facilmente customizáveis;

Uma *build* do NodeMCU para o Docker, em resumo, uma ferramenta de *containers* que dá controle sobre a forma com que o ambiente no qual o *firmware* é montado é gerenciado sem a necessidade de se montar e configurar todo um novo sistema operacional;

Um guia para usar um sistema operacional, montá-lo e configurá-lo desde o início, e ter total controle sobre o ambiente de montagem do seu *firmware* NodeMCU.

Ao se criar uma nova *build* é possível se escolher apenas os módulos que serão usados na aplicação, ajudando na economia de RAM.

Além dos serviços disponíveis para o uso da plataforma, ainda é possível desenvolver para ela diretamente através da Arduino IDE. Como dito anteriormente, possui grande semelhança em dimensão e disposição dos *GPIOs* com a placa de desenvolvimento Arduino e possui grande suporte da comunidade para desenvolvimento direto por essa ferramenta como alternativa.

# Capítulo 5

## Desenvolvimento

O escopo do desenvolvimento é a automação dos sistemas de aquecimento e transferência de fluidos em uma cozinha cervejeira tribloco, vista na Figura 21. O modelo de cozinha foi escolhido levando em consideração a simplicidade de sua configuração, sua disseminação nos meios cervejeiros caseiros na região, permitindo um possível benefício da comunidade local.

A cozinha da Figura 21 é composta de uma panela mais alta, chamada de panela de lavagem, colocada aí para não precisar do auxílio de uma bomba para transferir o líquido para a panela seguinte, a de mostura, que por sua vez fica ligada a uma bomba e a panela final, a de fervura. Vale observar que esta cozinha especificamente já é toda elétrica e ao menos parcialmente automática, vide a unidade de controle abaixo da panela de lavagem.



**Figura 21.** Cozinha Tribloco com fonte de calor por resistência elétrica.

Todo o processo, desde quando a água é adicionada na panela para a mostura até o início da fervura está coberto pelo projeto. A fervura em si precisa de mais interação humana do que as demais etapas: as adições de lúpulos e adjuntos podem ser feitas de diversas formas e com diversos materiais, tornando o custo de uma automação adequada muito complexo e elevado.

A Figura 22 permite um entendimento melhor da cozinha que será automatizada. A primeira panela a ser automatizada é a de lavagem. Composta apenas por uma fonte de calor, usada para o aquecimento da água e por uma válvula de saída, para a transferência do líquido para a panela seguinte. Nela, fica água usada na lavagem dos grãos presentes na panela de mosturação. Esta, além da fonte de calor, conta com um fundo falso – usado para comportar os grãos enquanto mantém um espaço com líquido entre eles o fundo da panela – e um jogo de bomba e válvulas para fazer a recirculação do líquido. No fim, todo o líquido termina na panela de fervura, composta por uma fonte de calor mais potente e uma válvula, que servirá para deslocar todo o conteúdo para os fermentadores.

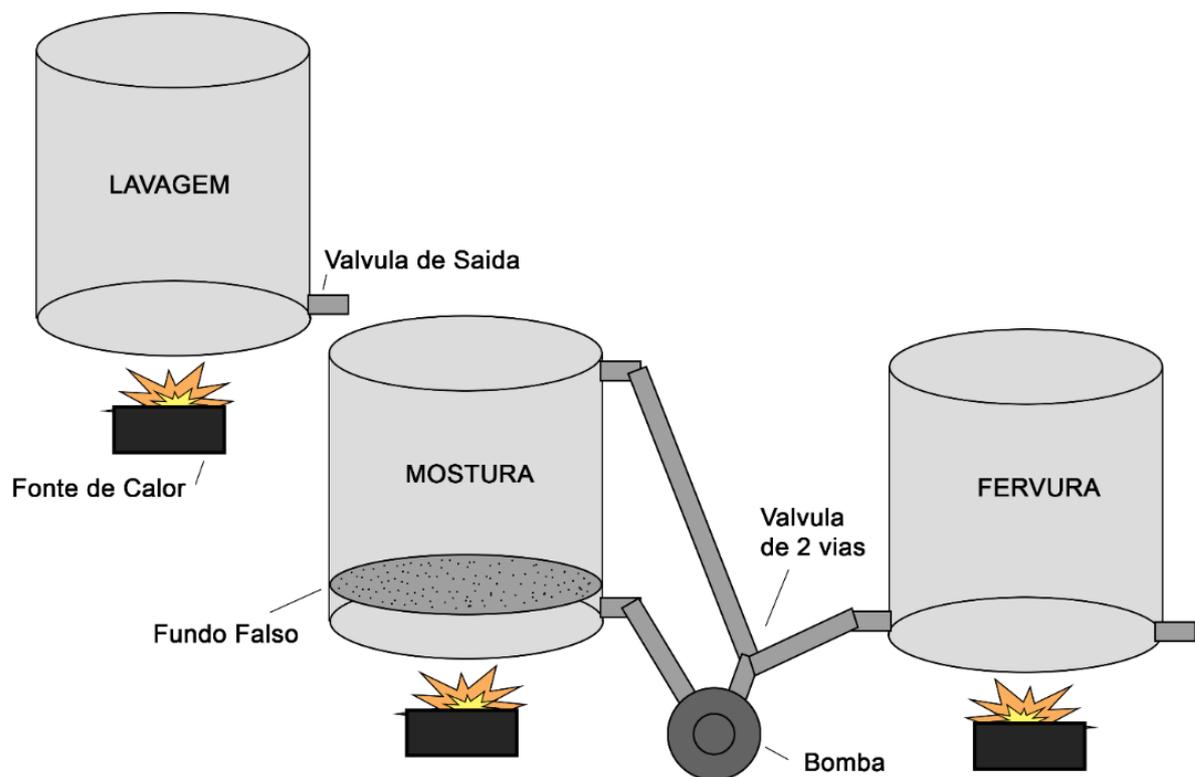
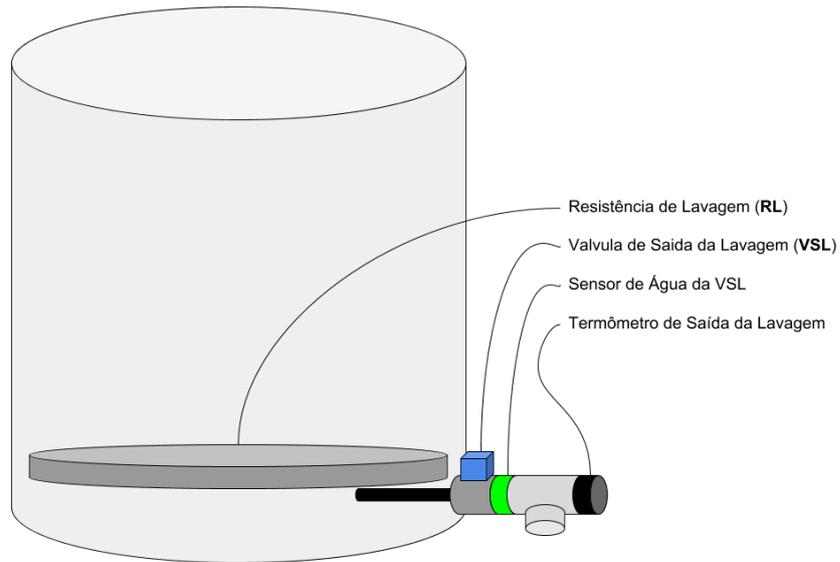
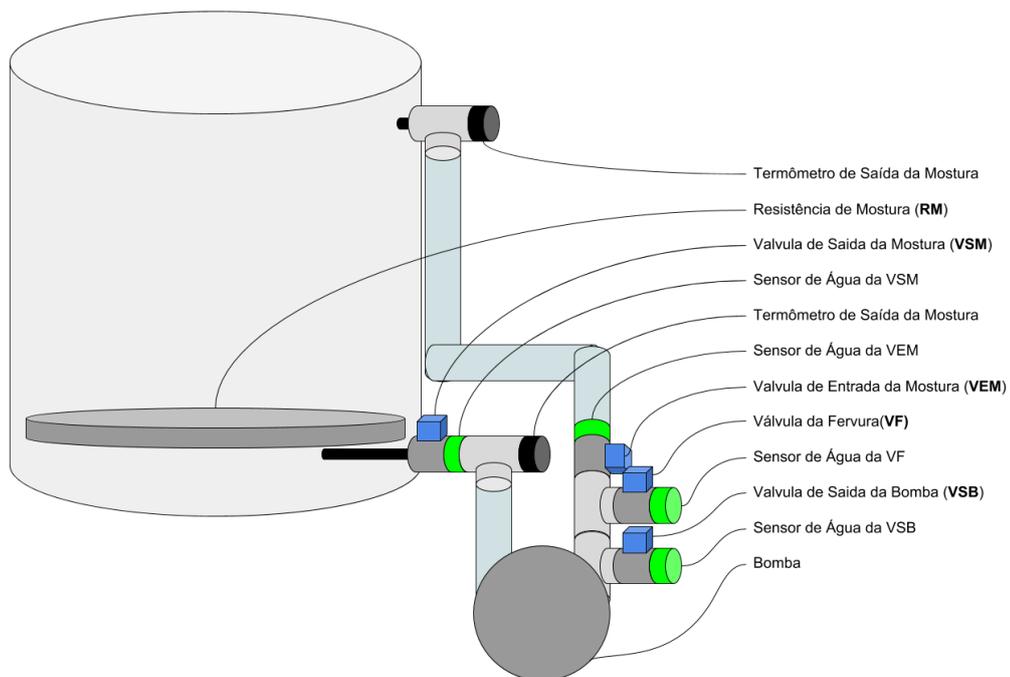


Figura 22. Diagrama de cozinha Tribloco

Os leiautes das figuras 23 e 24 são propostos no posicionamento dos sensores e atuadores nas painelas de lavagem e mosturação. O termômetro está sempre próximo da fonte de calor para garantir a medição da temperatura mais quente, evitando que ela ultrapasse o limite definido, e na segunda panela o segundo termômetro esta aonde se espera ser o ponto mais frio, após a circulação, para se buscar o equilíbrio.



