

**JOSÉ HENRIQUE CAMARGO LEOPOLDO E SILVA
RAFAEL JARDIM PASTOR**

**SISTEMA DE MONITORAMENTO E CONTROLE
DA FERMENTAÇÃO DE CERVEJAS**

São Paulo
2020

**JOSÉ HENRIQUE CAMARGO LEOPOLDO E SILVA
RAFAEL JARDIM PASTOR**

**SISTEMA DE MONITORAMENTO E CONTROLE
DA FERMENTAÇÃO DE CERVEJAS**

Trabalho apresentado à Escola Politécnica
da Universidade de São Paulo para ob-
tenção do Título de Engenheiro Eletricista.

São Paulo
2020

**JOSÉ HENRIQUE CAMARGO LEOPOLDO E SILVA
RAFAEL JARDIM PASTOR**

**SISTEMA DE MONITORAMENTO E CONTROLE
DA FERMENTAÇÃO DE CERVEJAS**

Trabalho apresentado à Escola Politécnica
da Universidade de São Paulo para ob-
tenção do Título de Engenheiro Eletricista.

Orientador:

Prof. Dr. Carlos Eduardo Cugnasca

São Paulo
2020

Autorizo a reprodução e divulgação total ou parcial deste trabalho, por qualquer meio convencional ou eletrônico, para fins de estudo e pesquisa, desde que citada a fonte.

Catálogo-na-publicação

Silva, José Henrique Camargo Leopoldo
Sistema de Monitoramento e Controle da Fermentação de Cervejas / J. H.
C. L. Silva, R. J. Pastor -- São Paulo, 2020.
79 p.

Trabalho de Formatura - Escola Politécnica da Universidade de São
Paulo. Departamento de Engenharia de Computação e Sistemas Digitais.

1.CERVEJA 2.FERMENTAÇÃO ALCOÓLICA 3.CONTROLE DE
PROCESSOS I.Universidade de São Paulo. Escola Politécnica.
Departamento de Engenharia de Computação e Sistemas Digitais II.t.
III.Pastor, Rafael Jardim

RESUMO

Na produção de cervejas, a fermentação alcoólica é um complexo bioquímico cuja função primária é converter os açúcares extraídos dos grãos maltados em álcool etílico. Esse processo tem grande impacto no sabor, aroma, aparência e textura do produto, sendo o processo mais importante durante a produção da bebida. As principais variáveis desse processo são: a temperatura, que influencia no metabolismo das leveduras, a densidade, que indica a evolução da fermentação, e o pH, que é um indicador adicional de qualidade. O objetivo desse projeto é criar um protótipo de um dispositivo que realize o monitoramento dessas variáveis e controle a temperatura, garantindo resultados mais precisos e reprodutíveis. O sistema é voltado para cervejarias de pequeno e médio porte que desejem monitorar e controlar o processo durante os testes de receitas em pequena escalas de produção, e também para produtores *hobbyistas* que almejem maior controle sobre o processo e resultados mais consistentes. O sistema idealizado é composto por dois subsistemas complementares: (i) um físico: agregando o controlador, hardware com sensores e atuadores e software embarcado; e (ii) um digital: que capta, processa e disponibiliza todas as informações adquiridas para o usuário por meio de uma plataforma online. A execução do projeto foi guiada por iterações curtas e prototipagem, focando na implementação gradativa de funcionalidades a cada iteração até a construção total do protótipo.

Palavras-Chave – Produção de Bebidas, Fermentação Alcoólica, Cerveja, Controle de Processos.

ABSTRACT

During beer production, the alcoholic fermentation is a complex biochemical process, which primary function is to convert the sugar, extracted from the malted grains, in ethyl alcohol. This process has great impact on the flavor, scent, aspect and texture of the product, being the most important process in beer production. The main variables during the process are: temperature, which impacts on yeast metabolism, specific gravity, which indicates fermentation evolution, and pH, a extra quality indicator. The object of this project is to develop a prototype able to monitor these variables and control the temperature, allowing more precise and reproducible results. The system is target to small and medium breweries that desires fermentation monitoring and control during small scale recipes tests, and also to homebrewers yearning a better control over the process and more consistent results. The system is composed by two complementary subsystems: (i) a physical one, consisting in the controller, hardware with sensors and actuator and embedded software; and a digital one which receive, process and presents all the data acquired in a web application to the user. The project execution was guided by short iterations and prototyping, aiming at continuous feature development until the final version of the prototype.

Keywords – Beverage Production, Alcoholic Fermentation, Beer, Production Control.

LISTA DE FIGURAS

1	Gráfico de evolução de variáveis ao longo da fermentação.	20
2	Desenho do problema de troca de calor entre o ambiente e fermentador. . .	26
3	Bloco de fonte de calor, que mantém a temperatura em um ponto do circuito constante.	27
4	Bloco de transferência de calor por convecção, seguindo a fórmula $Q = k \cdot A \cdot (T_A - T_B)$	27
5	Bloco de transferência de calor por radiação, seguindo a fórmula $Q = k \cdot A \cdot (T_A^4 - T_B^4)$	27
6	Bloco de transferência de calor por condução, seguindo a fórmula $Q = k \cdot \frac{A}{D} (T_A - T_B)$	27
7	Bloco de sensor de calor ideal	28
8	Circuito de blocos utilizado para a simulação térmica.	28
9	Seção do circuito da figura 8 referente à transferência de calor entre ambiente e isolamento do fermentador.	29
10	Seção do circuito da figura 8 referente à transferência de calor pelo isolamento do fermentador.	29
11	Seção do circuito da figura 8 referente à transferência de calor pelas paredes do fermentador.	30
12	Seção do circuito da figura 8 referente à transferência de calor entre paredes do fermentador e mosto em fermentação.	30
13	Curva de calor obtida a partir da simulação do circuito da figura 8.	31
14	Circuito térmico completo, com atuador e massa térmica.	31
15	Curva de temperatura em Kelvin por tempo em dezenas de milhares de segundos, obtida da simulação do circuito térmico da figura 8.	32
16	Curva de quantidade de calor em Watts por tempo em dezenas de milhares de segundos, obtida da simulação do circuito térmico da figura 8.	32

17	Circuito utilizado para sintonização do controlador.	33
18	Curva de temperatura em Kelvin por tempo em dezenas de milhares de segundos, obtida da simulação de sintonização do controlador.	33
19	Curva de calor em Watts fornecido pelo atuador por tempo em dezenas de milhares de segundos, obtida da simulação de sintonização do controlador.	34
20	Malha de controle PID.	35
21	Resposta do Sistema em malha aberta a sinal degrau.	36
22	Wemos D1	37
23	Sensor DS18B20	39
24	Sensor de diferença de pressão MP3V5010DP acoplado ao chip Pmod DPG1	41
25	Representação de um corpo (azul) submerso em um fluido (amarelo) e das forças gravitacional e de empuxo	42
26	Sensor de carga ZHIPU-200g	42
27	Sensor de pH E-201-C	43
28	Célula de Peltier modelo TEC1-12706.	44
29	Esquema de funcionamento de uma célula de Peltier.	44
30	Dissipador de calor e ventoinha, modelo utilizado em CPU.	45
31	Haste de Aço inoxidável.	45
32	Diagrama de Casos de Uso.	50
33	Diagrama de Entidade-Relacionamento.	51
34	Camada de Interface.	52
35	Camada Intermediária.	54
36	Camada de Negócio - Servidor de Registros.	55
37	Camada de Negócio - Servidor de Receitas.	55
38	Camada de Negócio - Servidores de Análises, Processamento e Autenticação.	56
39	Camada de Persistência.	56
41	Fluxo de telas e interações da Interface Web.	57

40	Diagrama de Componentes.	59
42	Esquema de conexão entre microcontrolador e sensor de temperatura DS18B20.	60
43	Esquema de conexão entre microcontrolador e sensor de pH E-201-C.	61
44	Modelo ponte H - L298N.	64
45	Onda quadrada do sinal PWM para alguns valores especificados no micro-controlador.	65
46	Esquema do perfil do dispositivo controlador de temperatura, escala aproximada.	66
47	Exemplo de gráfico de evolução das variáveis.	70

LISTA DE TABELAS

1	Descrição dos símbolos de trocas de calor referentes às figuras 4 a 6.	28
2	Descrição das variáveis do controlador PID da Equação 4.2.	35
3	Parâmetros do controlador PID seguindo método de Ziegler-Nichols.	36
4	Diferenças de pressão esperadas para cálculo da densidade relativa durante a fermentação.	40
5	Medições de tensão em soluções tampão para calibração do sensor de pH. .	62

SUMÁRIO

1	Introdução	13
1.1	Objetivo	13
1.2	Motivação	13
1.3	Organização do Trabalho	13
2	Contexto	15
2.1	História da Cerveja	15
2.2	Fabricação de Cervejas	16
2.2.1	Fermentação e Maturação	17
3	Metodologia do Trabalho	21
4	Projeto	22
4.1	Especificação de Requisitos Técnicos	22
4.1.1	Sistema Hardware	22
4.1.1.1	Requisitos funcionais de Hardware	22
4.1.1.2	Requisitos não funcionais de Hardware	23
4.1.2	Sistema Software	23
4.1.2.1	Requisitos funcionais de Software	23
4.1.2.2	Requisitos não funcionais de Software	24
4.2	Projeto de Controle	24
4.2.1	Modelagem Térmica do Sistema	24
4.2.2	Transferência de Calor entre Ar e Fermentador	25
4.2.3	Simulação Térmica	26
4.2.4	Modelo para Sintonização do Controlador	32

4.2.5	Dispositivo para Troca de Calor	34
4.2.6	Controle de Temperatura	34
4.2.6.1	Controlador PID	35
4.2.6.2	Método de Ziegler-Nichols	36
4.2.6.3	Anti Wind-up	36
4.3	Projeto de Hardware	37
4.3.1	Microcontrolador	37
4.3.2	Critérios de Escolha dos Sensores	38
4.3.3	Sensor de Temperatura	38
4.3.4	Sensor de Densidade Relativa	39
4.3.4.1	Medição por diferença de pressão	40
4.3.4.2	Medição por Empuxo	41
4.3.5	Sensor de pH	42
4.3.6	Atuador de Temperatura	43
4.4	Projeto de Software	45
4.4.1	Especificação dos Casos de Uso	46
	UC - 1: Registro de Dispositivo	47
	UC - 2: Cadastro de Receitas	47
	UC - 3: Cadastro de Lotes	48
	UC - 4: Envio das informações do Lote para Dispositivo	48
	UC - 5: Envio das informações do Dispositivo para o Sistema	49
	UC - 6: Visualização das Informações dos Lotes	49
4.4.2	Modelo de Entidade-Relacionamento	50
4.4.3	Arquitetura de Software	51
4.4.3.1	Camada de Interface	52
4.4.3.2	Camada Intermediária	53

4.4.3.3	Camada de Negócio	54
4.4.3.4	Camada de Persistência	56
4.4.4	Fluxo de Interação Interface Web	57
4.4.5	Definição de Tecnologias	57
5	Implementação	60
5.1	Implementação de Hardware	60
5.1.1	Sensor de Temperatura	60
5.1.2	Sensor de pH	61
5.1.3	Sensor de pressão diferencial para medida de densidade	62
5.1.4	Conexão e Comunicação do Microcontrolador	63
5.1.5	Controle de Temperatura	63
5.1.5.1	Circuito Ponte H	64
5.1.5.2	Montagem do Dispositivo	65
5.2	Implementação de Software	66
5.2.1	Camada de Persistência	67
5.2.2	Camada de Negócio	67
5.2.2.1	Servidores de Registro e Receitas	67
5.2.2.2	Servidor de Processamento	68
5.2.2.3	Servidor de Análises	68
5.2.3	Camada de Aplicação	69
5.2.4	Interface Web	69
5.2.5	Implantação	69
6	Testes e Avaliação do Protótipo	71
7	Conclusão	73
7.1	Considerações Finais	73

7.2 Perspectivas de Continuidade	74
Referências	75
Apêndice A – Código fonte	76

1 INTRODUÇÃO

1.1 Objetivo

O objetivo geral deste trabalho foi desenvolver um protótipo de um sistema que realize o monitoramento e controle do processo de fermentação de cervejas. O objetivo específico foi possibilitar às cervejarias de pequeno e médio porte o desenvolvimento da capacidade de testes de novas receitas de cervejas, a fim de garantir a reprodutibilidade e qualidade das mesmas por meio do controle do processo de fermentação.

1.2 Motivação

A principal motivação do grupo no projeto foi a ideia de criar um produto que facilitasse a fabricação de cervejas em pequena escala. Escolhemos o tema de cervejas, pois além de ser um interesse em comum dos integrantes, ele se mostra muito importante social e economicamente. Visamos os pequenos produtores, pois os mesmos possuem uma produção mais artesanal, o que representa uma grande oportunidade para soluções simples. Através desse desenvolvimento, o grupo visou aplicar conceitos aprendidos na engenharia elétrica durante a graduação.

1.3 Organização do Trabalho

O presente trabalho é dividido de forma linear, em partes que representam os passos percorridos até a sua conclusão. A primeira parte consiste na contextualização, onde são apontados os pontos principais sobre a cerveja e a sua produção, com um enfoque no processo de fermentação. Após isso, é apresentado o seu projeto, onde são definidas as bases teóricas. Nessa sessão serão enumerados os requisitos funcionais e não funcionais do projeto, que servirão de guia para o resto do projeto de software e hardware. A próxima parte apresentada é a de implementação com a montagem e codificação e readaptação do

projeto. Por último, são discutidos os testes de validação sobre o protótipo e feitas as considerações finais sobre o projeto.

2 CONTEXTO

Esse projeto tem natureza multidisciplinar, buscando harmonizar a bioquímica do processo de produção de cervejas com sistemas de controle e automação e geração e análise de dados estudados na Engenharia Elétrica. Em sua primeira parte, foi estudado o funcionamento das leveduras e como o ambiente age sobre seu processo metabólico, enquanto na outra parte, foram estudados sensores, atuadores e arquitetura de Internet das Coisas para captação, processamento e disponibilização de informações referente a ação dos microrganismos.

2.1 História da Cerveja

Segundo Kunze 2004, a cerveja provavelmente teve origem na revolução agrícola, na qual os humanos começaram a abandonar o nomadismo e se estabeleceram em comunidades que cultivam diversos grãos. Com o armazenamento de cereais, é provável que essa origem tenha sido acidental, com uma fermentação espontânea da cevada. Os mais antigos registros dessa bebida podem ser encontrados na antiga região mesopotâmia, atual Irã, e são datados em 2800 a.C. A sua importância em comunidades antigas do oriente médio era tanta, que em 1760 a.C., foi criada a primeira lei que regulamenta a produção e venda de cerveja. A lei conhecida como Estela de Hamurabi regulamentou a comercialização, fabricação e consumo, estabelecendo uma ração diária de cerveja para os habitantes da região.