**Figura 23.** Sensores e Atuadores na Painela de Lavagem



**Figura 24.** Sensores e Atuadores na Painela de Mosturação

A partir dos layouts propostos para as painéis, o fluxograma da Figura 25 foi projetado para demonstrar o funcionamento geral do sistema. Ele segue o funcionamento da cozinha verificando se o fluxo de líquido está correto, e controlando os tempos das etapas. Um fluxo alternativo de controle de temperatura faz parte dele e é definido na Figura 26.

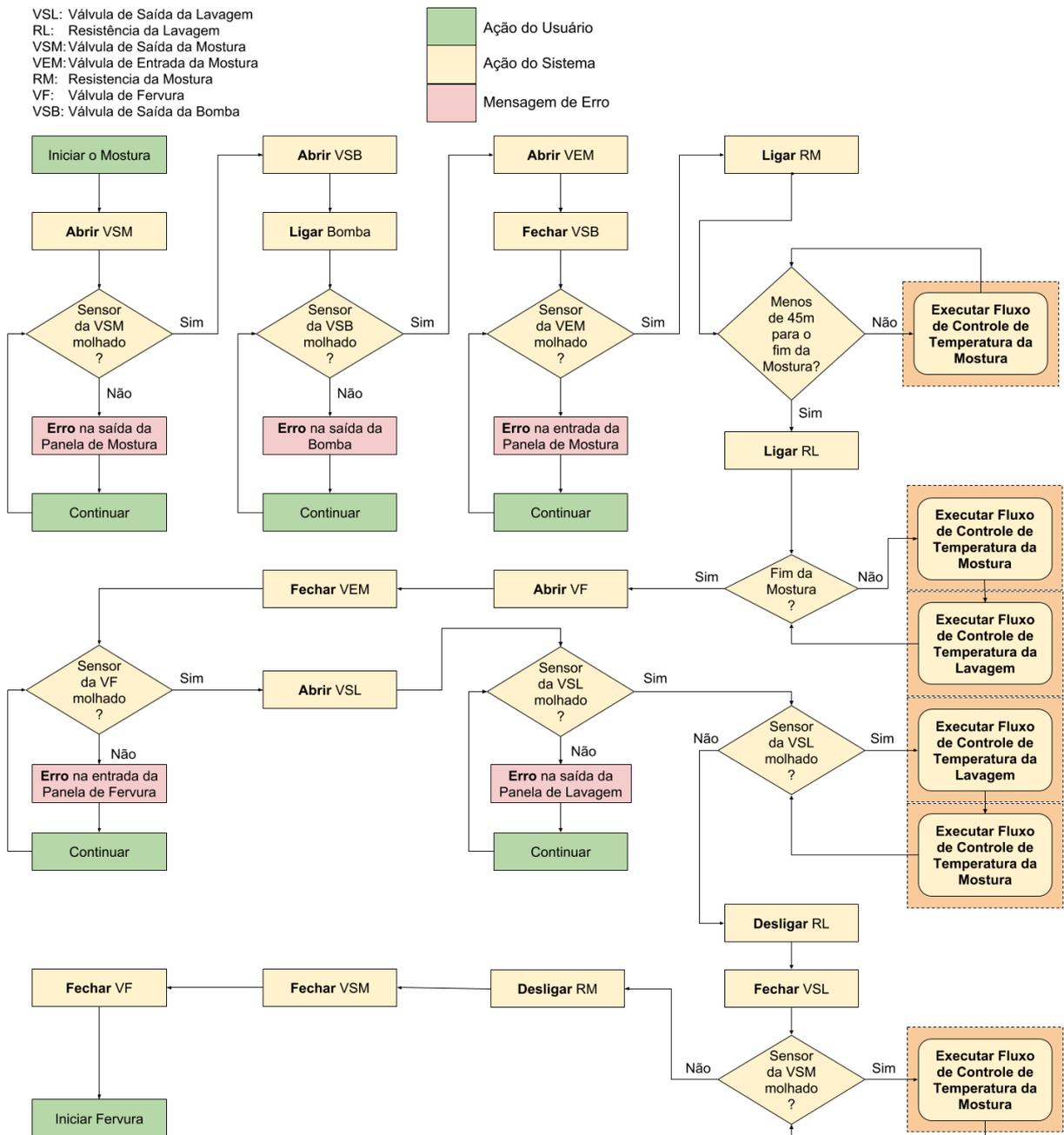


Figura 25. Fluxograma de Funcionamento do Sistema

O fluxo alternativo de controle de temperatura se limita a controlar se a resistência deve aumentar ou diminuir conforme a taxa de aumento de temperatura, tentando mantê-la sempre em 1 ° C/min. Menos do que essa taxa, o aquecimento é ineficiente, lento, mais do que isso, pode caramelizar o mosto na resistência.

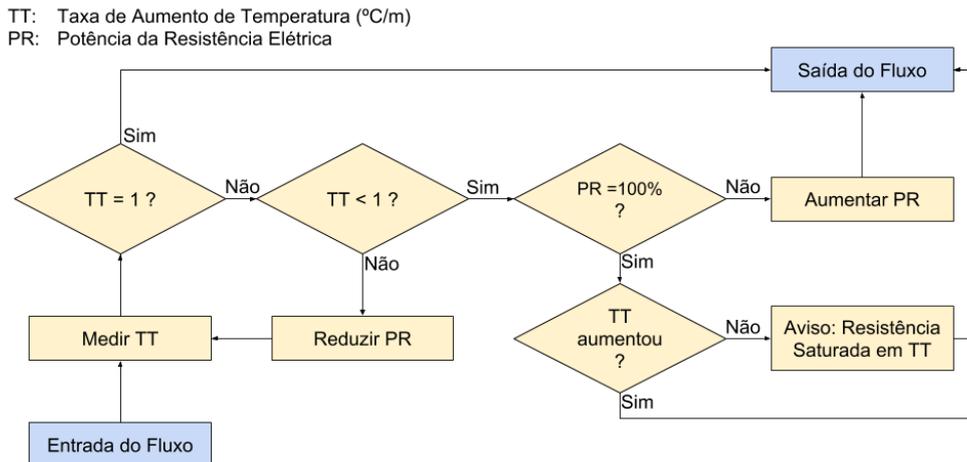


Figura 26. Fluxo de Controle de Temperatura

## 3.1 Projeto Elétrico

### 3.1.1 Arquitetura

O ponto principal deste trabalho é propor um modelo de automação eficiente, de baixo custo, e que permita desenvolvimentos de terceiros, englobando conceitos contidos nas especificações de IoT. A ideia é composta pela utilização de resistências elétricas e sensores de temperaturas para os quais fatores como tamanho, características de operação e padrões de controle impactaram de maneira significativa na escolha.

O funcionamento das resistências elétricas é controlado por um conjunto de relés, por serem de fácil troca e encontrados com facilidade no mercado especializado. O ideal são dois atuadores de calor em cada panela com 50% da potência cada, podendo estar desligados, a 50% ou 100% de potência, obedecendo o fluxograma de controle de temperatura.

O funcionamento da bomba será controlado por um relé, junto com o conjunto de válvulas e sensores de passagem de líquido, para garantir que ela não funcione seca.

### 3.1.2 Componentes

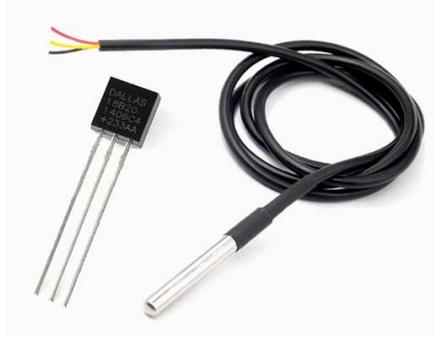
A resistência proposta tem revestimento em inox, por ser muito mais resistente, evitando oxidações e furos ou rompimentos, tendo uma durabilidade muito superior que as alternativas. Além disso deverá ser de baixa densidade, permitindo uma melhor distribuição do calor por sua área, evitando queimas durante elevações de temperatura. Lojas especializadas costumam vender sob encomenda, no tamanho e formato desejado pelo comprador, como na Figura 27, ideal para uma panela.



**Figura 27.** Resistencia Elétrica de Baixa Densidade

Para o termômetro a escolha foi o DS18B20 da Dallas Semiconductor que é um termômetro digital, mostrado na Figura 28. De custo acessível, vendido já instalado num acoplador de aço inox, usa interface 1-wire (similar a I2C), bastante fácil de se trabalhar e muito difundido no mercado.

Possui uma escala de  $-55^{\circ}\text{C}$  a  $+125^{\circ}\text{C}$  e sua precisão é de  $\pm 0,5^{\circ}\text{C}$  na escala de  $10^{\circ}\text{C}$  a  $+85^{\circ}\text{C}$ , com conversão de temperatura direta para digital. A resolução da temperatura do sensor é 12 bits, o que corresponde a incrementos de  $0,0625^{\circ}\text{C}$ , com um tempo de leitura de 750ms. É o componente ideal para o uso na medição de temperatura. E já vem pronto para ser colocado num tubo de passagem de líquido.



**Figura 28.** DS18B20

Os relés usados neste projeto estão expostos a uma condição constante calor e humidade, tal ambiente acelera a degradação de relés comuns. Relés de estados sólido, *solid state relays*, *SSR*, são acopladores óticos - usam um *LED* e um sensor de luz como parte central de seu funcionamento – sendo escolha ideal para essa aplicação.

*SSRs* são circuitos eletrônicos cujas entradas de controle são ligadas a um *LED*, que se acende por um espaço vazio sobre um sensor de luminosidade. O sensor é conectado a um transistor, que se abre ou fecha, provendo a carga do relé com corrente. Quando o transistor se fecha, a corrente pode correr livremente pelo relé, fornecendo corrente entre a fonte e a carga dele. Quando o transistor se abre, quase toda a corrente é bloqueada, causando uma interrupção entre os terminais.

O modelo SSR-25-DA, que pode ser observado na Figura 29, aceita entrada de 3 V a 32 V e alimenta saídas de 24 V a 380 V funcionando adequadamente com todos os componentes do projeto.



**Figura 29.** Relé de Estado Sólido

As válvulas solenoides funcionam através de um mecanismo de diafragma: um acionador elétrico empurra uma borracha que tampa a passagem da válvula, para ser liberada quando ela é levantada. Elas podem ser normalmente fechadas, e abertas quando existe corrente ou normalmente abertas e se fecham com a corrente. É importante que o material usado como diafragma seja silicone de grau alimentício, pois mosto de cerveja passara por ele. Todas essas características podem ser observadas na válvula 2S200-20 da VITON vista na Figura 30. Vale notar que seu acionamento também necessita de um relé, já que a solenoide funciona em 12 V a 220 V, dependendo do modelo adquirido.

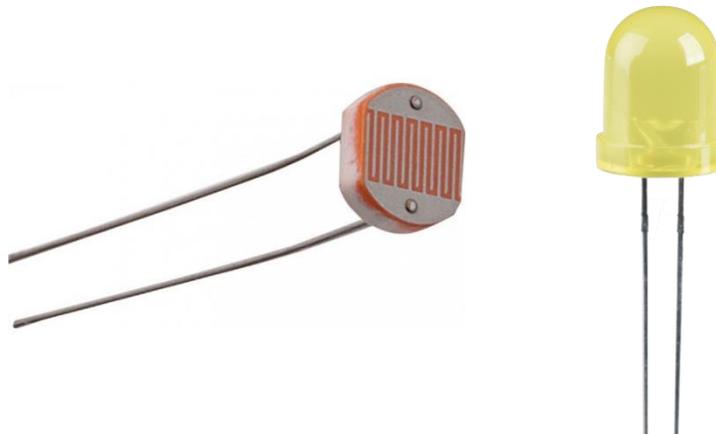


**Figura 30.** Válvula Solenoide

As maiores preocupações quanto a escolha de um sensor de passagem de líquido foram as temperaturas de operação, que poderiam variar de 20 a 80 graus e se eles teriam grau alimentício. A segunda informação não foi encontrada em nenhuma especificação técnica, mas os componentes de solda são conhecidamente tóxicos. Mesmo assim não existem soluções diversas disponíveis, e o que se tem não está disponível em uma forma que seria adaptável para se colocar dentro de um tubo para medição de passagem.

Uma solução interessante seria o uso de pares de LEDs e LDRs, vistos na Figuras 31. O LDR é um dispositivo cuja resistência varia com a quantidade de luz que incide sobre sua superfície. A resistência do LDR varia inversamente à quantidade de luz que incidente sobre ele, isto é, se fluxo luminoso é alto, o LDR oferece uma resistência muito baixa, caso contrário, sua resistência aumenta. Já o

LED, que é um diodo emissor de luz, é um condutor de energia elétrica, que quando energizado, emite luz visível a olho nu.



**Figura 31.** LDR e LED amarelo.

A ideia é ligar cada um de um lado de um tubo transparente, como o da Figura 32. Na ausência de líquido o LED iluminaria abundantemente o LDR, na presença ele seria praticamente escurecido. Uma cor ideal para o LED seria a mais próxima possível ao líquido passando pelo tubo, como amarelo, ou vermelho.



**Figura 32.** Tubo transparente com mosto passando.

Também tendo que atender a mesma restrição de grau alimentício, a bomba precisa ter a cabeça – parte em contato com o líquido – de inox. Bombas capazes de fazerem auto *priming*, de se encherem de líquido estando secas, com cabeças de inox são muito caras. Para atender o escopo deste projeto, seria usada uma bomba eletromagnética, vista na Figura 33, que custa hoje ¼ do preço da outra opção, ambas em inox.

Todavia, seu uso enquanto parte automatizada do projeto fica restrita a existência de uma solução de medição de passagem de líquido pelo tubo imediatamente após ela. Bombas magnéticas não podem funcionar secas ou irão quebrar, primeiro elas precisam encher por gravidade, para só então serem ligadas. A ausência de líquido precisa ser rapidamente comprovada para ela ser desligada e mais tempo ser dado para seu enchimento e uma nova partida ser tentada.



Figura 33. Bomba Magnética

### 3.1.3 Elaboração do Projeto do Circuito Elétrico

Com a escolha dos componentes e suas características e exclusões de módulos não possíveis de automação partiu-se para a prototipação do circuito.

A implementação será feita em um *software* que permitisse a modelagem e do circuito desejado e a realização dos testes sobre o funcionamento dos

componentes escolhidos. Mas, a dificuldade em encontrar algum que trouxesse a maioria dos componentes em sua biblioteca, aliada a grande disseminação do uso desses componentes no meio de projetos de circuitos elétricos apontou para a escolha de uma modelagem manual, apenas para ilustrar o esquema, observado na Figura 34, acima, temos os componentes da panela de lavagem. As resistências e a válvula acionadas pelos relés estão ligadas numa fonte externa de 220 V. Os relés ocupam, cada um, uma posição numa E/S da placa NodeMCU para acionamento, das portas D1 a D3. Todos os termômetros, devido à especificação do protocolo 1-wire, podem ocupar uma mesma E/S, no caso D4, com a necessidade de um resistor de 4,7k ohm ligado da alimentação com o cabo de E/S para que eles possam operar sem a precisar de uma fonte externa.

Logo abaixo estão os componentes da panela de mostura. As montagens se repetem, apenas o número de componentes e pinagem é diferente: aqui temos dois conjuntos de resistências e relés, igual acima, mas nenhuma válvula e dois termômetros. Os termômetros se ligam ao mesmo D4 do da panela de lavagem, como explicado acima, os conectores GND e 3V3 ficam do outro lado placa dessa vez, mas tem a mesma função. Os relés usam D5 e D6.

A bomba foi ligada após um relé na porta D8, a segunda válvula de saída no D0 e o LED está ocupando a D7 acompanhado de um resistor de 200 ohm. O LDR tem o funcionamento analógico, então precisa ser conectado na A0, a única conexão analógica do NodeMCU. Estão previstos mais 3 conjuntos de válvulas, LEDs e LDRs na panela de mostura, que ocupariam outras 9 E/S, porém os conectores se esgotaram. A troca por válvulas manuais e o disparo de alarmes via software solicitando o usuário que faça o acionamento das válvulas restantes servem de alternativa nesse caso.