Como o consumo da cerveja era mais seguro do que a água, visto que o processo de fervura ajuda a purificar a bebida, seu consumo em algumas sociedades era visto como uma necessidade básica diária, continua Kunze 2004. Esse fato ajudou a aumentar a popularidade da bebida na Europa durante a idade média, principalmente nas comunidades Germânicas. Uma das instituições mais importantes para o desenvolvimento dos processos de produção eram os mosteiros dessa região e época. Os monges eram responsáveis pela fabricação da bebida e como eram os únicos que reproduziam os manuscritos, puderam

conservar e aperfeiçoar a sua produção, sendo muito influentes até hoje. Eles foram responsáveis por incluir diversas ervas na fabricação, sendo a mais importante delas o lúpulo, utilizado para trazer o amargor da bebida. Uma das leis mais conhecidas e importantes da indústria cervejeira é a da pureza Alemã. Devido a diversas mudanças aplicadas pelos fabricantes e a percepção de uma queda de qualidade, essa lei foi criada no século XIV na região da Bavária (Sul da Alemanha) buscando uma padronização da bebida. A lei instituiu que a cerveja deveria ser fabricada apenas com os seguintes ingredientes: água, malte de cevada e lúpulo. Atualmente, apesar da legalização do uso de qualquer ingrediente na região, muitos cervejeiros ainda seguem essa restrição e ela é um sinônimo de qualidade.

Ainda segundo Kunze 2004, uma das mais importantes inovações na fabricação foi a Pasteurização, que permite a preservação do gosto por mais tempo. O processo consiste, basicamente, no aquecimento da bebida a uma determinada temperatura, por determinado tempo, e depois a bebida é resfriada de forma a eliminar os microorganismos ali presentes. O processo foi nomeado pelo seu criador o francês Louis Pasteur que atendendo a solicitação de alguns dos vinicultores e cervejeiros da região que lhe pediram para descobrir como os vinhos e a cervejas azedaram. Durante sua investigação, através do uso de microscópio, ele pôde constatar que a levedura ocasionava este processo e assim criou esse processo de purificação. A partir dele, a indústria cervejeira conseguiu chegar em um novo nível e crescer em escala e alcance, sendo essencial para grandes fábricas atualmente.

2.2 Fabricação de Cervejas

Lewis 2001 apresenta a produção de cerveja como uma atividade que deve equilibrar séculos de tradição e arte, desenvolvida por gerações de cervejeiros, e a abordagem científica e avanços tecnológicos, de forma que estes possam proporcionar maior entendimento, controle e melhorias sobre o processo mas sem abandonar suas raízes históricas e descaracterizá-lo. Justamente, as etapas principais do processo ainda seguem a produção tradicional, apesar da evolução de métodos e técnicas adicionais.

Nesta seção é descrito o processo mais comum de produção de cervejas, considerando os principais ingredientes: água, malte, lúpulo e levedura. Tradicionalmente, são utilizados grãos maltados de cevada, mas atualmente podem ser utilizados outros grãos, maltados ou não, como por exemplo: trigo, milho, arroz e aveia. A partir da descrição de Kunze 2004, são listadas as principais etapas de produção:

1. Maltagem dos grãos: processo de germinação parcial e controlada dos grãos, com a finalidade de produzir enzimas como a amilase. A germinação é interrompida por um processo de secagem quando atinge o estágio desejado. A partir desse momento, os grãos passam a ser denominados malte;
2. Moagem do malte: quebra do malte em pequenos fragmentos para expor as enzimas e componentes internos. É desejável que parte da casca seja mantida intacta para auxiliar na filtragem após a mostura;
3. Mostura: o malte moído é misturado em água, e aquecida em temperaturas que estimulem a ação das enzimas obtidas na maltagem. As enzimas realizam a quebra de moléculas insolúveis de amido em moléculas menores de açúcares, que são dissolvidas, formando uma solução denominada mosto;
4. *Lautering*: o mosto é separado do restante do malte que não foi dissolvido. As cascas dos grãos auxiliam essa etapa formando um filtro natural, mas também podem ser utilizados filtros;
5. Fervura: o mosto é fervido, conseqüentemente: a ação enzimática é interrompida e a solução é esterilizada. Nessa etapa, lúpulos são adicionados em diferentes momentos da fervura, fornecendo extratos que conferem amargor e aroma à cerveja;
6. Fermentação: após o resfriamento do mosto, leveduras são adicionadas e a solução é aerada. Os açúcares são consumidos pelo metabolismo das leveduras, gerando etanol e dióxido de carbono;
7. Maturação: após o consumo dos açúcares, as leveduras passam a reabsorver parte dos subprodutos gerados durante a fermentação, melhorando a qualidade geral da bebida. Ao final do processo, as leveduras floculam e decantam, podendo ser extraídas e reaproveitadas;
8. Envase: transferência para o recipiente final. Nessa etapa, também é realizada a carbonatação da bebida, geralmente por injeção de gás carbônico ou por meio de uma segunda fermentação, aproveitando-se leveduras ainda presentes na solução. Industrialmente, é comum a filtragem pré-envase, e a pasteurização pós- envase.

2.2.1 Fermentação e Maturação

Kunze 2004 posiciona a fermentação como o processo mais importante da produção de cervejas, sendo as etapas antecedentes responsáveis para garantir as condições necessárias

para que as leveduras possam converter os açúcares, extraídos do malte e quebrados em moléculas menores pelas enzimas, em álcool e gás carbônico. As leveduras consomem os açúcares para gerar energia e se reproduzirem, e nesse processo também geram centenas de subprodutos como ésteres, álcoois pesados e compostos sulfúricos que mesmo em pequena quantidade são essenciais para as características organolépticas da cerveja, como pontuam White e Zainasheff 2010.

Leveduras são organismos unicelulares membros do reino fungi. Na produção de cervejas são utilizadas principalmente duas espécies: a *Saccharomyces cerevisie* (em cervejas do tipo *Ale*) e a *Saccharomyces pastorianus* (em cervejas do tipo *Lager*). Ao serem inoculadas ao mosto, as leveduras rapidamente absorvem o oxigênio dissolvido para revitalizar sua membrana celular e iniciar a absorção dos açúcares e nutrientes do mosto, descrevem White e Zainasheff 2010. A partir do açúcar absorvido, a levedura pode convertê-lo em energia por meio da respiração aeróbica que ocorre na presença de oxigênio, ou pela respiração anaeróbica que ocorre na ausência de oxigênio ou em solução com alta concentração de glicose (pelo chamado efeito de Crabtree). Além da produção energia no formato de moléculas de adenosina trifosfato (ATP), a respiração anaeróbica também produz etanol e gás carbônico, por isso também é denominada de fermentação alcoólica; a equação 2.1 ilustra essa reação. A molécula de ATP provém energia para a síntese de proteínas e replicação de DNA, necessários para a multiplicação celular.



A etapa de fermentação e maturação de uma cerveja leva em média entre 7 e 14 dias, desde a inoculação das leveduras até a cerveja estar pronta para envase. Vale notar que podem ser aplicados processos extras na maturação que prolonguem esse tempo. Para uma fermentação básica, White e Zainasheff 2010 classificam o processo em 3 fases, que ocorrem com certo grau de sobreposição:

1. Fase de retardamento, durante as primeiras quatro a 15 horas após a adição da levedura no mosto, caracterizada pela climatização das células ao ambiente e preparação para a próxima fase;
2. Fase de crescimento exponencial, que dura entre quatro horas e quatro dias, quando ocorre o consumo dos açúcares e replicação logarítmica das células de leveduras;
3. Fase estacionária, em que o crescimento diminui e alguns compostos, como o diacetil, são absorvidos pelas leveduras, maturando o produto durante três a dez dias.

A variação nas durações das etapas é decorrente tanto da quantidade e saúde das leveduras inoculadas, quanto da temperatura durante a fermentação. De acordo com White e Zainasheff 2010, tradicionalmente, cervejas do tipo *Ale* são fermentadas a 20°C e cervejas do tipo *Lager* a 10°C. Temperaturas inferiores tornam o processo mais lento, e em casos extremos culminar na parada do processo, impedindo a fermentação completa ou que os subprodutos como o diacetil seja absorvido pelas leveduras. temperaturas superiores aceleram a atividade celular, tornando o processo mais rápido, mas também podendo fazer com que as leveduras se multipliquem demais e gerem uma quantidade de subprodutos alta que pode impactar negativamente na qualidade da bebida, gerando sabores indesejados denominados *off-flavors*. Justamente por essa razão, o controle de temperatura durante a fermentação é essencial para a obtenção de um produto de qualidade com resultados consistentes, principalmente nas primeiras 72 horas do processo, que representam o pico de multiplicação celular, geração de calor decorrente da atividade celular e produção de sub-produtos.

Após a temperatura, White e Zainasheff 2010 colocam a densidade relativa e pH do mosto como segunda e terceira variáveis mais importantes para se monitorar urante a fermentação. A densidade relativa que indica o grau de evolução do processo. Como parte do gás carbônico produzido na equação 2.1 é dissolvido e capturado pela atmosfera, a densidade do mosto diminui durante a fermentação à medida em que os açúcares vão sendo consumidos. A partir disso, é possível: identificar as etapas da fermentação pelo gráfico de evolução da densidade relativa ao longo do tempo; estimar a graduação alcoólica a partir dos valores inicial e final da densidade relativa; e estipular o grau de atenuação, ou quanto dos açúcares foram consumidos pelas leveduras. Enquanto o pH, que sofre uma leve queda durante o processo, pode ser utilizado principalmente como uma amostra das condições iniciais da fermentação, com intuito de se obter reprodutividade do processo, e também pode servir como uma variável de diagnóstico de problemas que podem ocorrer como não adaptação das leveduras ao mosto e contaminação.

Com caráter ilustrativo, é incluído gráfico com o perfil dessas e outras variáveis durante a fermentação de uma cerveja do tipo *Lager* (Figura 1), retirado de Munroe 2006. Na figura, *specific gravity* se refere à densidade relativa do mosto, e *cells in suspension* à quantidade de células de levedura em suspensão, vale notar que ao final do processo, as leveduras floculam e decaem, por isso a queda.

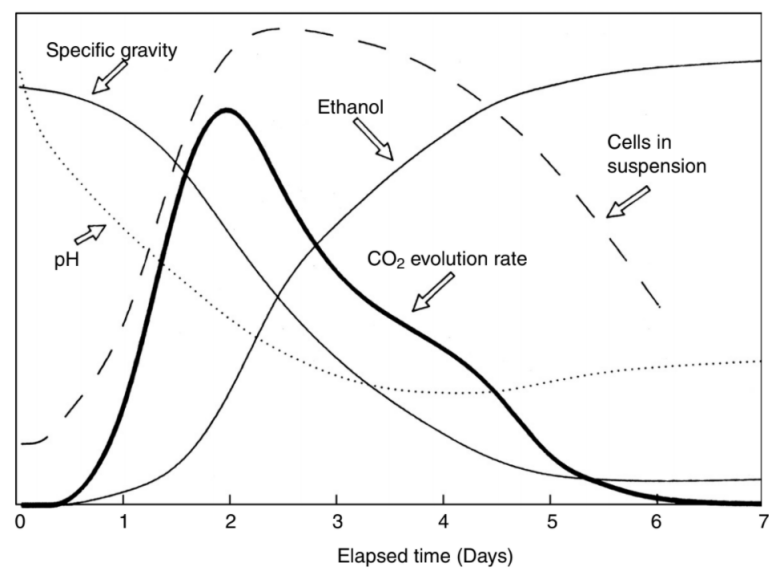


Figura 1: Gráfico de evolução de variáveis ao longo da fermentação.
Fonte: Munroe 2006.

3 METODOLOGIA DO TRABALHO

A elaboração desse projeto segue metodologia típica de projetos de controle de processos, estudada na graduação, consistindo inicialmente de estudo sobre o processo a ser controlado e levantamento dos requisitos funcionais e não funcionais necessários a se atingir o objetivo deste trabalho.

A partir dos requisitos, pretende-se seguir com o desenvolvimento seguindo uma metodologia de prototipagem guiada por iterações curtas com incremento de funcionalidades constante. com o objetivo de acelerar a obtenção de resultados.

Para avaliar cada iteração do protótipo, o processo de produção de cervejas, incluindo a fermentação, foi reproduzido em pequena escala. Essa abordagem prática aguça o conhecimento do processo e tem grande potencial em evidenciar erros no projeto.

4 PROJETO

4.1 Especificação de Requisitos Técnicos

O protótipo deve ser capaz de monitorar e controlar o processo de fermentação de cervejas, seguindo as configurações de receita definidas pelo usuário. Todas as informações coletadas devem ser disponibilizadas ao usuário com a finalidade de possibilitar maior entendimento e reprodutibilidade do processo. O projeto foi dividido em dois sistemas: um Hardware, encarregado das medições e controle, e um Software, responsável por exibir informações e estabelecer uma interface com o usuário. A partir dessas premissas, foram determinados os seguintes requisitos para cada um dos sistemas.

4.1.1 Sistema Hardware

4.1.1.1 Requisitos funcionais de Hardware

HW-F-1) O sistema deve monitorar a temperatura (entre 0 e 25 °C), com precisão de 0,5 °C e intervalo de 1 minuto.

HW-F-2) O sistema deve monitorar o pH (entre 3 a 7,5) em intervalo de 1 minuto.

HW-F-3) O sistema deve monitorar a densidade relativa (entre 1,000 e 1,150), com precisão de 0,001 em relação à água a 20°C e intervalo de 1 minuto.

HW-F-4) O sistema deve controlar a temperatura de até 20 Litros de mosto em fermentação, com desvio máximo de 0,5°C em relação ao valor definido pelo usuário e diferença máxima de 10°C em relação ao ambiente.

HW-F-5) O sistema deve seguir o perfil de controle (temperatura x tempo) definido pelo usuário no Software.

HW-F-6) Os dados monitorados devem ser enviados para o Software a cada 5 minutos por meio de rede sem fio.

4.1.1.2 Requisitos não funcionais de Hardware

HW-NF-1) O sistema deve ser acoplável a fermentadores de até 20 Litros disponíveis no mercado.

HW-NF-2) Em caso de perda de conexão com o Software, o sistema deve tentar enviar os dados ainda não enviados a cada ciclo de envio.

HW-NF-3) Caso o sistema tenha uma oscilação no fornecimento de energia, ele deve ser capaz de voltar ao funcionamento normal.

HW-NF-4) Os dados monitorados devem ser armazenados temporariamente, por no mínimo 15 dias, no Hardware.

4.1.2 Sistema Software

4.1.2.1 Requisitos funcionais de Software

SW-F-1) O sistema deve fornecer acesso ao usuário após identificação com usuário e senha

SW-F-2) O sistema deve fornecer as informações instantâneas das fermentações em progresso.

SW-F-3) O sistema deve permitir acesso às informações históricas de fermentações já realizadas.

SW-F-4) O sistema deve permitir o cadastro de receitas. Uma receita é definida por: identificação, nome, estilo e campo livre para observações. O campo livre pode evoluir para um cadastro padronizado dos ingredientes e processos realizados.

SW-F-5) O sistema deve permitir o cadastro de lotes. Um lote é definido por: identificação, receita utilizada, instante de início da fermentação, instante de fim da fermentação, variáveis personalizadas, perfil de controle e observação.

SW-F-6) O sistema deve permitir o cadastro de perfis de controle. Um perfil de controle é definido por: identificação, nome e temperatura alvo, instante (em relação ao início da fermentação).

SW-F-7) O sistema deve permitir o cadastro de variáveis personalizadas. Uma variável personalizada é definida por: identificador, lote correspondente, chave, valor e instante (em relação ao início da fermentação).