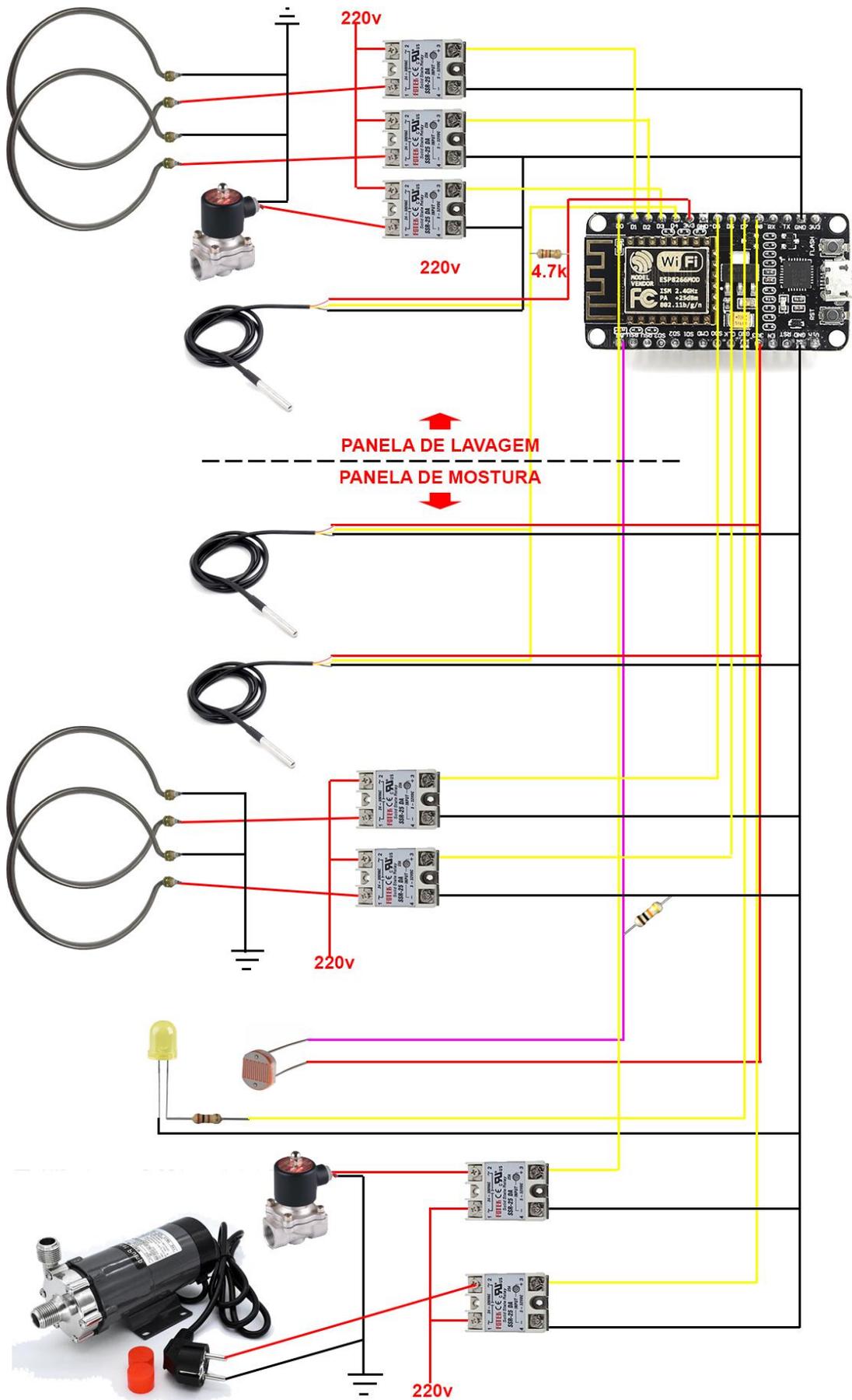


Figura 34. Protótipo do Circuito Elétrico

### 3.1.4 Arquitetura do Algoritmo

Com o circuito esquematizado, parte-se o código que vai comandar todo o seu funcionamento. A ideia, como mostra a Figura 35, é implementar blocos de instruções, que serão chamados através de eventos, disparados pelo bloco de instruções principal, que comanda os componentes das painelas.

No bloco de controle das painelas estariam todas as instruções principais, tratadas nos fluxos acima. O início é dado com a escolha da temperatura, através de uma chamada para a interface remota, solicitando que o usuário faça a configuração. A função de controle de temperatura executa a verificação das temperaturas nos termômetros e do tempo de cozinha e controla as resistências, quando necessário ela aciona a função de controle de válvulas, que cuidam delas e da bomba, observando o fluxo. Estão presentes também, diversos eventos que disparam funções dos outros blocos do sistema.

O segundo bloco seria o do dispositivo remoto, que pode ser qualquer dispositivo capaz de se conectar via WiFi com a NodeMCU para acessar as funções de visualização e controle do sistema. Além da escolha de temperatura, aqui também são mostradas as temperaturas dos termômetros das painelas, alimentadas pelos eventos disparados pelo bloco principal. Ademais, alarmes para erros que aconteçam, como bomba sem líquido, resistência sem aquecer e acionamento de válvulas manuais podem ser mostrados através da função de alarmes. Interações com o bloco da nuvem podem ser feitas autonomamente ou disparadas através de chamadas pelo bloco principal, como atualização de relatórios que estejam em exibição e acabaram de ser alimentados por ele.

Finalmente, o bloco que compreende funcionalidades de nuvem coletaria os dados medidos e os compartilharia, permitindo um ajuste do algoritmo utilizado para o controle dos componentes e futuras atualizações.

Vale notar que existe suporte para conexão SSL pelo NodeMCU, apesar da pouca memória ser um obstáculo no uso de conexões seguras, não é impeditivo para aplicações simples de envio de dados como a proposta. De outra forma, chaves de conexão e dados trafegariam na rede sem nenhuma segurança.

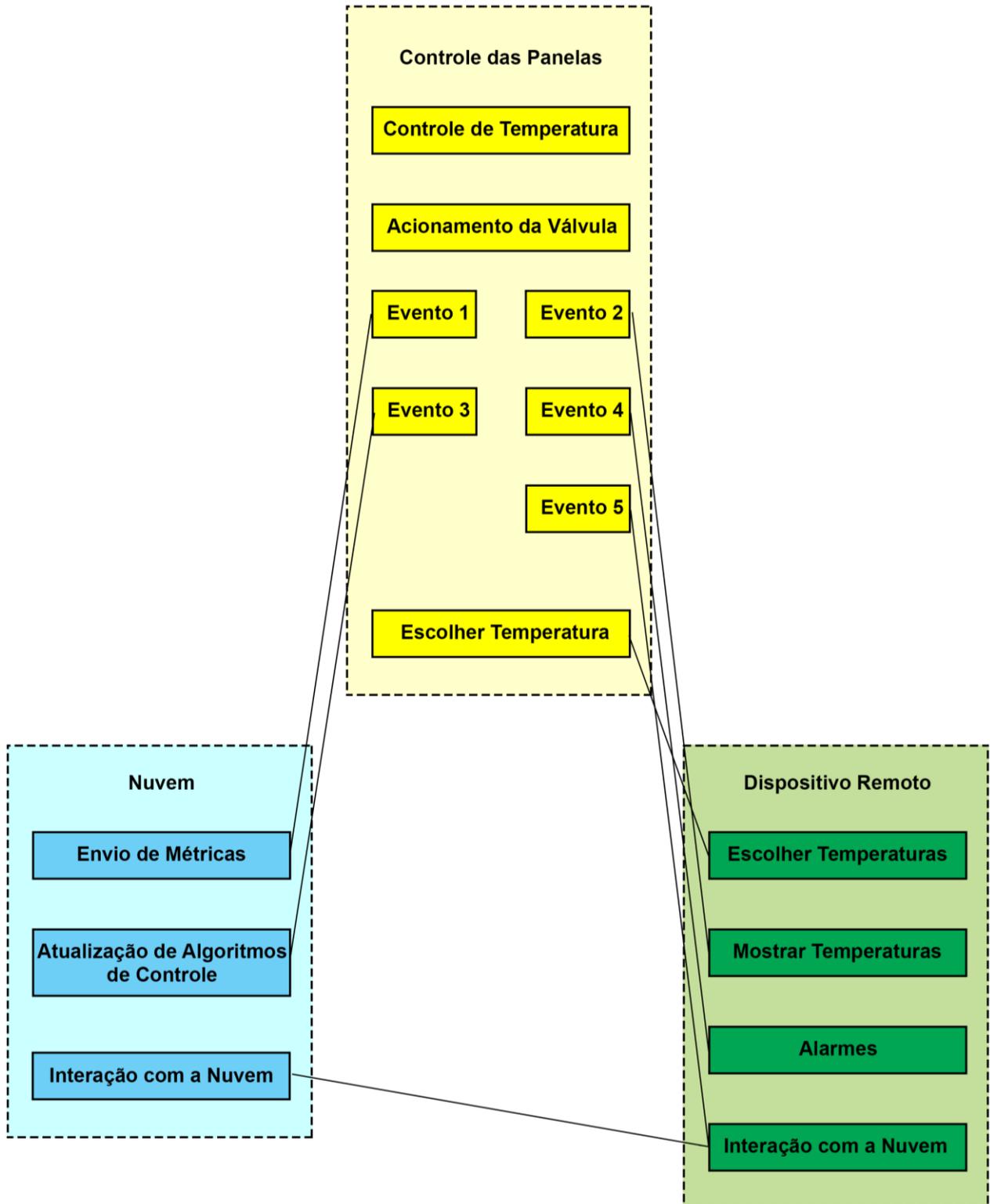
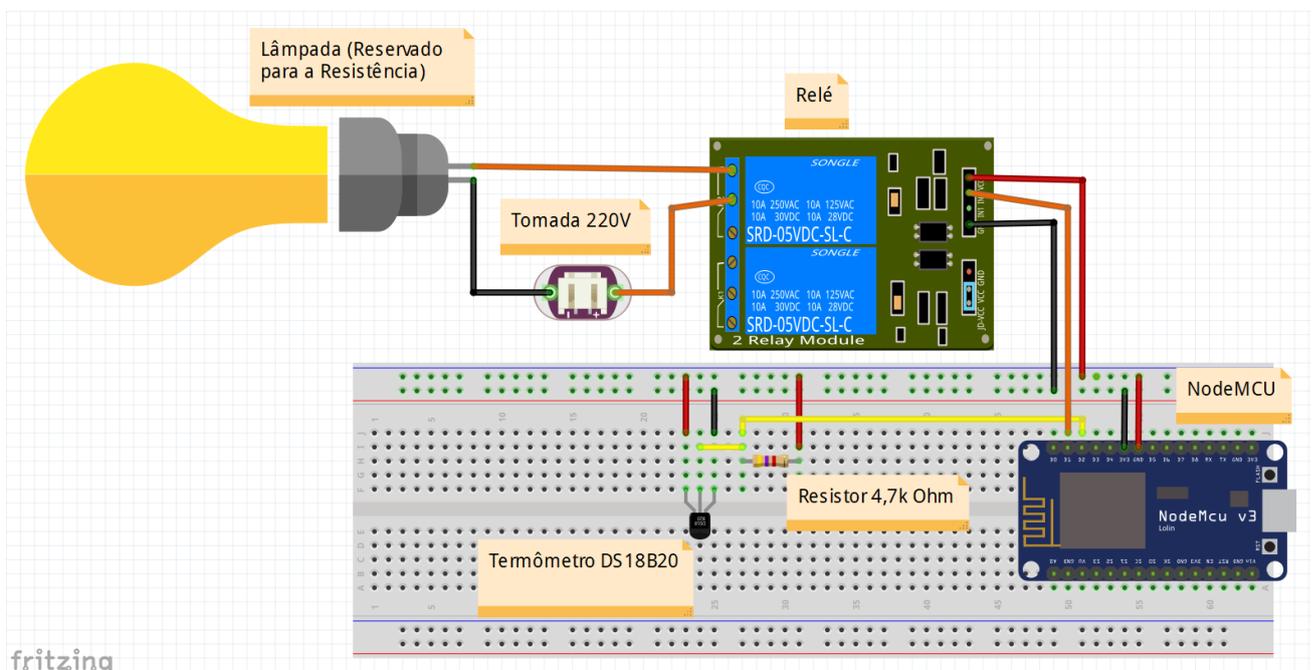