SW-F-8) O sistema deve disponibilizar, para cada lote um gráfico com a evolução de cada variável monitorada ao longo do tempo de fermentação.

SW-F-9) Os dados recebidos pelo Hardware devem ser salvos em banco de dados

SW-F-10) Em caso de perda de conexão com o Hardware, o usuário deve ser notificado por e-mail.

SW-F-11) O sistema deve permitir que o usuário realize o download de seus dados em formato de planilha.

SW-F-12) O sistema deve permitir que o usuário registre seus dispositivos.

4.1.2.2 Requisitos não funcionais de Software

SW-NF-1) As informações de cada usuário são, por padrão, particulares de cada usuário e devem seguir padrões de segurança.

SW-NF-2) As informações instantâneas devem estar disponíveis em até 1 minuto após o recebimento dos dados pelo HW.

SW-NF-3) O sistema deve ser desenvolvido na forma de Web-App, e ser responsivo a dispositivos mobile e computadores.

4.2 Projeto de Controle

4.2.1 Modelagem Térmica do Sistema

O primeiro passo para a implementação do controle de temperatura da fermentação é a modelagem térmica do processo. Nota-se que essa modelagem é bastante complexa pois envolve diversos coeficientes térmicos desconhecidos, devido a composição heterogênea do Mosto com Leveduras.

O objetivo do sistema é controlar a temperatura do líquido fermentado (solução de mosto e leveduras). O líquido estará dentro de um fermentador, sem entrada de ar dado que a fermentação é anaeróbica e o ar interfere na qualidade do experimento. Para expulsar o CO_2 gerado pela fermentação e não permitir a entrada de O_2 o fermentador utiliza um Air-Lock, dispositivo que funciona como válvula só permitindo a direção única desse fluxo. O fermentador é embalado em uma manta térmica, com a intenção de evitar a troca térmica com o ambiente.

A fermentação ocorre na maior parte das receitas para cervejas do tipo Ale em temperaturas entre 14°C e 20°C, que são, muitas vezes, inferiores à temperatura do ambiente. Sendo assim, existe a necessidade de criar um sistema que troque calor com o mosto e leveduras mantendo um gradiente de temperatura entre o fermentador e o ambiente constante conforme configuração do usuário.

Para isso acontecer durante o processo, o calor retirado pela refrigeração deve ser o mesmo que o gerado pela convecção com o ambiente e pelo próprio processo de fermentação, que é exotérmico.

É importante destacar que o dispositivo irá ter duas funções: a primeira é manter a temperatura do líquido fermentado; a segunda é levar o líquido até determinada temperatura. Ambas as funcionalidades envolvem a capacidade do dispositivo de retirar ou fornecer calor do sistema de forma eficiente e constante.

4.2.2 Transferência de Calor entre Ar e Fermentador

Nessa modelagem, serão analisados os seguintes elementos:

- Mosto e Leveduras
- Tanque de Fermentação
- Manta Térmica

Nessa dinâmica, considerando que o fermentador vai operar na maior parte do tempo em temperaturas inferiores às do ambiente, o calor vai fazer o seguinte caminho:

1. A manta térmica recebe calor do ambiente através da convecção e radiação do ar.
2. A manta térmica transfere calor por condução para o fermentador.
3. O fermentador transfere calor por condução para o líquido fermentado.
4. O líquido fermentado produz calor pelo processo de fermentação (processo exotérmico).

$$\textit{Ambiente} \longrightarrow \textit{Manta térmica} \longrightarrow \textit{Fermentador} \longrightarrow \textit{Mosto e leveduras}$$

Algumas hipóteses foram adotadas visando simplificar o problema:

1. O gradiente de temperatura no interior do Mosto + Leveduras é desprezível;

2. O coeficiente de troca de calor por convecção entre o Mosto + leveduras e o tanque é elevado o bastante para que não sejam observadas diferenças de temperatura entre esses elementos;
3. O regime é permanente e as propriedades são constantes
4. A condução é unidimensional no plano X
5. A transferência de calor por radiação é desprezível nas superfícies
6. Resistências de contato desprezíveis.

As hipóteses 1, 2, 3 podem ser adotadas devido ao horizonte de tempo da fermentação, no qual é necessário manter a mesma temperatura durante dias. As hipóteses 4, 5 e 6 foram adotadas com a intenção de simplificar o problema. A figura 2 exemplifica o desenho desse problema.

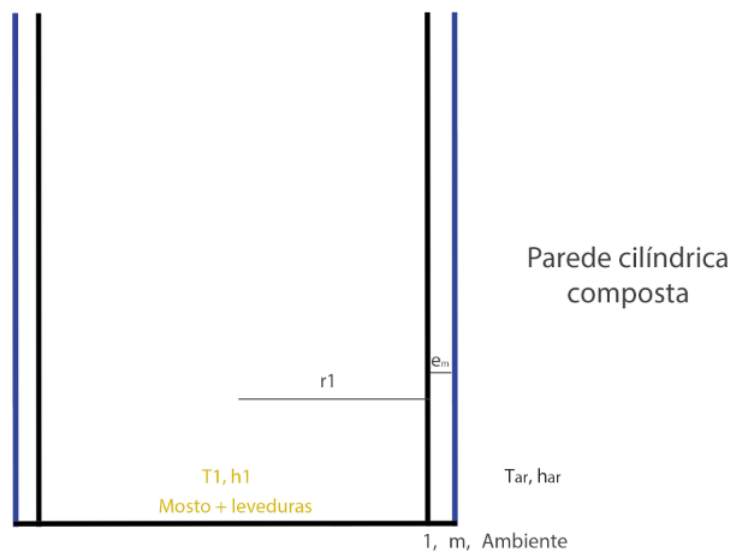


Figura 2: Desenho do problema de troca de calor entre o ambiente e fermentador.
Fonte: Autores.

4.2.3 Simulação Térmica

A Simulação térmica do sistema possui três objetivos:

1. Estimar a potência de troca de calor necessária para manter o sistema com 10°C de diferença em relação à temperatura ambiente;

2. Estimar o tempo morto do sistema para levar o sistema do equilíbrio térmico para uma diferença de 10°C ;
3. Simular um controle PID utilizando uma fonte de calor variável, no caso do sistema real, uma pastilha de Peltier para a sintonização inicial do sistema.

Com esses testes é possível entender se a utilização de pastilhas de Peltier é suficiente para o controle de temperatura.

Foi utilizada a biblioteca Simscape do software Matlab. Essa biblioteca contém blocos que simulam elementos térmicos e funcionam de maneira análoga a um circuito elétrico. No caso, a diferença de temperatura seria equivalente à diferença de potencial elétrico e o calor que circula no sistema seria equivalente a corrente elétrica. As figuras 3 a 7 representam blocos utilizados.



Figura 3: Bloco de fonte de calor, que mantém a temperatura em um ponto do circuito constante.



Figura 4: Bloco de transferência de calor por convecção, seguindo a fórmula $Q = k \cdot A \cdot (T_A - T_B)$



Figura 5: Bloco de transferência de calor por radiação, seguindo a fórmula $Q = k \cdot A \cdot (T_A^4 - T_B^4)$



Figura 6: Bloco de transferência de calor por condução, seguindo a fórmula $Q = k \cdot \frac{A}{D} (T_A - T_B)$

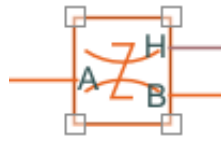


Figura 7: Bloco de sensor de calor ideal

Tabela 1: Descrição dos símbolos de trocas de calor referentes às figuras 4 a 6.

Símbolo	Descrição
Q	Quantidade de Calor
k	Coeficiente de condutividade térmica do material
A	Área normal à transmissão de calor
D	Espessura do material
T_A, T_B	Temperaturas da camada A e B, respectivamente

Com as devidas simplificações justificadas anteriormente, o sistema foi simulado com o circuito da figura 8. A seguir, cada seção de blocos do circuito é descrita em detalhes. A temperatura ambiente adotada na simulação é 25°C e a temperatura do líquido, 15°C .

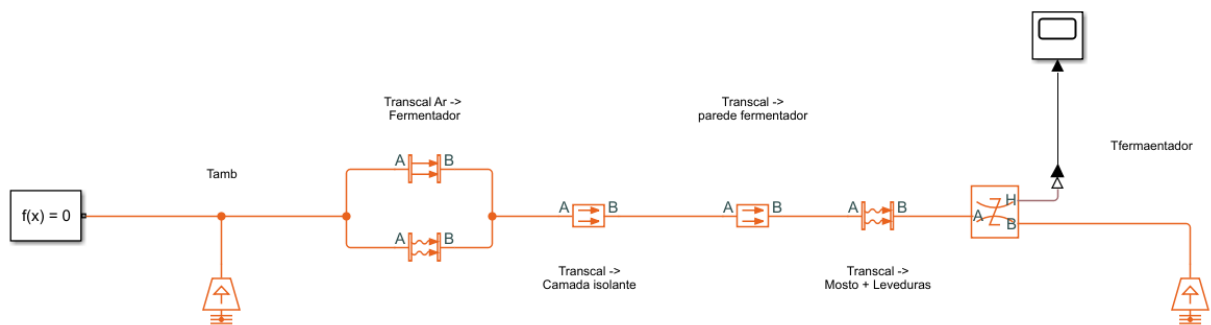


Figura 8: Circuito de blocos utilizado para a simulação térmica.

Fonte: Autores.

Para a simulação, foi considerado um fermentador cilíndrico com 37 cm de altura e 26 cm de raio. Ele tem paredes de polipropileno com coeficiente térmico de $0,25\text{W}/\text{m} \cdot \text{K}$ e 0,1 cm de espessura. Esse fermentador é encoberto por uma camada de isolante de espuma de poliestireno com coeficiente térmico de $0,03\text{W}/\text{m} \cdot \text{K}$ e 0,2 cm de espessura. O líquido é considerado como água, visando simplificar os cálculos.

O conjunto de blocos da figura 9 representa a transferência de calor entre o ar e a camada isolante que circunda o fermentador. Foram consideradas duas formas de transferência, por convecção e radiação. A área de contato adotada foi $2\pi \cdot 0,37 \cdot 0,263m^2$, o coeficiente de radiação, $5,667e^{-8} \cdot 0,7W/m^2 \cdot K^4$, e o coeficiente de convecção, $25W/m^2 \cdot K$ (ar).

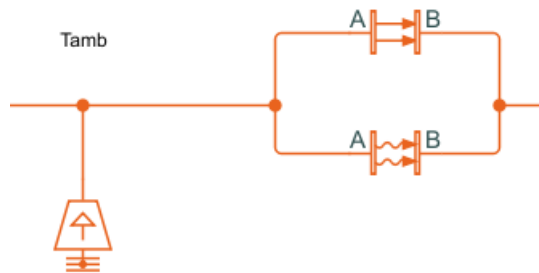


Figura 9: Seção do circuito da figura 8 referente à transferência de calor entre ambiente e isolamento do fermentador.

O bloco da figura 10 representa a transferência de calor por condução entre as paredes do isolante térmico que circunda o fermentador. Foi considerada área de $2\pi \cdot 0,37 \cdot 0,263m^2$, espessura de 0,2 cm e coeficiente térmico de condução, $0.03W/m \cdot K$.



Figura 10: Seção do circuito da figura 8 referente à transferência de calor pelo isolamento do fermentador.

A transferência entre extremidades da parede do fermentador é simbolizada pelo bloco da figura 11. Para esse bloco, foi adotada área de $2\pi \cdot 0,37 \cdot 0,261m^2$, espessura de 0,1 cm e coeficiente térmico de condução, $0.25W/m \cdot K$.



Figura 11: Seção do circuito da figura 8 referente à transferência de calor pelas paredes do fermentador.

Finalmente, a transferência de calor entre a parede do fermentador e do mosto em fermentação está representado pelo bloco de transferência de calor da figura 12. Para a simulação foram considerados os valores de área como $2\pi \cdot 0,37 \cdot 0,261m^2$ e coeficiente $1000 \text{ W} / m^2 \cdot K$ (considerando a referência da água).

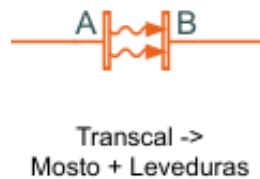


Figura 12: Seção do circuito da figura 8 referente à transferência de calor entre paredes do fermentador e mosto em fermentação.

Executando a simulação, a curva de calor resultante (figura 13) indica que para manter a temperatura sistema $10^\circ C$ abaixo do ambiente, é necessário retirar cerca de 57 W de calor do sistema. Esse valor é compatível com a potência máxima das pastilhas de Peltier disponíveis no mercado, que variam entre 90 e 230 W .

Uma segunda simulação foi realizada para determinar o tempo necessário para levar temperatura do sistema do equilíbrio térmico até uma novo valor. Para isso é ligada uma massa térmica, com o calor específico igual à água de $4184 J/K \cdot Kg$, ocupando de $15L$ do fermentador, e com massa de aproximadamente 15 Kg . O atuador é uma fonte de calor variável, que irá retirar do sistema uma quantidade de calor constante de 57 W . Simulando para intervalo de tempo de 20 horas, foram obtidas as curvas de temperatura (figura 15) e quantidade de calor (figura 16).

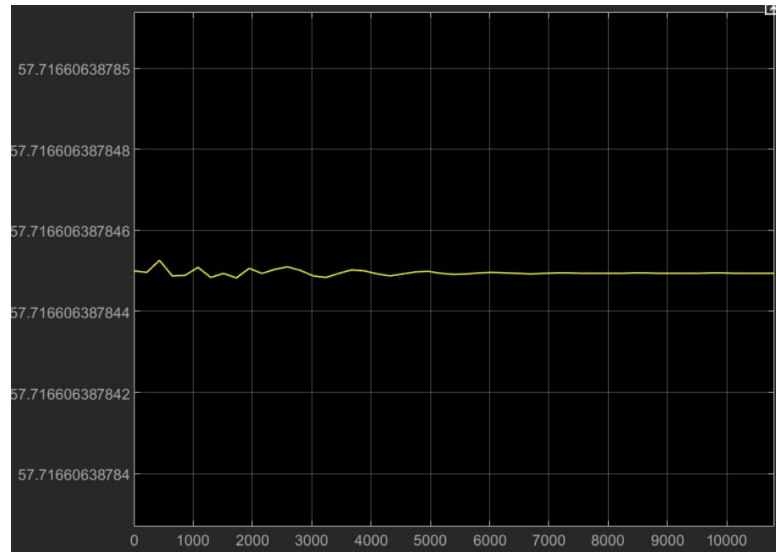


Figura 13: Curva de calor obtida a partir da simulação do circuito da figura 8.

Fonte: Autores.