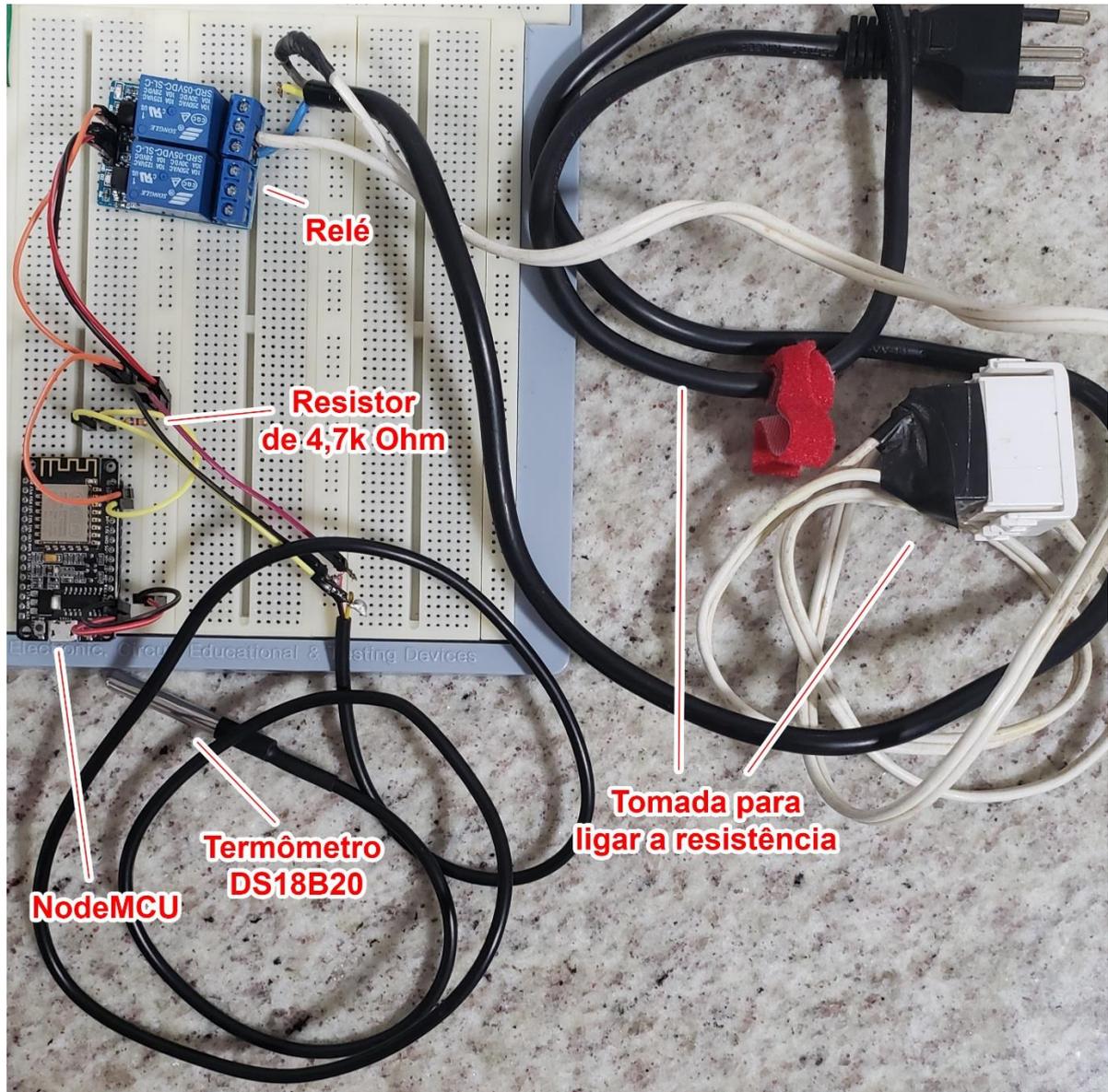
Figura 35. Arquitetura do Algoritmo

### 3.1.5 Prova de Conceito

Buscando validar toda esquemática proposta, a placa NodeMCU foi usada num protótipo simulando a função de aquecimento da panela de lavagem. Seguiu-se a montagem segundo o circuito diagramado a Figura 36 e montado na Figura 37. Vale observar que existe uma lâmpada no diagrama do circuito de modelo simulando a resistência. É uma confecção praticamente direta do que foi projetado na Figura 34, com exceção da substituição do SSR pelo relé comum SRD-05VDC-SL-C, já disponível para uso, eliminando a necessidade da aquisição de um novo componente, e sem impactos nos propósitos do experimento.