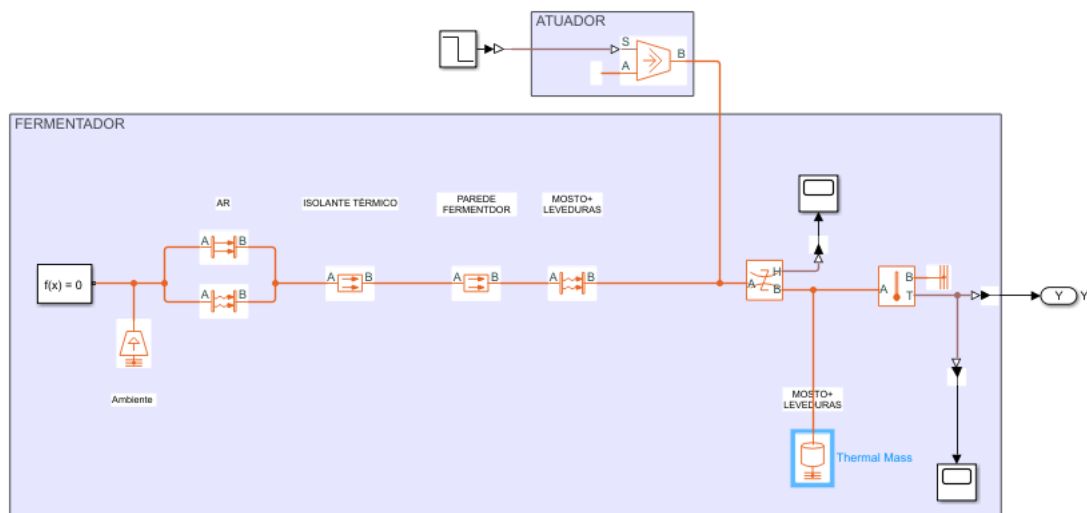


Figura 14: Circuito térmico completo, com atuador e massa térmica.

Fonte: Autores.

Observando o sistema, conclui-se que o tempo necessário para mover o sistema do equilíbrio térmico para uma temperatura alvo é muito alto e isso pode ser prejudicial nas primeiras horas da fermentação. Dessa forma, é interessante iniciar o processo com a primeira temperatura alvo, o que já é de certa forma realizado na prática, pois a temperatura de início da fermentação deve ser atingida antes do inoculação da levedura e fechamento do fermentador.

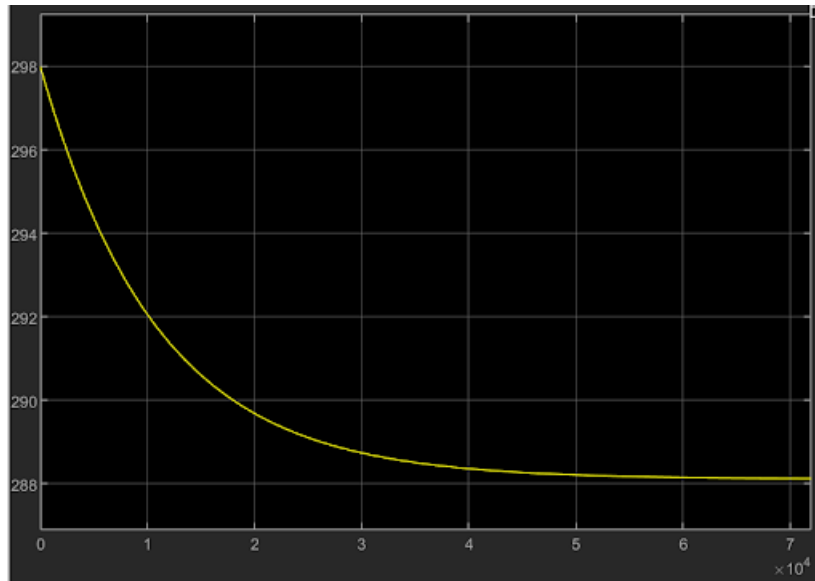


Figura 15: Curva de temperatura em Kelvin por tempo em dezenas de milhares de segundos, obtida da simulação do circuito térmico da figura 8.

Fonte: Autores.

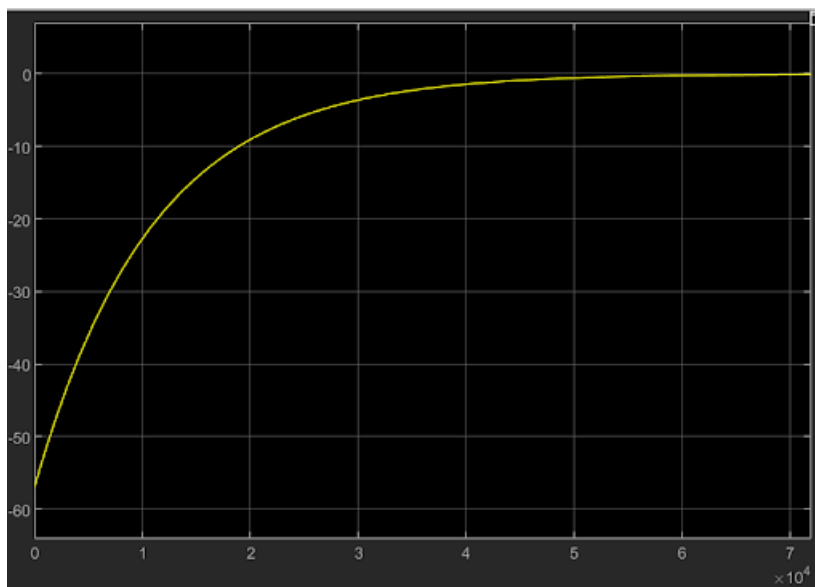


Figura 16: Curva de quantidade de calor em Watts por tempo em dezenas de milhares de segundos, obtida da simulação do circuito térmico da figura 8.

Fonte: Autores.

4.2.4 Modelo para Sintonização do Controlador

Foi construído o modelo da figura 17 no Software Matlab para auxiliar na sintonização do controlador. Foram utilizados o fermentador anteriormente modelado, uma fonte de calor variável para representar o nosso atuador, um bloco PI (inicialmente considerando o ganho derivado como nulo) e uma referência de temperatura.

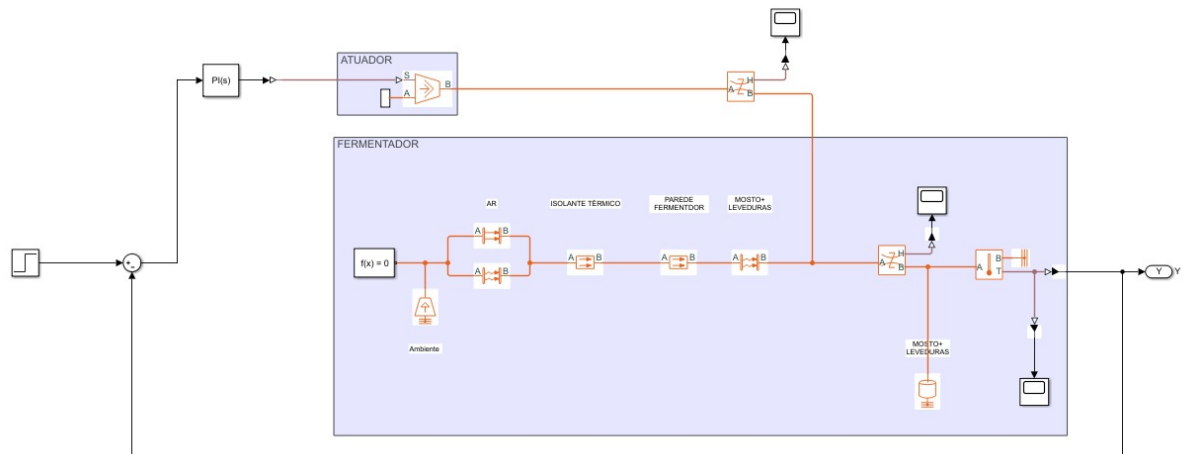


Figura 17: Circuito utilizado para sintonização do controlador.

Fonte: Autores.

Com o auxílio do sintonizador do Matlab, chegou-se a um valor de $K_p = 6,05$ e de $K_i = 0,0010245$; onde $K_i = K_p/T_i$. Esse valor se mostrou capaz de gerar uma resposta rápida e pouco oscilante, mas melhorias empíricas podem ser obtidas no futuro. A figura 18 ilustra a curva de resposta com uma referência de 290K de temperatura e uma temperatura ambiente de 300K, e a figura 19, o calor fornecido pelo atuador ao longo do tempo. Conclui-se que o sistema consegue atingir a referência, porém possui resposta inicial demorada.

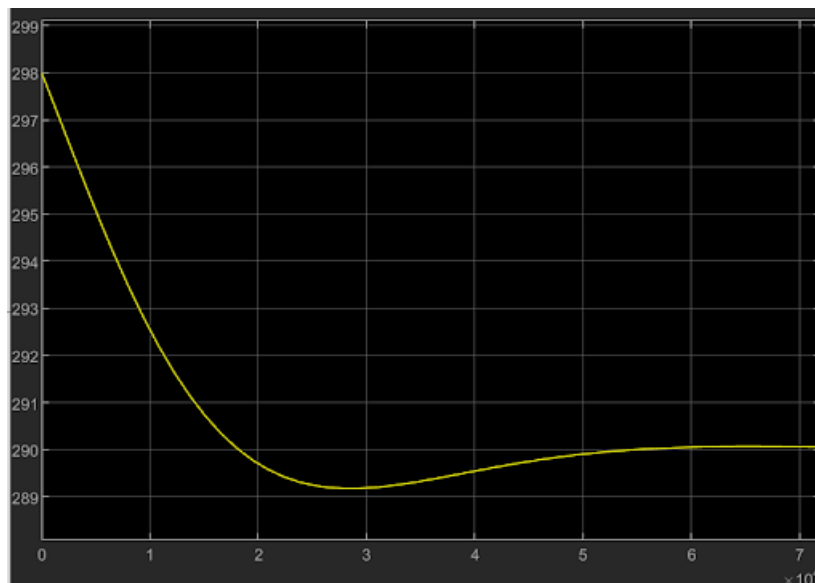


Figura 18: Curva de temperatura em Kelvin por tempo em dezenas de milhares de segundos, obtida da simulação de sintonização do controlador.

Fonte: Autores.

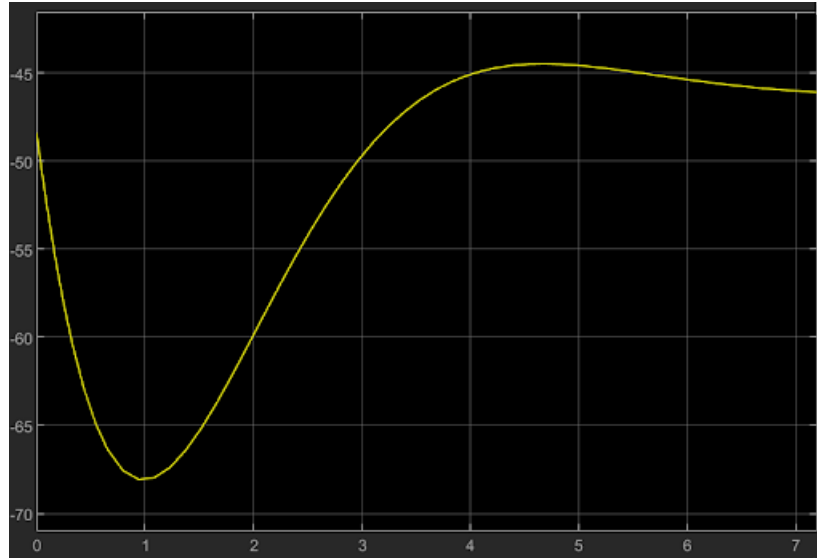


Figura 19: Curva de calor em Watts fornecido pelo atuador por tempo em dezenas de milhares de segundos, obtida da simulação de sintonização do controlador.

Fonte: Autores.

4.2.5 Dispositivo para Troca de Calor

Um dos maiores desafios do projeto é criar um sistema que consiga trocar calor de forma eficiente, não seja intrusivo e consiga ser simples o suficiente para ser utilizado por um hobbysta. Conforme anteriormente especificado, a ideia é utilizar placas de peltier para realizar essa troca. A maior vantagem do uso dessas placas é a possibilidade de controlar o calor associado proporcionalmente a quantidade de corrente fornecida ao módulo, através da seguinte equação 4.1. Possibilitando assim o uso de um circuito junto com o microcontrolador para o controle de temperatura.

$$Q_P = \pi \cdot 1 \quad (4.1)$$

4.2.6 Controle de Temperatura

Um controlador de malha fechada proporcional iterativo derivativo (PID) é utilizado para controlar a temperatura do fermentador. Esse método é amplamente utilizado na indústria, possuindo boa precisão e confiabilidade, além de ser facilmente sintonizado. Inicialmente são definidos parâmetros analógicos e depois é criado o controle digital.

4.2.6.1 Controlador PID

O controlador PID utiliza 3 ações (proporcional, integrativa e derivativa) controlar a planta minimizando o erro. Sua saída pode ser definida pela equação 4.2. A figura 20 esquematiza a malha de controle.

$$u(t) = K_p \cdot (e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_d \frac{de(t)}{dt}) \quad (4.2)$$

Na qual:

Tabela 2: Descrição das variáveis do controlador PID da Equação 4.2.

Símbolo	Descrição
K_p	ganho
T_i	constante integrativa
T_d	constante derivativo
e	erro
t	tempo
τ	tempo de integração

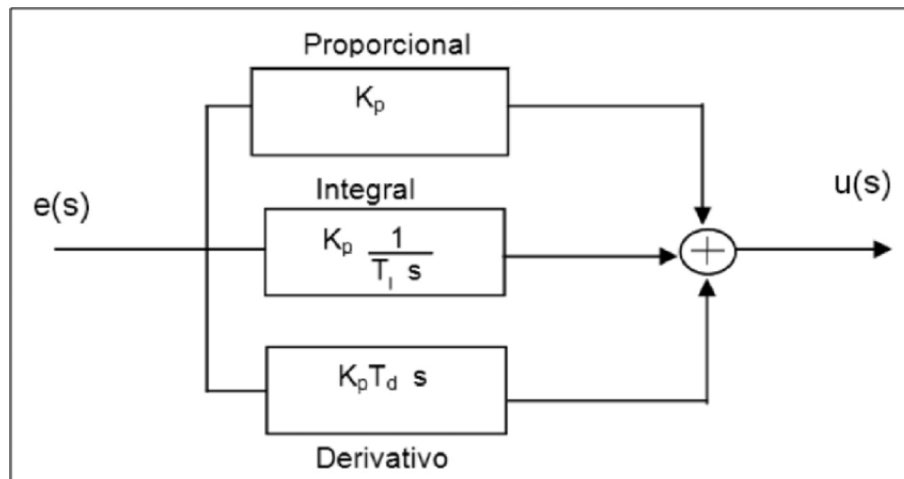


Figura 20: Malha de controle PID.

A ação proporcional do controlador produz um sinal de saída proporcional à amplitude do erro $e(t)$. Essa ação é útil para criar uma resposta equivalente ao tamanho do erro do sistema. A ação integral produz um sinal de saída proporcional à magnitude do erro, dependendo não só do seu valor, mas também da sua duração. Essa ação corrige o erro de off-set gerado pela ação proporcional e acelera a resposta do sistema. Por fim, a ação

derivativa produz um sinal de saída proporcional à velocidade de variação do erro. Essa ação melhora a estabilidade e velocidade de resposta do sistema através de uma correção antecipada do erro.

4.2.6.2 Método de Ziegler-Nichols

Para escolha dos parâmetros de ganho, o método de Ziegler-Nichols foi utilizado. Esse método foi escolhido principalmente pela sua simplicidade e facilidade na sintonização, visto que não é necessário uma modelagem matemática precisa da planta para o seu ajuste. A escolha dos parâmetros é feita a partir da observação da resposta do sistema em malha aberta com a aplicação de um degrau. A figura 21 ilustra a curva de resposta do sistema, e a tabela 3 lista os parâmetros provenientes do método.