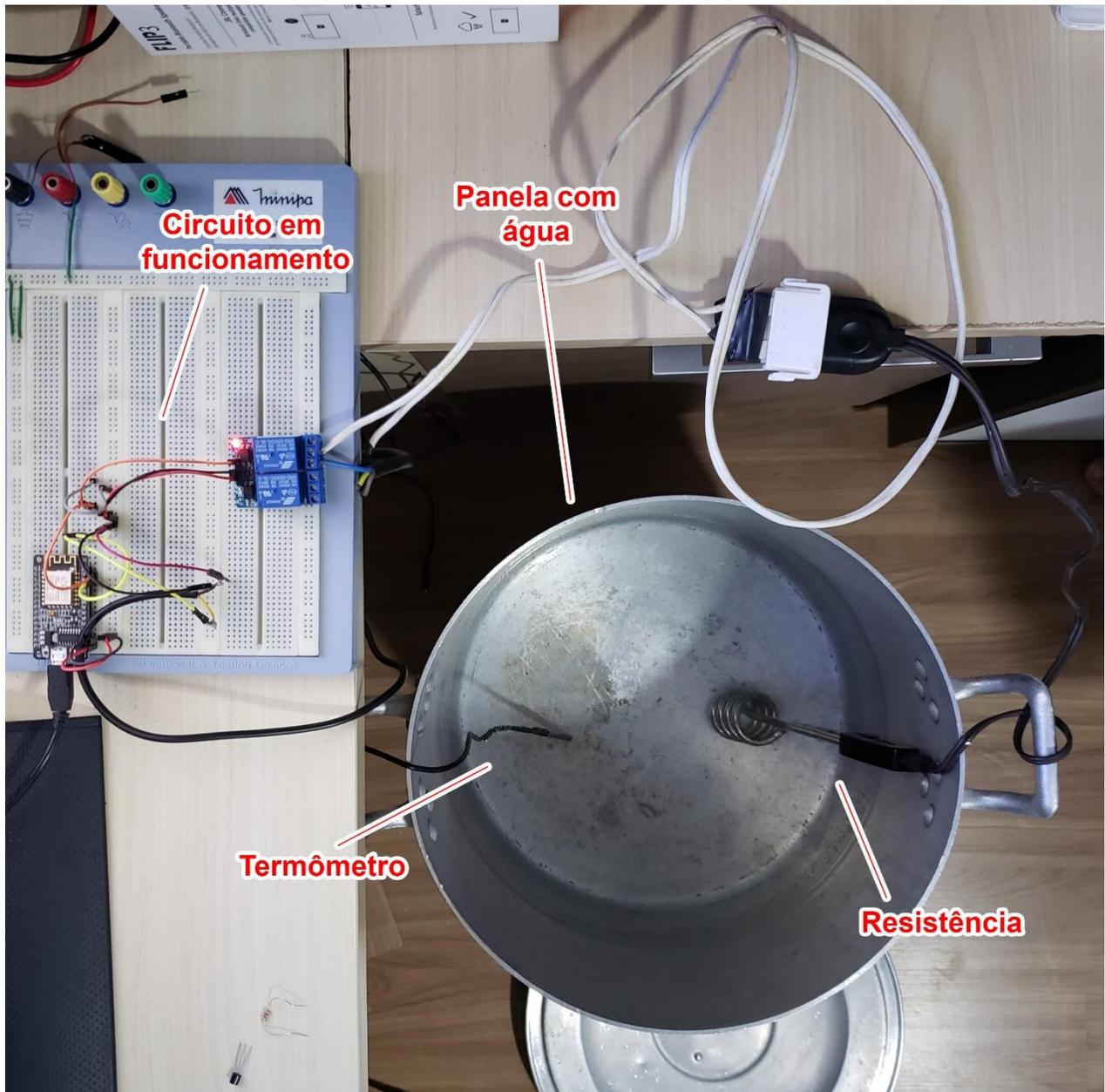


**Figura 36.** Diagrama do Circuito Elétrico feito na Ferramenta Fritzing.



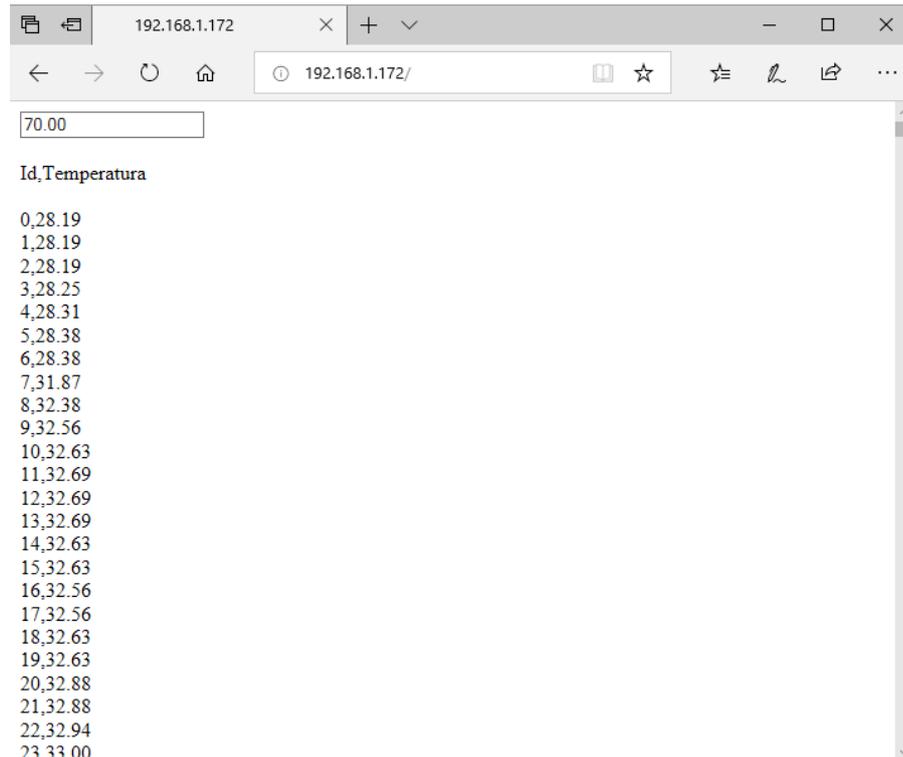
**Figura 37.** Protótipo do Circuito Elétrico Implementado em Placa de Prototipação.

Utilizando uma das ferramentas disponíveis para a NodeMCU foi escrito código utilizando Arduino IDE que permitisse o acesso da placa pela rede sem fio para definir a temperatura desejada e acompanhar a sua evolução no líquido na panela. Além de controlar o relé que aciona a resistência, conforme o comportamento temperatura. A abordagem utilizada foi a de *on-off*: conforme a temperatura é atingida, a resistência é desligada, e se a temperatura cai abaixo do desejado, ela é religada. Tudo pode ser visto montado na Figuras 38. O código funcional pode ser observado no Apêndice B.



**Figura 38.** Panela Montada com Resistencia Termômetro e Circuito Elétrico.

O acesso ao sistema é feito pela rede local. A placa é atribuída um endereço IP automaticamente e anuncia pelo terminal. Quando acessado pelo browser, esse endereço leva a uma página que mostra uma caixa de texto com a temperatura alvo e uma lista com as temperaturas aferidas a cada 15 segundos, numeradas de 0 em diante, como é visto na Figura 39.



**Figura 39.** Programa em Execução

A partir do experimento foi possível extrair alguns dados e observar o comportamento do experimento através dos gráficos das figuras 40 e 41. Com 10 litros de água, uma resistência de 1kW conseguiu elevar a temperatura a uma velocidade de 1,2°C/min, com 15 litros caiu para 0,8°C/min. O ideal seria 1°C/min para que a brasagem ocorra num tempo confortável, e se evite que o mosto carameliza na resistência elétrica. Considerando a panela de mostura e as perdas de calor com a recirculação da bomba, seria interessante manter a proporção de 1kW/10L de água, assim se tem potência para alguma necessidade de uso em algum dia mais frio, ou mais ventilado.

No primeiro experimento com os 10 litros a temperatura alvo foi 70°C, a resistência foi configurada para desligar/ligar a -2°C da temperatura alvo, temendo a dilatação térmica, no caso 68°C. A temperatura se manteve entre 66,3°C e 65,2°C. Um delta de 1,1°C.

No experimento seguinte, com 15 litros, mantendo o alvo em 70°C e o limiar em 0°C (desligando/ligando nos mesmos 70°C), ela se manteve entre 70,6°C e 69,8°C do alvo. Delta de 0,8°C. Como a busca é que não se passe da temperatura

alvo em nenhum momento, novos testes com 1°C a 0,5°C podem permitir chegar numa variação que satisfaça os requisitos e permitam que o usuário se preocupe apenas em configurar a temperatura alvo.

Uma última observação é que com 10 litros o experimento estabilizou a temperatura mais rapidamente, por isso precisou de menos tempo, e foi executado por apenas 45 minutos, contra 75 minutos quando com 15 litros. E apesar de dar a impressão de queda, o gráfico da Figura 40 se estabiliza nos últimos 4 minutos. Ambos os gráficos apresentam um *overshooting* das temperaturas, ela é mais acentuada no primeiro, o que ocasiona uma queda maior rumo a linha estável de temperatura, provocando uma impressão de decrescente.