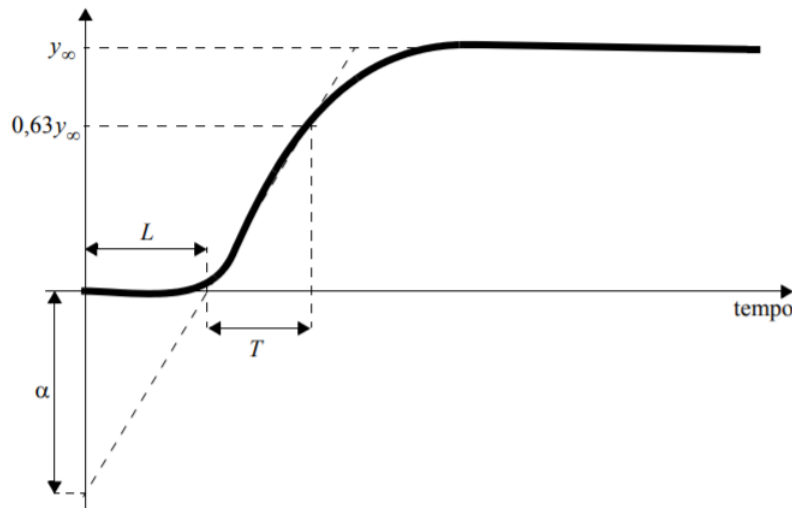


Figura 21: Resposta do Sistema em malha aberta a sinal degrau.

Tabela 3: Parâmetros do controlador PID seguindo método de Ziegler-Nichols.

Controlador	K_p	T_I	T_D
P	$1/\alpha$		
$P + I$	$0,9/\alpha$	$3L$	
$P + I + D$	$1,2/\alpha$	$2L$	$0,5L$

4.2.6.3 Anti Wind-up

Nesta aplicação, devido a limitações do atuador, é esperado uma saturação na saída. No caso onde o sistema não consegue chegar no setpoint, teremos uma situação chamada

de wind-up, na qual, a sua resposta integral irá crescer de forma indefinida. Para evitar problemas no controlador, é necessário a implementação de um sistema anti wind-up, no caso, a ação escolhida é congelar a ação do controlador em caso de saturação.

4.3 Projeto de Hardware

Com base nos requisitos, foram determinados os principais componentes do projeto para o controle e monitoramento do sistema

4.3.1 Microcontrolador

O microcontrolador Wemos D1 (figura 22) é versátil, muito semelhante a microcontroladores Arduino, e ideal para a prototipagem e foi escolhido para prototipagem do projeto. Ele possui portas de entrada e saída analógicas e digitais, fontes de alimentação de 3,3V a 5V e é programado na linguagem C. O circuito da placa foi desenvolvido tendo como base o microchip ESP8266 que oferece conectividade *Wi-Fi*, diferente dos modelos mais simples de Arduino.

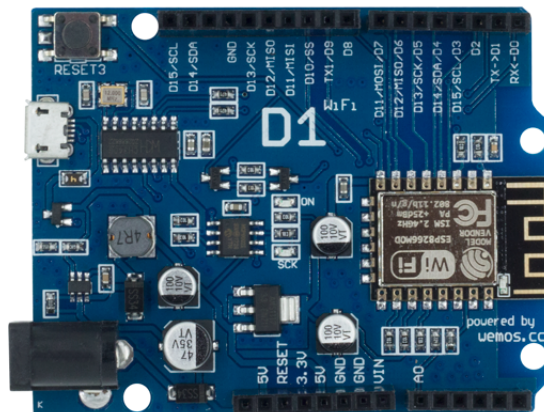


Figura 22: Wemos D1

Fonte: <http://www.esp8266learning.com/>.

4.3.2 Critérios de Escolha dos Sensores

A definição dos sensores a serem utilizados se baseou nos seguintes critérios, em ordem de importância:

- Faixa de operação compatível com os requisitos e especificidades do sistema
- Disponibilidade no mercado brasileiro
- Compatibilidade com o Microcontrolador
- Custo de aquisição

O primeiro critério é trivial, e deve ser eliminatório em qualquer avaliação. O critério de disponibilidade no mercado brasileiro teve essa classificação pelo desejo de se adquirir e trabalhar com os sensores o mais rápido possível e mitigar o risco com atrasos devido a importações e falta de estoque. A compatibilidade com o Wemos D1 exprime quão diretamente um sensor pode ser utilizado, principalmente em relação a alimentação, sendo preferíveis sensores que tenham tensões de entrada em comum com o microcontrolador, i.e., 3,3 ou 5V. Finalmente, custos de aquisição menores são preferíveis.

4.3.3 Sensor de Temperatura

Para a medição de temperatura o sensor escolhido foi o circuito integrado DS18B20 (Figura 23) com uma vedação impermeável. É um sensor que atende as especificações e é amplamente utilizado em conjunto com o Wemos D1, Arduino e outros microcontroladores semelhantes, e possui alta disponibilidade no mercado a um baixo custo, sendo comercializado em uma versão impermeável, muita prática para esse projeto. A leitura da temperatura pelo microcontrolador é realizada por apenas um fio de dados, utilizando a interface One-Wire.

Ficha técnica:

- Tensão de operação: 3-5,5V
- Faixa de medição: -55°C a $+125^{\circ}\text{C}$
- Precisão: $\pm 0.5^{\circ}\text{C}$ entre -10°C e $+85^{\circ}\text{C}$
- Ponta de aço inoxidável

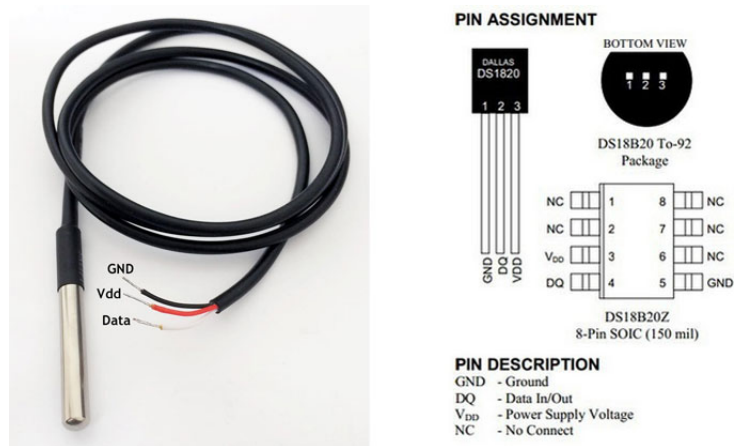


Figura 23: Sensor DS18B20

Fonte: <https://escapequotes.net/>.

4.3.4 Sensor de Densidade Relativa

A medição da densidade relativa é menos trivial pela ausência de sensores completos comercializados a baixo custo. Quando não é automatizada, a medição é comumente realizada com auxílio de um densímetro ou um refratômetro, que medem a densidade relativa e a concentração de açúcar dissolvido em graus Brix, respectivamente. Ambas as ferramentas requerem interação humana e a extração de uma pequena amostra da solução, aproximadamente 100 a 250ml e algumas gotas, respectivamente.

Quanto a medição automática e contínua da densidade relativa, Boulton e Quain [Boulton e Quain 2001] apresentam algumas formas de medir a grandeza, destacam-se: a utilização de sensores de pressão posicionados em diferentes alturas do fermentador, e computando a diferença de pressões medidas; e utilizando um sensor ultrassônico, medindo o tempo que um pulso leva para ser transmitido entre dois pontos fixos, entremeado pela solução.

Em pesquisa por soluções existentes no mercado, a abordagem de dois produtos são dignas de consideração: o Beer Bug utiliza uma célula de carga para medir o empuxo sofrido por um peso submerso na solução, e o Plaato, que utiliza a medição do gás carbônico expelido durante a fermentação para calcular indiretamente a densidade relativa.

Dentre as opções listadas, a medição por diferença de pressões e empuxo foram selecionadas para serem adotadas em primeiro momento no projeto, com prioridade da primeira. Essas são as soluções que aparentam apresentar menor complexidade na medição e maior facilidade na calibração para obtenção de resultados consistentes.

As opções listadas são discutidas a seguir, contemplando as especificações necessárias

para cada sensor, e os modelos escolhidos para utilização no projeto.

4.3.4.1 Medição por diferença de pressão

Esse método se baseia no conceito de pressão estática $P_{estatica}$, que é pode ser calculada para um determinado ponto em um fluido como o produto da densidade ρ desse fluido, da aceleração gravitacional g , e da altura h da coluna de líquido sobre o ponto escolhido.

$$P_{estatica} = \rho \cdot g \cdot h \quad (4.3)$$

Consequentemente, considerando a densidade homogênea e a variação da aceleração gravitacional desprezível, a diferença de pressão entre dois pontos em diferentes alturas desse fluido, é:

$$\Delta P = \rho \cdot g \cdot \Delta h \quad (4.4)$$

Dessa forma, mantendo a diferença de alturas Δh fixa e conhecida, é simples inferir o valor da densidade a partir da medida da diferença de pressão medida pelo sensor.

Seguindo a especificação do requisito HW-F-3, o sistema deve ser capaz de medir densidades relativas de 1,000 e 1,150, em relação à água a 20 °C, o que corresponde a uma faixa entre 998,203 e 1147,933 kg/m³. Considerando a aceleração gravitacional 9.807 m/s² e uma distância de 20 cm entre os dois pontos medidos (o que é razoável para fermentadores pequenos, até 50L), a faixa de operação do sensor de diferencial de pressão deve ser, em diferentes unidades comerciais:

Tabela 4: Diferenças de pressão esperadas para cálculo da densidade relativa durante a fermentação.

	N/m ² ou Pa	bar	psi
Valor Mínimo	1957,805	0,019578	0,284
Valor Máximo	2251,476	0,022515	0,327

Para obter a precisão de 1 milésimo de densidade relativa, o sensor necessita de uma precisão de aproximadamente 0,1%.

A partir das especificações, foram buscados os sensores disponíveis no mercado, principalmente os produzidos pela Mouser Electronics (<https://br.mouser.com/>). Seguindo os

critérios definidos, o modelo escolhido foi o MP3V5010DP, que é oferecido no chio Pmod DPG1 pela Diligent (Figura 24), que possui as seguintes especificações:

- Faixa de medição: 0 a 10 kPa
- Acurácia: 5%
- Tensão de alimentação: 3.3V

Apesar do sensor não apresentar uma acurácia adequada, acredita-se que o uso da média de diversas medidas e calibração com algumas medidas feitas pelo usuário podem propiciar resultados satisfatórios.

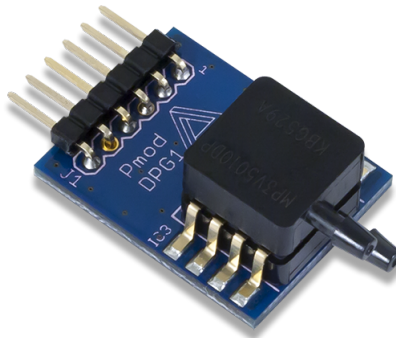


Figura 24: Sensor de diferença de pressão MP3V5010DP acoplado ao chip Pmod DPG1
Fonte: <https://store.digilentinc.com/>.

4.3.4.2 Medição por Empuxo

O empuxo hidrostático E sofrido por um corpo submerso em um fluido é expresso pelo produto entre a densidade ρ_f do fluido, o volume submerso V_f do corpo, e da aceleração da gravidade g :

$$E = \rho_f \cdot V_f \cdot g \quad (4.5)$$

Naturalmente, o corpo também sofre ação da força gravitacional P , que é definida pelo produto da massa m do corpo pela aceleração gravitacional g :

$$P = mg \quad (4.6)$$

As duas forças atuam em sentidos opostos (como ilustrado na figura a seguir), de forma que, utilizando uma célula de carga conectada por meio de um fio a um corpo, é possível notar uma diferença no “peso natural” do objeto submerso, que é igual a força resultante entre a força gravitacional e o empuxo. Esse é o conceito de uma balança hidrostática.

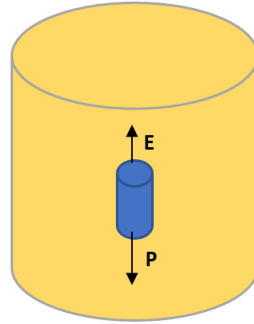


Figura 25: Representação de um corpo (azul) submerso em um fluido (amarelo) e das forças gravitacional e de empuxo

Fonte: Autores.

Considerando um corpo de massa 250g e volume de 125cm^3 arbitrários, e a variação de densidade especificada, uma célula de carga adequada para o projeto deve ser capaz de medir massas entre 112,75 e 125,22g, com precisão mínima de aproximadamente 0,125g. Nota-se que é desejável um corpo mais denso que o fluido para que ele fique completamente submerso, facilitando aplicação da relação do empuxo. O sensor ZHIPU-200g (Figura 26) foi escolhido, medindo entre 0 a 200g, com precisão de 0,1g.



Figura 26: Sensor de carga ZHIPU-200g

Fonte: <https://mercadolivre.com.br/>.

4.3.5 Sensor de pH

Para a medição de pH o sensor escolhido foi o E-201-C (Figura 27) com uma placa condicionadora que permite interface direta com o microcontrolador. Assim como o DS18B20,

é um sensor amplamente utilizado em conjunto com o Arduino e facilmente encontrado no mercado nacional. A comunicação com o microcontrolador é realizada por comunicação serial, utilizando os protocolos UART ou I2C.

Ficha técnica:

- Tensão de operação: 5V;
- Temperatura de operação: 0°C a 60°C;
- Tempo de resposta: 5s;
- Tempo de sedimentação: 60s;
- Faixa de medição: pH de 0 a 14



Figura 27: Sensor de pH E-201-C

Fonte: <https://www.crcibernetica.com/>.

4.3.6 Atuador de Temperatura

Para a montagem do nosso dispositivo que realiza trocas de calor com o fermentador, as células termoelétricas de Peltier (figura 28) se mostraram uma boa opção. Elas são versáteis, possuem a capacidade de esfriar ou aquecer o fermentador dependendo do sentido da corrente aplicada em seus terminais e fornecem calor proporcionalmente a corrente fornecida ao sistema.

A célula é constituída por duas chapas de material isolante com um material condutor entre elas, como é esquematizado na figura 29. Quando uma diferença de tensão é aplicada entre os terminais da célula, o movimento dos semicondutores de tipo “n” e “p” transforma a energia elétrica em energia térmica, criando um fluxo de calor que aquece uma célula e esfria outra. Em um semicondutor do tipo-n, o calor é absorvido próximo ao terminal

negativo e rejeitado próximo ao terminal positivo, já em um semicondutor do tipo-p o processo se dá de maneira inversa. Como os pares tipo “n” e “p” tem características diferentes, é possível alterar o fluxo de calor dependendo do sentido da corrente.



Figura 28: Célula de Peltier modelo TEC1-12706.

Fonte: <https://mercadolivre.com.br/>.



Figura 29: Esquema de funcionamento de uma célula de Peltier.

Fonte: <https://pt.wikipedia.org/>.

Um exemplo de aplicação comercial das células de Peltier para controle da temperatura de fermentação, pode ser observada no produto desenvolvido pela empresa Brew Jacket. Inspirado nessa solução, o sistema de troca de calor é composto por um dissipador de calor e uma ventoinha na face quente (figura 30), e uma haste de aço inoxidável (figura 31) ligado no lado frio, o servirá como condutor de calor entre a o fermentador e o módulo de Peltier. O aço inoxidável não conduz calor tão bem quanto o cobre ou o alumínio, mas devido resistência a corrosão em meio ácido, é a única opção segura das três.



Figura 30: Dissipador de calor e ventoinha, modelo utilizado em CPU.

Fonte: <https://www.hexus.net/>.



Figura 31: Haste de Aço inoxidável.

Fonte: <https://mercadolivre.com.br/>.

Uma alternativa, ainda se utilizando a célula de Peltier, é realizar a troca de calor com a solução pela parede do fermentador, de forma menos invasiva e com menores riscos de contaminação, mas gerando uma dependência maior à geometria e material do fermentador. Uma alternativa seria utilizar a células em contato próximo à parede do fermentador; e a construção de um pequeno sistema que utilize um fluido circulando por serpentinas em torno ao fermentador, com as células atuando a temperatura do fluido.

4.4 Projeto de Software

A partir dos Requisitos Técnicos levantados, foi realizado o projeto de software. O projeto se iniciou com a definição de casos de uso a serem implementados, de forma a satisfazer os requisitos técnicos, em conjunto o diagrama de casos de uso definido pela UML foi elaborado para prover apresentação visual. Em seguida, as informações que devem ser armazenadas pelo sistema foram levantadas e a relação entre elas foi estabelecida, sendo representada no diagrama de entidade-relacionamento. O passo seguinte foi a definição da arquitetura de software do sistema.

Foram seguidos os padrões da UML apresentados por Bezerra 2015, tentando-se atentar às convenções definidas, contudo, alguns recursos gráficos como utilização de cores foram utilizadas com a finalidade de tornar o diagrama mais informativo.

No projeto a arquitetura de microsserviços foi escolhida como padrão para o sistema. Essa arquitetura define padrões de modularização do sistema em componentes pequenos e altamente especializados, conferindo facilidades de manutenção e escalabilidade em contraste com a arquitetura de monólito. Em contrapartida, o sistema é mais complexo de se implementar e implantar, devido a separação dos módulos, contudo, considerando o desejo de continuar este projeto após a entrega e os padrões de mercado atuais a abordagem de microsserviços é considerada a mais adequada.

Definida a arquitetura, cada módulo do sistema foi especificado e foi elaborado o diagrama de componentes, definido pela UML, para ilustrar os pontos de comunicação e componentização da solução completa. Nessa etapa foram definidas as tecnologias a serem empregadas em cada módulo, a justificativa das escolhas é apresentada após o detalhamento dos componentes da arquitetura de software. O projeto de implantação foi então realizado, com planejamento da disponibilização do sistema de software com ferramentas de computação de nuvem disponíveis no mercado.

4.4.1 Especificação dos Casos de Uso

Na definição dos casos de uso do projeto foram definidos dois atores que interagem com o sistema de software a ser desenvolvido, identificados como usuário e dispositivo. O usuário representa o utilizador humano do sistema a ser desenvolvido, responsável por todas as interações humanas necessárias.

O usuário se comunica com o sistema por duas interfaces, uma aplicação web, que se comunica diretamente com o sistema, e um aplicativo para smartphone, necessário para configurações iniciais do dispositivo. O dispositivo representa o sistema hardware de controle e monitoramento, também desenvolvido neste projeto. Em relação ao sistema de software, ele é tratado como um ator com suas devidas interações; seu projeto e especificações são discutidos na seção destinado ao projeto de hardware.

Segue a especificação dos casos de uso em si, contendo a identificação de cada caso de uso, sua breve descrição, enumeração dos passos que o definem e listagem dos requisitos técnicos relacionados ao caso de uso. Com caráter ilustrativo, o Diagrama de Casos de Uso da UML é apresentado na figura 32.

UC - 1: Registro de Dispositivo

Descrição: ao obter um novo dispositivo, o usuário deve configurar seu acesso à rede Wi-Fi e registrá-lo, de modo que o sistema reconheça que aquele dispositivo pertence ao usuário.

1. Usuário acessa aplicativo em seu smartphone
2. Sistema autentica acesso do usuário
3. Aplicativo se conecta ao dispositivo
4. Usuário informa configurações da rede Wi-Fi
5. Aplicativo envia informações da rede para dispositivo
6. Dispositivo se conecta na rede e se prepara para receber mensagens do sistema
7. Aplicativo envia informações do dispositivo para o sistema
8. Sistema cadastra informações do dispositivo e usuário

Requisito relacionado: SW-F-12

UC - 2: Cadastro de Receitas

Descrição: fluxo de cadastro de receitas.

1. Usuário acessa tela de listagem de receitas
2. Sistema exibe todas as receitas referentes ao usuário
3. Usuário seleciona opção “Criar Receita” e acessa tela de cadastro de receita
4. Usuário informa nome, estilo e observações da receita e clica em “Salvar”
5. Sistema cadastra a receita no banco de dados

Requisito relacionado: SW-F-4

UC - 3: Cadastro de Lotes

Descrição: fluxo de cadastro de lotes, perfis de controle e associação de lote a um dispositivo.

1. Usuário acessa tela de listagens de receitas
2. Sistema exibe todas as receitas referentes ao usuário
3. Usuário escolhe uma receita e seleciona a opção “Criar Lote”, e acessa a tela de cadastro de lote
4. Sistema carrega listagem de perfis de controle já cadastrados e dispositivos do usuário
5. Usuário informa identificação e observações do lote
6. Usuário escolhe um perfil de controle já existente ou cria um novo perfil, informando uma identificação e cada um dos passos de controle (instante e valor alvo de temperatura)
7. Usuário seleciona qual dispositivo irá controlar a produção do lote
8. Usuário clica em “Salvar”
9. Sistema cadastra lote, perfil de controle (caso novo), associação de lote e perfil de controle, e associação de lote e dispositivo
10. Sistema envia informações do lote para dispositivo selecionado

Requisitos relacionados: SW-F-5 e SW-F-6.

UC - 4: Envio das informações do Lote para Dispositivo

Descrição: fluxo de envio das informações do lote para o dispositivo associado ao controle daquele lote.

1. Dispositivo recebe informações do lote que foi associado por tópico de mensagens
2. Dispositivo salva informações localmente
3. Quando pronto, dispositivo inicia rotina de monitoramento e controle

Requisito relacionado: HW-F-5

UC - 5: Envio das informações do Dispositivo para o Sistema

Descrição: fluxo de envio das informações obtidas pelo monitoramento do processo pelo dispositivo para o sistema.

1. Dispositivo envia dados coletados para o sistema
2. Sistema processa dados e salva informações no banco de dados

Requisitos relacionados: HW-F-6, SW-F-9

UC - 6: Visualização das Informações dos Lotes

Descrição: fluxo para visualização das informações gerais e de evolução dos lotes correntes e passados

1. Usuário acessa tela de listagem dos lotes
2. Sistema exibe todas os lotes referentes ao usuário
3. Usuário escolhe um lote e seleciona a opção “Ver Informações”, acessando a tela de informações do lote
4. Sistema exibe informações gerais sobre o lote, como identificação, receita, observações, status, data de início, data de término, densidades relativas inicial e final/atual, estimativa de teor alcoólico, pH final/atual, temperatura final/atual.
5. Sistema exibe um gráfico com as variáveis monitoradas em relação ao tempo
6. Caso usuário clique em “Baixar Dados”, sistema efetua o download dos dados do gráfico em arquivo de texto

Requisitos relacionados: SW-F-2, SW-F-3, SW-F-8, SW-F-11

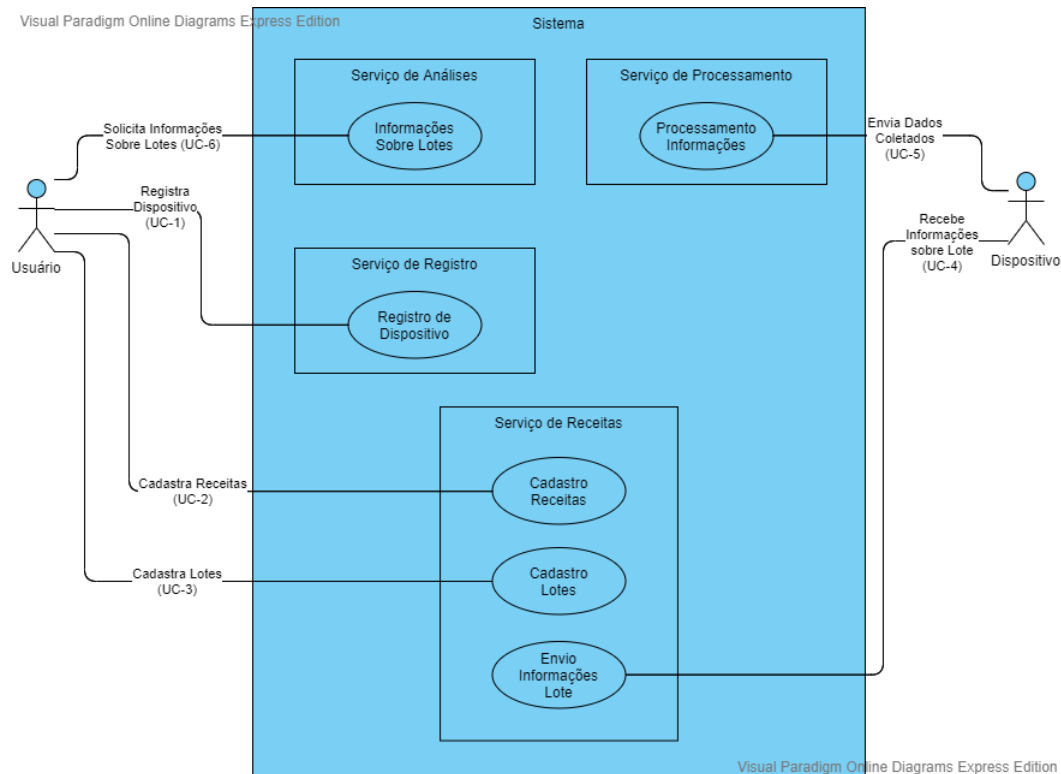


Figura 32: Diagrama de Casos de Uso.

Fonte: Autores.

4.4.2 Modelo de Entidade-Relacionamento

O Modelo Entidade Relacionamento descreve como as informações são organizada no sistema e como o banco de dados do sistema é estruturado. Cada entidade no modelo representa um tipo de informação que é produzida e consumida pelo sistema, na execução dos casos de uso.

Todas as informações são armazenadas em um banco de dados relacional, dividido em schemas que atuam como partições de dados em diferentes domínios. Foram definidos três domínios, traduzidos em schemas no banco de dados, para os dados: user, recipe e control. O schema user contém as informações referentes aos usuários e dispositivos do sistema; o domínio recipe contempla as entidades relacionadas às receitas e lotes de produção, assim como os dados coletados em cada execução de uma receita; e o schema control, por fim, os perfis de controle que são seguidos pelo dispositivo durante seu funcionamento. A separação das entidades em domínios é importante na arquitetura de microsserviços para assegurar que cada serviço tenha controle apenas às informações de sua competência.

A definição das entidades e seus relacionamentos é ilustrada pelo Diagrama Entidade Relacionamento da figura 33, com destaque em cor para cada um dos schemas determi-

nados.

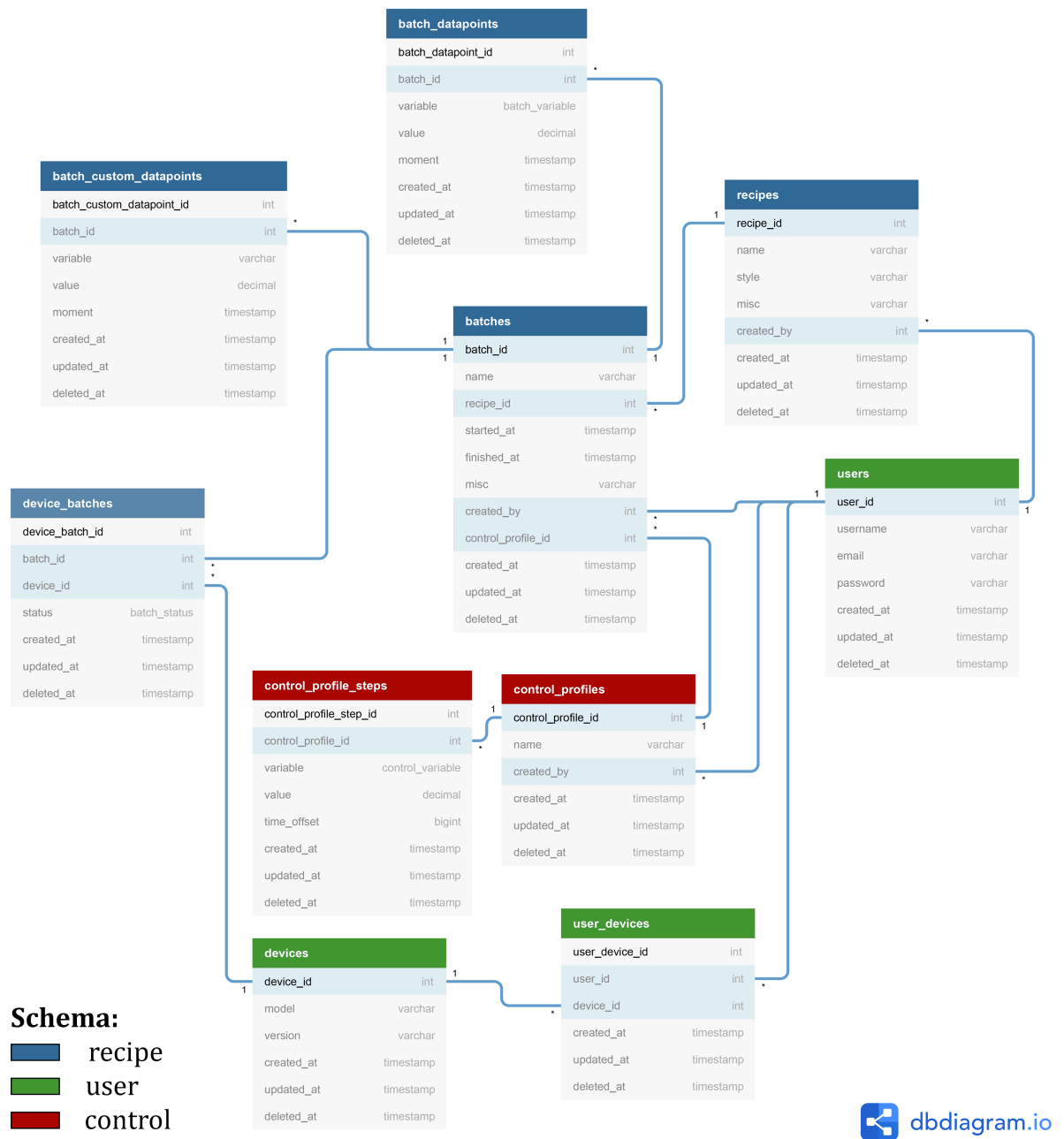


Figura 33: Diagrama de Entidade-Relacionamento.