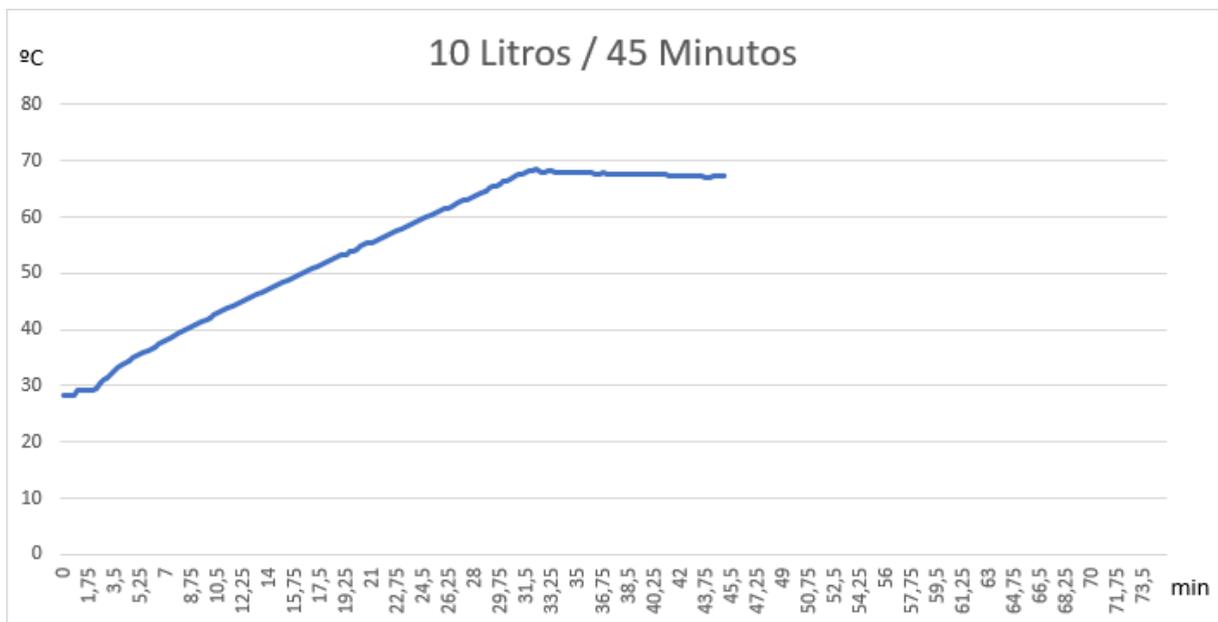


Figura 40. Gráfico para 10 litros

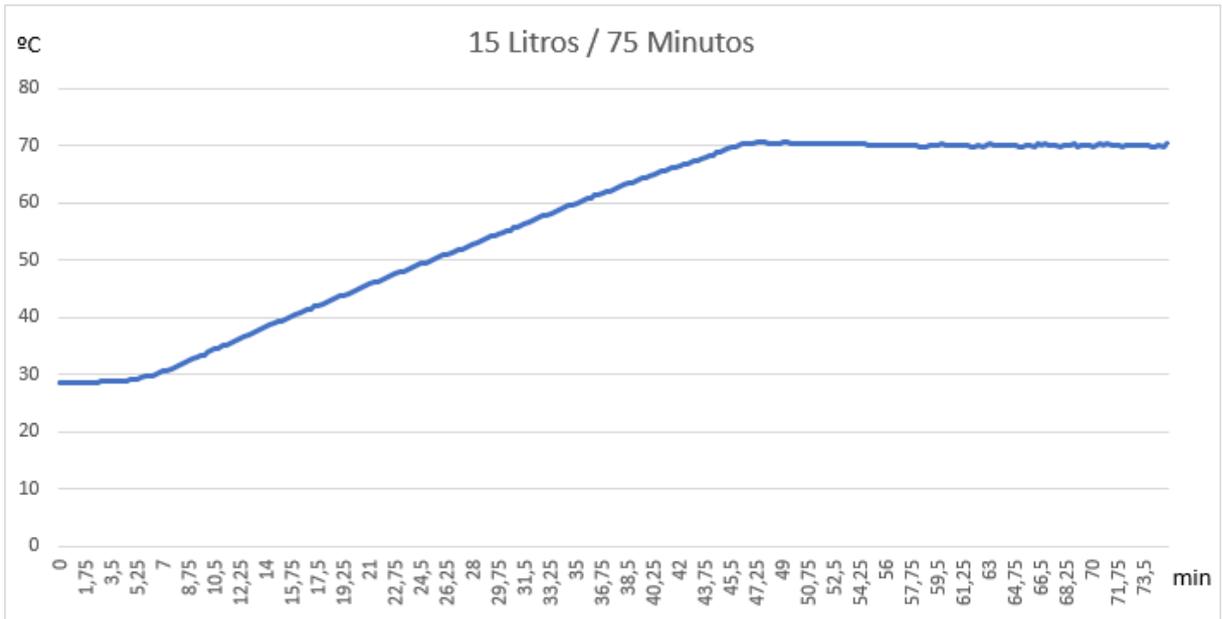


Figura 41. Gráfico para 15 litros

# Capítulo 6

## Conclusão e Trabalhos Futuros

### 4.1 Conclusão

Este trabalho propõe o uso de uma placa controladora NodeMCU para automatizar uma cozinha caseira permitindo a aplicabilidade de conceito de IoT tanto na execução do processo quanto em possíveis implementações futuras.

O projeto em foco é desenvolver uma proposta de *hardware* e *software*, dotados de materiais eletrônicos e algoritmos de controle e interação, que englobe todo o procedimento. A construção teve por opção a escolha de equipamentos eletrônicos consolidados e de uso simples, além do custo baixo e adequação com a proposta do uso alimentício quando em contato direto com a cerveja, focando em atingir o objetivo deste. Problemas foram encontrados ao se projetar o circuito elétrico. O número de conectores necessários eram muito maiores que os disponíveis. Desta forma algumas válvulas tiveram que ser migradas para o acionamento manual, com um sistema de alarmes disparados pelo sistema instruindo o usuário quando acioná-las.

O objetivo principal pode ser atingido, vez que a interação com as válvulas representa apenas uma fração do tempo total no processo como um todo, e aliado ao sistema de alarmes e ao controle de temperaturas, o projeto apresenta grande conveniência e possibilidade de ganhos em eficiência para o cervejeiro.

### 4.2 Trabalhos Futuros

A implementação de um algoritmo que usasse uma abordagem mais inteligente que *on-off* para o controle da resistência seria bem-vinda. Uma vez que o foco é sempre não passar da temperatura alvo, tentando se manter o mais próximo possível dela, um algoritmo de controle PI (*Proportional-Integral*) adaptativo se

apresenta ideal para essa situação, capaz de ajustar o controle necessário ao erro lido entre a temperatura atual e a desejada com ação proporcional e integral.

O uso de um circuito multiplexador para o acionamento dos componentes restantes, como o MUX74HC4067, por exemplo, é necessário para se integrar todos os componentes desejados ao NodeMCU.

O desenvolvimento completo do sistema proposto, a elaboração de um esquema de relatórios para acompanhamento da evolução da eficiência no uso do equipamento, o polimento da interface, a evolução do sistema de controle de temperaturas para um esquema de rampas no lugar do atual que vai de 0% a 50% e a 100%. São apenas mais alguns trabalhos futuros possíveis.

---

# Referências

- [1] ATZORI, Luigi; IERA, Antonio; MORABITO, Giacomo. The Internet of Things: A survey. *Computer Networks*, [s.l.], v. 54, n. 15, p.2787-2805, out. 2010. Elsevier BV. <http://dx.doi.org/10.1016/j.comnet.2010.05.010>.
- [2] BARNAGHI, Payam; SHETH, Amit. The Internet of Things: The Story So Far. *IEEE – Internet of Things*. 09 set. 2014. Disponível em: <<https://iot.ieee.org/newsletter/september-2014/the-internet-of-things-the-story-so-far.html>>. Acesso em: 17 abr. 2018.
- [3] BEER JUDGE CERTIFICATION PROGRAM. **Style guidelines for beer, mead and cider**. 2015 edition. Disponível em: <[http://www.bjcp.org/docs/2015\\_Guidelines\\_Beer.pdf](http://www.bjcp.org/docs/2015_Guidelines_Beer.pdf)> . Acesso em: 13 abr. 2018.
- [4] **BRAUMEISTER of all brewing devices**. Disponível em: <<https://www.speidels-braumeister.de/en/braumeister/10-20-50-litre-braumeister.html>>. Acesso em: 29 abr. 2018.
- [5] Build NodeMCU. Disponível em: < <https://nodemcu-build.com/>> Acesso em: 28 abr. 2018.
- [6] DRAGON Bier. Disponível em: < <http://microcervejarias-dragonbier.com.yolasite.com/cozinha-1000.php>>. Acesso em: 29 abr. 2018.
- [7] ESP8266EX Datasheet Version 4.3. Disponível em: [https://www.espressif.com/sites/default/files/documentation/0a-esp8266ex\\_datasheet\\_en.pdf](https://www.espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf)> Acesso em: 28 abr. 2018.
- [8] FACCIONI FILHO, Mauro. **Internet das Coisas**. Palhoça: Unisul Virtual, 2016.
- [9] **HOW long does it take to brew a batch of beer?**. Disponível em: <<https://www.brewersfriend.com/2012/03/10/how-long-does-it-take-to-brew-a-batch-of-beer/>>. Acesso em: 25 mar. 2018.