Fonte: Autores.

4.4.3 Arquitetura de Software

A arquitetura de software do projeto foi estruturada tendo como base o fluxo da informação como especificado nos casos de uso e os conceitos de arquitetura de micro-serviços. Dessa forma, foram definidas quatro camadas para organizar os sistemas a

serem implementados: Camada de Interface, contemplando as interfaces utilizadas diretamente pelos atores; Camada Intermediária, contendo um API Gateway para isolar os microserviços das interfaces e um message broker para intermediar a comunicação entre dispositivo e sistema; Camada de Negócio, contendo os microserviços que exercem as regras de negócio do sistema; e Camada de Persistência, com os serviços de armazenamento de dados. Os componentes de cada camada são descritos em detalhes quanto a suas responsabilidades e detalhes de implementação em sequência. A arquitetura completa é representada visualmente pelo Diagrama de Componentes da figura 40.

4.4.3.1 Camada de Interface

A Camada de Interface (figura 34) contém os componentes que fornecem interface direta aos atores definidos na modelagem de casos de uso. Sua função, portanto, reside na interação com os atores e comunicação de suas ações para a próxima camada, a Camada Intermediária. São três componentes de interface que compõe essa camada: Aplicação Front-end, Aplicação Mobile e o Software Embarcado no dispositivo.

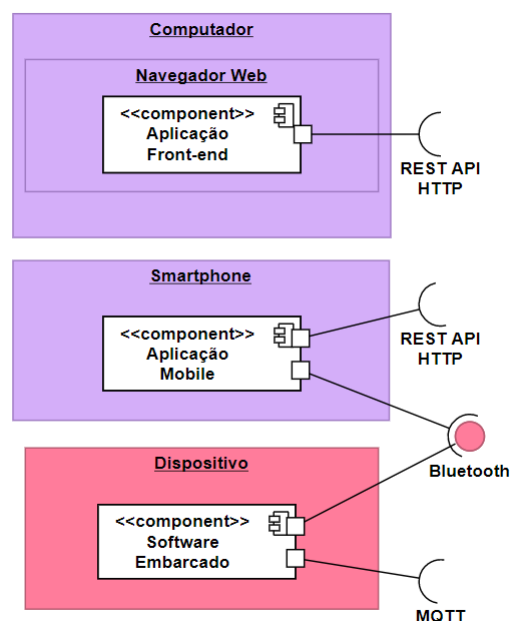


Figura 34: Camada de Interface.

Fonte: Autores.

A Aplicação Front-end é executada pelo navegador web do usuário do sistema, e permite que ele realize o cadastro e gerenciamento de todas as informações referentes ao sistema, como receitas, perfis de controle e lotes, ela também permite que o usuário visualize os dados gerados pelo dispositivo em formato de gráficos e tabelas. Essa aplicação

contém o componente gráfico das telas que o usuário utilizará e algumas regras de validação simples, como campos obrigatórios de cadastro. As ações do usuário são traduzidas em requisições HTTP que são enviadas para o Serviço de API Gateway, na próxima camada.

A Aplicação Mobile foi a solução encontrada para intermediar a comunicação entre usuário e dispositivo, suas principais funções são: enviar as informações necessárias para conexão à rede Wi-Fi do usuário, que é realizado pela tecnologia de bluetooth, e informar o sistema que aquele dispositivo foi ativado por determinado usuário. Após essa configuração, o usuário poderá associar um lote de uma receita para o dispositivo monitorar e controlar, e o dispositivo deve estar pronto para se comunicar com o sistema por meio de mensagens.

O Software Embarcado no dispositivo, além das funcionalidades de monitoramento e controle, que são detalhadas nas seções de Modelagem de Controle e Projeto de Hardware, deve ser capaz de se comunicar via tecnologia bluetooth com a Aplicação Mobile, e enviar e receber mensagens via protocolo MQTT com o Message Broker da Camada Intermediária. O dispositivo deve enviar mensagens com os dados que estão sendo coletados durante a fermentação, como temperatura, densidade relativa e pH, e receber informações sobre um novo lote que foi alocado para ele realizar o controle.

4.4.3.2 Camada Intermediária

Toda a comunicação entre os atores e o sistema é realizada por mediação da Camada Intermediária (figura 35). O tráfego pode ocorrer pelos protocolos HTTP ou MQTT. O fluxo por HTTP é mediado pelo Serviço de API Gateway, que roteia as requisições externas para os devidos microserviços, enquanto que o fluxo MQTT, composto por mensagens, é controlado pelo Message Broker, que organiza as mensagens nos fluxos dispositivo para sistema e sistema para dispositivo.

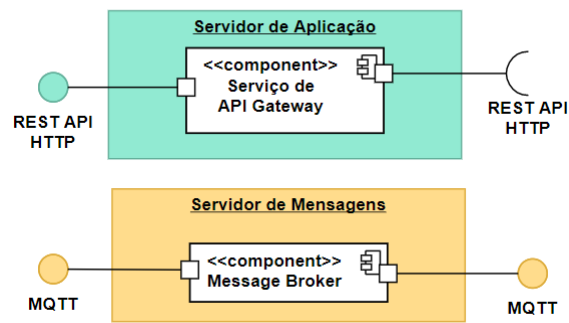


Figura 35: Camada Intermediária.

Fonte: Autores.

O Serviço de API Gateway funciona como um serviço de proxy reverso que, além centralizar as solicitações do usuário, consolida o resultado de cada microserviço necessário para atender uma demanda. Apesar de ser um serviço muito simples, ele é importante por questões de segurança, uma vez que apenas um endereço fica exposto externamente e a autenticação externa é centralizada nele, e também por questões de escala futura, sendo mais fácil implantar um balanceador de carga ou sistema de fila de processamento com essa separação de serviços, caso seja necessário no futuro.

O Message Broker é simplesmente um corretor de mensagens, que opera sobre o protocolo MQTT, e é responsável pelo recebimento, entrega e armazenamento das mensagens que trafegam no sistema. O protocolo MQTT foi escolhido por ser extremamente leve e desenvolvido especialmente para o uso em Internet das Coisas.

4.4.3.3 Camada de Negócio

Os microserviços do sistema, que aplicam as regras de negócios, estão presentes na Camada de Negócio. Os componentes dessa camada são responsáveis pela validação de consistência das informações, interação com o banco de dados na Camada de Persistência através da interface ODBC, e processamento dos dados coletados pelo dispositivo. Cada microserviço que compõe essa camada é especializado e responsável por uma quantidade restrita de entidades, prezando-se pelo baixo acoplamento do sistema. Simplificando a implantação do sistema, alguns serviços foram agrupados em um mesmo servidor, mediante sua afinidade em relação às informações em que se especializa.

O Servidor de Registro (figura 36) comporta os Serviços de Usuários e de Dispositivos, que são responsáveis pelo cadastro, recuperação, atualização e deleção (CRUD, em inglês) dos registros de usuários e registros, respectivamente. Além disso, o Serviço de Usuários

controla a associação entre um usuário e seus dispositivos.

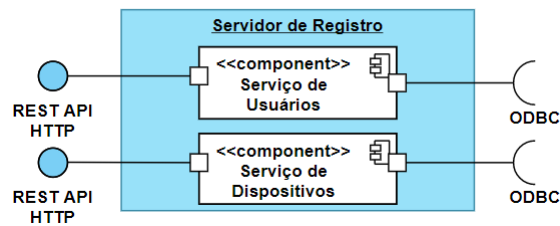


Figura 36: Camada de Negócio - Servidor de Registros.

Fonte: Autores.

Os Serviços de Controle, Receitas e Lotes também exercem a responsabilidade sobre o CRUD de perfis de controle, receitas e lotes, respectivamente, e foram agrupados no Servidor de Receitas (figura 37). O Serviço de Lotes também realiza a associação de um lote com o dispositivo e envia todas as informações necessárias para o controle do lote da receita para o dispositivo que foi associado, essa comunicação é realizada por meio de mensagens sob o protocolo MQTT, que serão consumidas pelo dispositivo conectado à Internet.

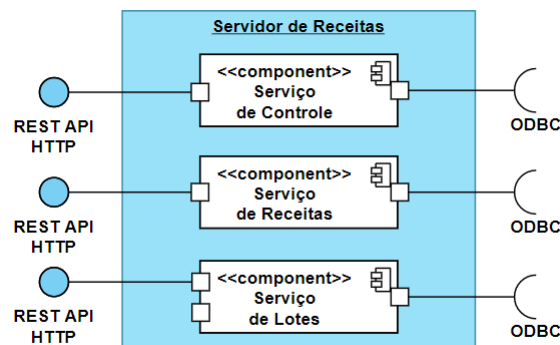


Figura 37: Camada de Negócio - Servidor de Receitas.

Fonte: Autores.

A figura 38 ilustra os demais serviços da Camada de Negócios. O Serviço de Análises empenha função analítica sobre os dados coletados que foram consolidados, fornecendo relatórios de desempenho para o usuário, além de dados para serem exibidos na Aplicação Web. O Serviço de Processamento, ou Processador, acessa os dados enviados pelo dispositivos ao Message Broker e aplica as transformações necessários para consolidar as informações no banco de dados. Adicionalmente, um Serviço de Autenticação foi incluído para atender às necessidades de segurança dos dados e controle de acesso ao sistema.

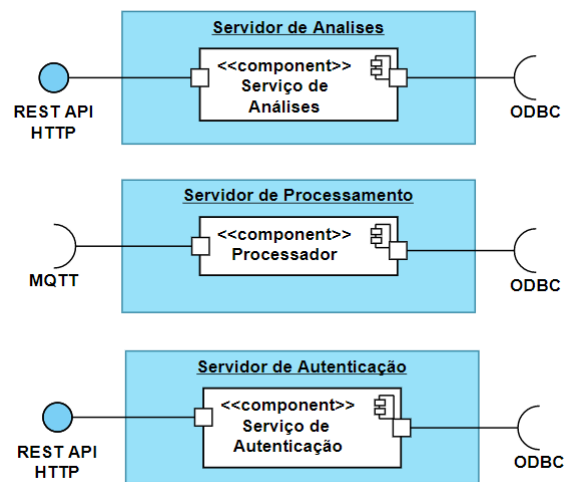


Figura 38: Camada de Negócio - Servidores de Análises, Processamento e Autenticação.
Fonte: Autores.

4.4.3.4 Camada de Persistência

A Camada de Persistência (figura 39) abriga os serviços de armazenamento de dados necessários ao sistema. Neste projeto, está previsto apenas um servidor de banco de dados relacional, discutido em maior detalhe na Modelagem de Entidade-Relacionamento.

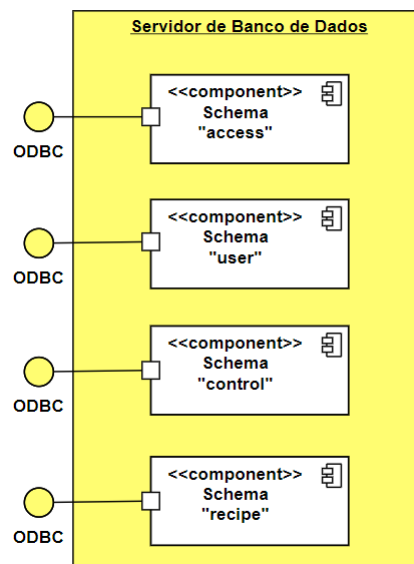


Figura 39: Camada de Persistência.
Fonte: Autores.

4.4.4 Fluxo de Interação Interface Web

Seguindo o mapeamento de interações do usuário com o sistema definido pelos casos de uso apresentados, foi projetado o diagrama da figura 41 contendo as telas previstas para a Interface Web e o fluxo de interação entre as telas esperado. No diagrama, os blocos simbolizam as telas da interface, enquanto as setas representam ações que levam de uma tela a outra, que podem ser botões (como um botão de "listar receitas") ou o resultado de uma operação (como o fim da operação de cadastrar um novo usuário).

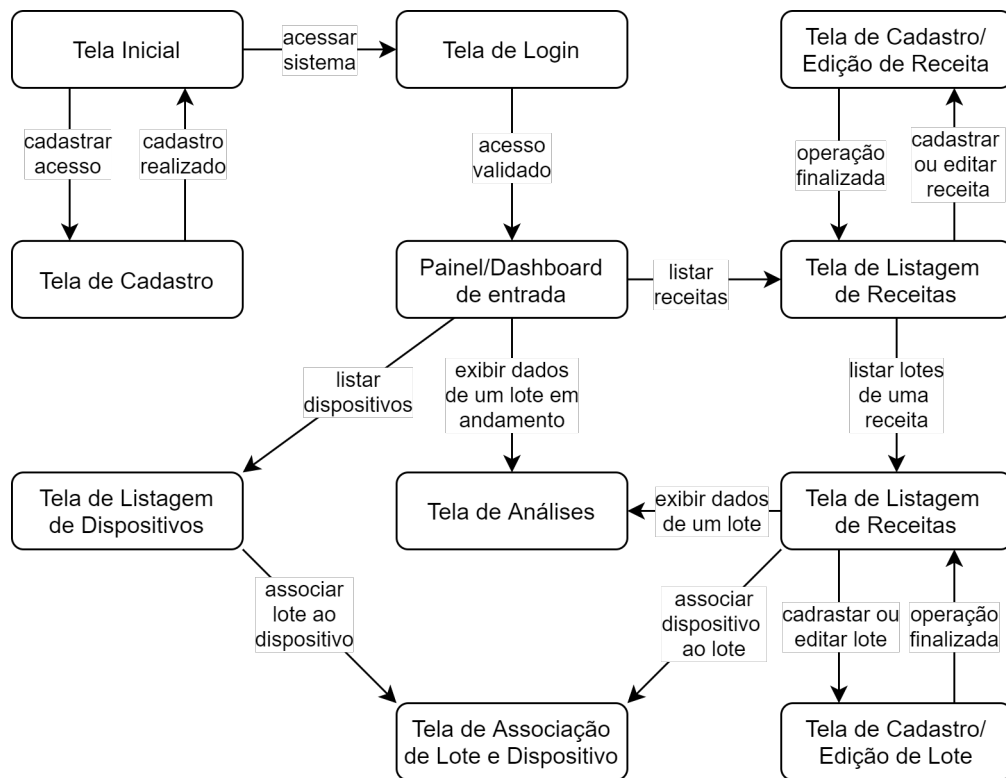


Figura 41: Fluxo de telas e interações da Interface Web.

Fonte: Autores.

4.4.5 Definição de Tecnologias

As tecnologias a serem utilizadas na implementação implantação deste projeto foram definidas levando em consideração sua adequação e familiaridade dos autores. Dessa forma, para implementação dos microsserviços que transacionam os dados, foi escolhida a linguagem de programação Java, sendo uma linguagem fortemente tipificada, é natural controlar o formato dos dados que devem ser salvos, simplificando validações de consistência. A biblioteca Spring Boot irá prover a API REST necessária.