- 
- [10] KOLBAN, Neil. **Kolban's Book on ESP8266**. Texas, USA. 2015.
- [11] Magnificent Multitude of Beer. Disponível em: <<https://craftbeeracademy.com/wp-content/uploads/2013/10/magnificent-chart-of-beer-full.jpg>> . Acesso em: 13 abr. 2018.
- [12] MARCUSSO, Eduardo Fernandes; MÜLLER, Carlos Vitor. **A cerveja no brasil: O ministério da agricultura informando e esclarecendo**. Ministério da Agricultura, 2017. Disponível em: <<http://www.agricultura.gov.br/assuntos/inspecao/produtos-vegetal/pasta-publicacoes-DIPOV/a-cerveja-no-brasil-28-08.pdf>>. Acesso em: 25 mar. 2018.
- [13] MARTINS, Sílvio de Melo. **Como fabricar cerveja**. São Paulo: Ícone Editora LTDA, 1991.
- [14] MEDINA, Renata; CAON, Glauco. XIX curso de fabricação caseira de cerveja. PUCRS, 2015. Disponível em: <[https://kupdf.com/download/curso-de-fabrica-ccedil-atilde-o-caseira-de-cerveja-pucrs\\_59f0ffebe2b6f58c05afe5eb\\_pdf](https://kupdf.com/download/curso-de-fabrica-ccedil-atilde-o-caseira-de-cerveja-pucrs_59f0ffebe2b6f58c05afe5eb_pdf)> Acesso em: 3 abr. 2018.
- [15] MINERVA, Roberto; BIRU, Abyi; ROTONDI, Domenico. **Towards a Definition of the Internet of Things (IoT)**. IEEE Internet Initiative - Telecom Italia. 27 maio 2015. Disponível em: <[https://iot.ieee.org/images/files/pdf/IEEE\\_IoT\\_Towards\\_Definition\\_Internet\\_of\\_Things\\_Revision1\\_27MAY15.pdf](https://iot.ieee.org/images/files/pdf/IEEE_IoT_Towards_Definition_Internet_of_Things_Revision1_27MAY15.pdf)>. Acesso em: 25 mar. 2018.
- [16] MORADO, Ronaldo. **Larousse da cerveja**. São Paulo: Larousse do Brasil, 2009. ISBN 978-85-7635-394-2.
- [17] MÜLLER, Arno. **Cerveja**. Canoas: Editora Ulbra, 2002. ISBN 85-7528-058-9
- [18] NACHEL, Marty. **Cerveja para leigos**. Rio de Janeiro: Alta Books, 2013. ISBN 978-85-7608-827-1.
- [19] NodeMCU Documentation. Disponível em: <<http://nodemcu.readthedocs.io/en/master/en/>> Acesso em: 28 abr. 2018.

- [20] NodeMCU. Disponível em: <[http://nodemcu.com/index\\_en.html](http://nodemcu.com/index_en.html)> Acesso em: 14 jun. 2018.
- [21] PAPAZIAN, C. The Complete joy of Homebrewing. 3 ed. New York, NY. 2003. Harper Collins Publishers Inc.
- [22] PRASAD, R.durga et al. NodeMcu based Waste Management system for smart city. International Journal Of Applied Sciences, Engineering And Management. Sattenapalli, p. 102-105. mar. 2017
- [23] Processo de Fabrico da Cerveja Disponível em: <[http://www.esac.pt/noronha/pgs/0910/Trabalhos\\_mod1/CERVEJA.pdf](http://www.esac.pt/noronha/pgs/0910/Trabalhos_mod1/CERVEJA.pdf)> Acesso em: 13 abr. 2018.
- [24] SANTOS, S.P. dos. Os primórdios da cerveja no Brasil. 2.ed. Cotia: Ateliê Editorial, 2004.
- [25] SHETH, Jagdish N.; SISODIA, Rajendra S.; WOLFE, David B. Os segredos das empresas mais queridas: como empresas de classe mundial lucram com a paixão e os bons propósitos. São Paulo: Bookman Companhia Editora, 2007.
- [26] VARGAS, V. **Cultos em cerveja: discursos sobre a cerveja artesanal no Brasil**. Sociedade e Cultura, vol. 18, nº. 1, enero-junio, p. 101-111, 2015.

# Apêndice A

## Trechos de códigos de referência do NodeMCU

```
-- a simple HTTP server
srv = net.createServer(net.TCP)
srv:listen(80, function(conn)
  conn:on("receive", function(sck, payload)
    print(payload)
    sck:send("HTTP/1.0 200 OK\r\nContent-Type: text/html\r\n\r\n<h1> Hello, NodeMCU.</h1>")
  end)
  conn:on("sent", function(sck) sck:close() end)
end)
```

```
-- connect to WiFi access point
wifi.setmode(wifi.STATION)
wifi.sta.config("SSID", "password")
```

---

```
-- register event callbacks for WiFi events
wifi.sta.eventMonReg(wifi.STA_CONNECTING, function(previous_state)
  if(previous_state==wifi.STA_GOTIP) then
    print("Station lost connection with access point. Attempting to reconnect...")
  else
    print("STATION_CONNECTING")
  end
end)
end)
```

```
-- manipulate hardware like with Arduino
pin = 1
gpio.mode(pin, gpio.OUTPUT)
gpio.write(pin, gpio.HIGH)
print(gpio.read(pin))
```

# Apêndice B

## Código do programa da prova de conceito

```
#include <ESP8266WiFi.h>
#include <ESP8266WebServer.h>
#include <WiFiClient.h>
#include <OneWire.h>
#include <DallasTemperature.h>

//Resistencias
#define PINO_RES D1 //Pino da resistencia 1

//Termometro
#define PINO_TERM D2 //Pino do termometro
#define MAX_HISTORICO 1440 //Maximo de temperaturas registradas a cada 15s

OneWire oneWire(PINO_TERM);
DallasTemperature DS18B20(&oneWire);
long ultimaMedicao;
long ultimaHist;
const int deltaTemp = 1000;
const int deltaHist = 15000;
float tempHist[MAX_HISTORICO];
int indTempHist = 0;
float tempAtual;
float tempDesejada = 50;

//WIFI
const char* ssid = ""; //Definir Rede
const char* password = ""; //Senha da Rede

//HTTP
ESP8266WebServer server(80);

void SetupTermometro(){
    DS18B20.begin();

    ultimaMedicao = millis();
    ultimaHist = ultimaMedicao - deltaHist;
    DS18B20.requestTemperatures();
    float temp = DS18B20.getTempCByIndex(0);
    tempAtual = temp;
    Serial.println(temp);
}
```

```
void HandleTermometro(long now) {
    if( now - ultimaMedicao > deltaTemp ){
        float tempC = DS18B20.getTempCByIndex(0);
        tempAtual = tempC;
        Serial.println(tempC);
        if( now - ultimaHist > deltaHist ){
            if (indTempHist< MAX_HISTORICO){
                Serial.println("----");
                Serial.println(tempC);
                tempHist[indTempHist] = tempC;
                ultimaHist = millis();
                indTempHist++;
            }
        }
        DS18B20.waitForConversion(false);
        DS18B20.requestTemperatures();
        ultimaMedicao = millis();
    }
}

void SetupResistencia() {
    pinMode(PINO_RES, OUTPUT);
}

void HandleResistencia() {

    if (tempDesejada - tempAtual > 0) {
        digitalWrite(PINO_RES, HIGH);
    } else {
        digitalWrite(PINO_RES, LOW);
    }
    delay(1000);
}

void HandleRoot() {

    String message = "<input type='text' value='"+String(tempDesejada)+"'/>";
    message += "<br><br>";
    message += "Id, Temperatura<br><br>";
    for(int i=0; i<MAX_HISTORICO; i++){
        message += i;
        message += ",";
        message += tempHist[i];
        message += "<br>";
    }

    server.send(200, "text/html", message );
}

void HandleTemperatura() {

    if (server.arg("Temperatura")== ""){
    }else{
        tempDesejada = server.arg("Temperatura").toInt();
    }

    HandleRoot();
}
}
```

```
void HandleTemperatura() {

    if (server.arg("Temperatura")== ""){
    }else{
        tempDesejada = server.arg("Temperatura").toInt();
    }

    HandleRoot();
}

void HandleNotFound(){
    String message = "File Not Found\n\n";
    message += "URI: ";
    message += server.uri();
    message += "\nMethod: ";
    message += (server.method() == HTTP_GET)?"GET":"POST";
    message += "\nArguments: ";
    message += server.args();
    message += "\n";
    for (uint8_t i=0; i<server.args(); i++){
        message += " " + server.argName(i) + ": " + server.arg(i) + "\n";
    }
    server.send(404, "text/html", message);
}

void setup() {

    Serial.begin(9600);

    WiFi.begin(ssid, password);
    Serial.println("");

    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print(".");
    }
    Serial.println("");
    Serial.print("Connected to ");
    Serial.println(ssid);
    Serial.print("IP address: ");
    Serial.println(WiFi.localIP());

    server.on("/", HandleRoot);
    server.on("/Temperatura", HandleTemperatura);
    server.onNotFound( HandleNotFound );
    server.begin();
    Serial.println("HTTP server started at ip " + WiFi.localIP().toString() );

    SetupTermometro();
    SetupResistencia();
}

void loop() {
    long t = millis();

    server.handleClient();
    HandleTermometro( t );
    HandleResistencia();
}
```