Para o desenvolvimento do API Gateway e dos serviços mais pontuais, como Processa-

mento, Análise e Autenticação, foi escolhida a linguagem Python, que para componentes menores, oferece um overhead menor de arquivos de classe e de configuração, conferindo um pouco mais de agilidade na implementação. A interface de programação é simplesmente provida pela união do framework Flask com o servidor Gunicorn. A segurança da aplicação seguirá padrão OAuth 2.0 implementado pela biblioteca oauthlib.

Para desenvolvimento da interface da Aplicação Web optou-se pela utilização da biblioteca React da linguagem JavaScript, utilizando também a biblioteca chart.js para construção de gráficos. Para a interface do Aplicativo Mobile foi definida a linguagem Java, para utilização em smartphones Android.

Todos os dados do sistema serão armazenados no banco de dados relacional PostgreSQL. Como o projeto prevê diversos componentes, torna-se vantajosa a utilização de tecnologia de containerização de aplicações, nesse caso, foi escolhida a tecnologia Docker que permite definir programatizar a implantação dos vários componentes do sistema, através de arquivos de configuração, tornando a tarefa de disponibilização dos serviços mais rápida e escalável.

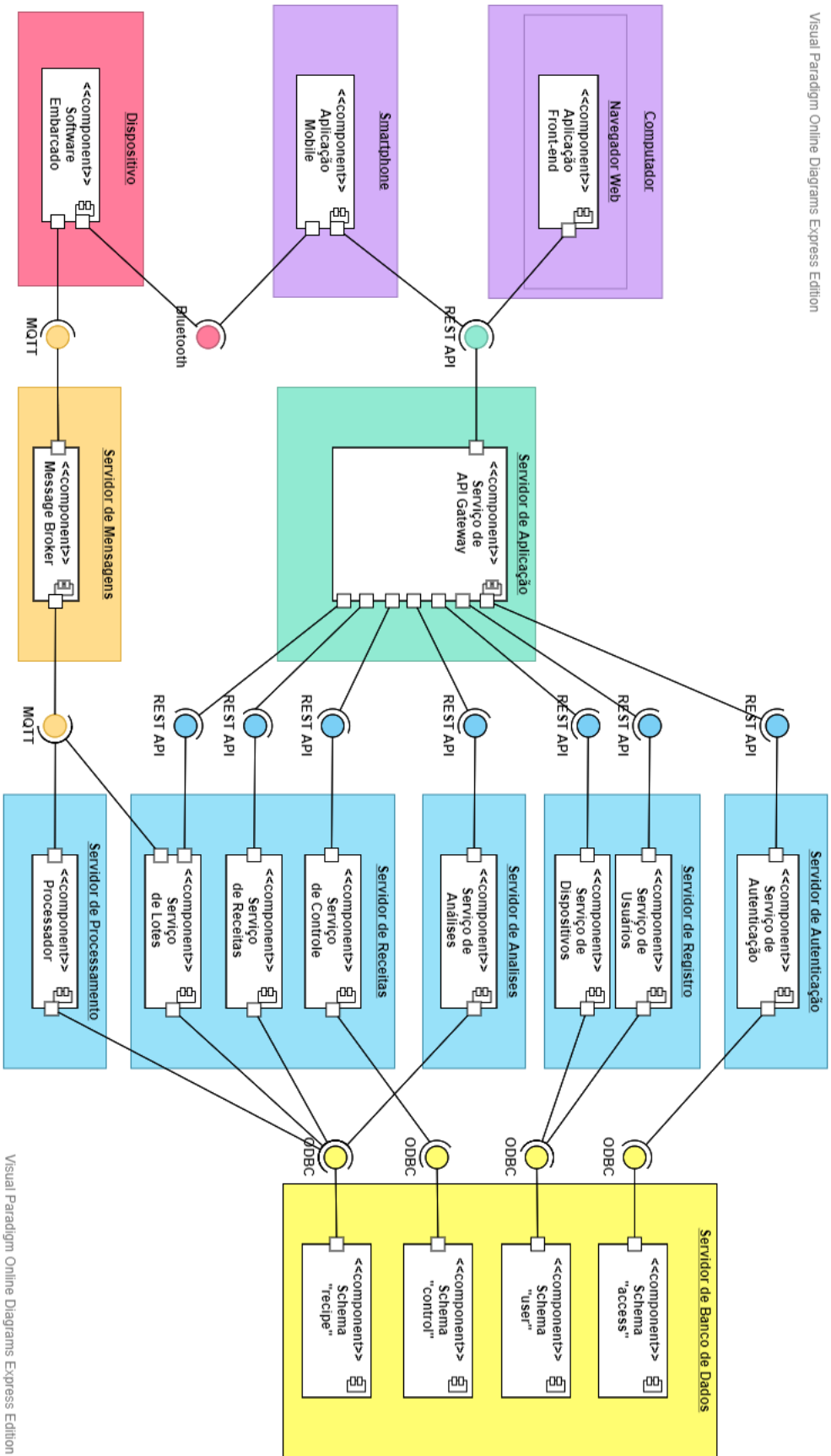


Figura 40: Diagrama de Componentes.

Fonte: Autores.

5 IMPLEMENTAÇÃO

5.1 Implementação de Hardware

5.1.1 Sensor de Temperatura

O sensor de temperatura DS18B20 é alimentado pelos pinos de 5V e GND (terra) do microcontrolador. seu sinal de leitura é enviado para uma porta digital, sendo utilizado um resistor de 4,7 k Ω como pull up, caso exista uma leitura errada do sensor. A figura 42 ilustra a conexão do microcontrolador com o sensor de temperatura.

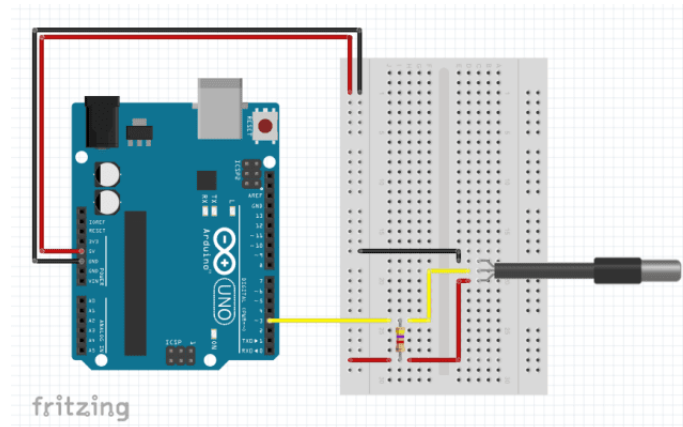


Figura 42: Esquema de conexão entre microcontrolador e sensor de temperatura DS18B20.

Fonte: <https://portal.vidadesilicio.com.br/>.

O excerto de código em linguagem Arduino a seguir representa a função utilizada para ler o valor de saída do sensor. A biblioteca OneWire (<https://github.com/PaulStoffregen/OneWire>) implementa o protocolo proprietário de comunicação serial da Dallas Semiconductor, fabricante do sensor. Ela é utilizada em conjunto com a biblioteca livre DallasTemperature (<https://github.com/milesburton/Arduino-Temperature-Control-Library>) para estabelecer a comunicação com o DS18B20.

```
#include <OneWire.h>
```

```

#include <DallasTemperature.h>
#define ONE_WIRE_BUS D6

OneWire oneWire(ONE_WIRE_BUS);
DallasTemperature sensors(&oneWire)

void setup() {
    sensors.begin();
}

float readTemperature() {
    return sensors.getTempCByIndex(0);
}

```

5.1.2 Sensor de pH

O sensor de pH é conectado a um circuito auxiliar que trata o sinal proveniente da ponta de prova. Esse circuito é conectado aos pinos de 5V e GND do microcontrolador para alimentação, e envia do dado coletado por meio de sinal analógico, como observado na figura 43.

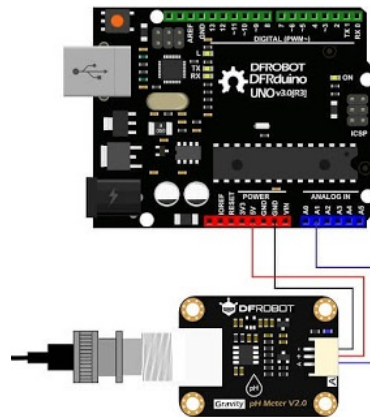


Figura 43: Esquema de conexão entre microcontrolador e sensor de pH E-201-C.

Para utilização desse sensor, é necessário sua calibração, que consiste na medição da tensão de saída do sensor em soluções com diferentes concentrações de H^+ e obtenção de uma equação linear que traduza a voltagem medida em um valor de pH. Para isso

utilizadas soluções tampão com pH iguais a 4, 7 e 10. A calibração seguiu o seguinte processo:

1. Com a solução de pH igual a 7, ajustar o ganho do circuito (por um potenciômetro) até que a leitura de voltagem seja 2,5 Volts. Essa tensão foi escolhida de forma a coincidir valor médio da tensão de alimentação do circuito com o valor médio da escala de pH;
2. Capturar a tensão medida nas soluções com pH 4 e 10;
3. Aplicar regressão linear, minimizando a soma dos erros quadráticos, obtendo a reta que relaciona a tensão medida com o pH da solução

Com os valores obtidos na tabela 5, foi obtida a equação linear 5.1.

$$pH = 2.0171 \cdot Voltagem + 1.7152 \quad (5.1)$$

Tabela 5: Medições de tensão em soluções tampão para calibração do sensor de pH.

Voltagem	pH
2.50	7.0
1.20	4.0
4.16	10.0

A leitura do pH medido pelo sensor pelo microcontrolador segue o seguinte algoritmo:

1. Valor inteiro entre 0 e 1024 recebido pelo sensor é lido na porta analógica;
2. Valor lido é convertido em uma diferença de tensão seguindo $Voltagem = medida \cdot 5/1024$;
3. Diferença de tensão é convertida na leitura de pH seguindo equação linear obtida na calibração do sensor.

5.1.3 Sensor de pressão diferencial para medida de densidade

A medição de densidade foi implementada indiretamente através do sensor de pressão diferencial MP3V5010DP. Esse sensor é inserido no líquido fermentado através de dois

tubos posicionados em uma distância fixa de 18 centímetros. A densidade é inferida através da fórmula:

$$\rho = \frac{\Delta P}{g \cdot \Delta h} \quad (5.2)$$

O chip Pmod DPG1 se comunica com o microcontrolador através da interface periférica serial (SPI), que é um sistema de comunicação que utiliza as portas: SCK: Serial Clock (output from master), MISO: Master In Slave Out (data output from slave) e SS: Slave Select.

O sensor é alimentado com 3.3 V e é utilizado um conversor de nível lógico bidirecional para converter o sinal digital de 5V para a tensão do sensor. A biblioteca SPI (<https://github.com/PaulStoffregen/SPI>) foi utilizada para realizar essa comunicação e leitura. O valor recebido pelo microcontrolador é convertido através da fórmula fornecida no datasheet do chip:

$$\Delta P = \frac{\frac{value}{4096} - 0.08}{0.08} \quad (5.3)$$

5.1.4 Conexão e Comunicação do Microcontrolador

Com os sensores propriamente conectados e calibrados, o microcontrolador foi configurado para se conectar à Internet através de conexão *Wi-Fi* e a enviar os dados coletados por meio do protocolo MQTT. O Wemos D1 é controlada pelo módulo ESP8266, de forma que ofereça conectividade *Wi-Fi* nativa. A biblioteca ESP8266 *Wi-Fi* (<https://github.com/esp8266/Arduino/tree/master/libraries/ESP8266WiFi>) foi utilizada para estabelecer a conexão *Wi-Fi*, em conjunto com a biblioteca PubSubClient (<https://github.com/knolleary/pubsubclient>) para publicar mensagens e se inscrever em tópicos MQTT.

5.1.5 Controle de Temperatura

Um controlador de malha fechada proporcional interativo derivativo (PID) é utilizado para controlar a temperatura do fermentador. Esse método é amplamente utilizado na indústria, possuindo boa precisão e confiabilidade, além de ser facilmente sintonizado. Inicialmente são definidos parâmetros analógicos e depois é criado o controle digital.

Sendo assim, o sistema com a pastilha de Peltier deve retirar do sistema uma quantidade de calor proporcional a essa recebida e gerada para manter uma temperatura estável abaixo do ambiente. Como a quantidade de calor retirada por um Peltier é proporcional a corrente, a tensão fornecida a pastilha é variada através de uma circuito ponte H que tem a sua amplitude de tensão de saída regulada por uma porta PWM.

5.1.5.1 Circuito Ponte H

O circuito ponte H (figura 44) permite que a amplitude e sentido da tensão de entrada da pastilha de Peltier seja controlado digitalmente por um sinal PWM (Pulse Width Modulation) do microcontrolador. No Software embarcado, um valor entre 0 e 255 é utilizado para definir o duty cycle da onda quadrada do sinal PWM, como exemplificado na figura 45.

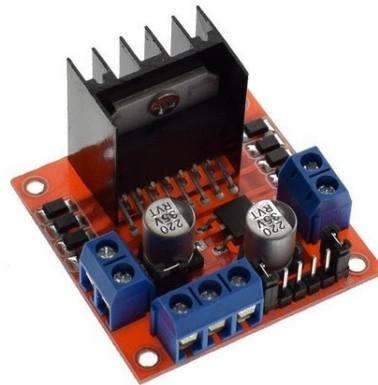


Figura 44: Modelo ponte H - L298N.

Fonte: <https://mercadolivre.com.br/>.

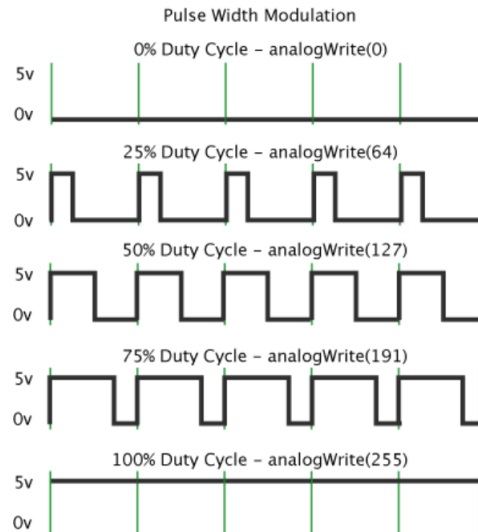


Figura 45: Onda quadrada do sinal PWM para alguns valores especificados no micro-controlador.

Fonte: <https://www.arduino.cc/>.

5.1.5.2 Montagem do Dispositivo

Com as definições de controle discutidas, foi projetado o dispositivo da figura 46, em escala aproximada. O dispositivo pode ser dividido em três partes descritas a seguir:

1. Parte quente : Essa parte é ligada na face quente da pastilha e é formada por uma ventoinha e um dissipador de calor. Uma boa dissipação de calor garante uma melhor eficiência da partilha e permite que ela forneça quantidades de calor maiores para o sistema;
2. Pastilha de Peltier: Pastilha que recebe a corrente e transforma energia elétrica em térmica. A face quente e face fria são isoladas por material isolante térmico, que ajuda na não interferência entre as partes e melhora a eficiência do sistema;
3. Parte fria: essa parte é ligada na face fria da pastilha. Ela retira calor do sistema através de uma placa de cobre e barras de aço inox. Apesar do cobre ter uma condutividade térmica melhor, o aço inox foi escolhido para fazer a transmissão para o mosto + leveduras, por não interferir no seu gosto na exposição de pHs menores.

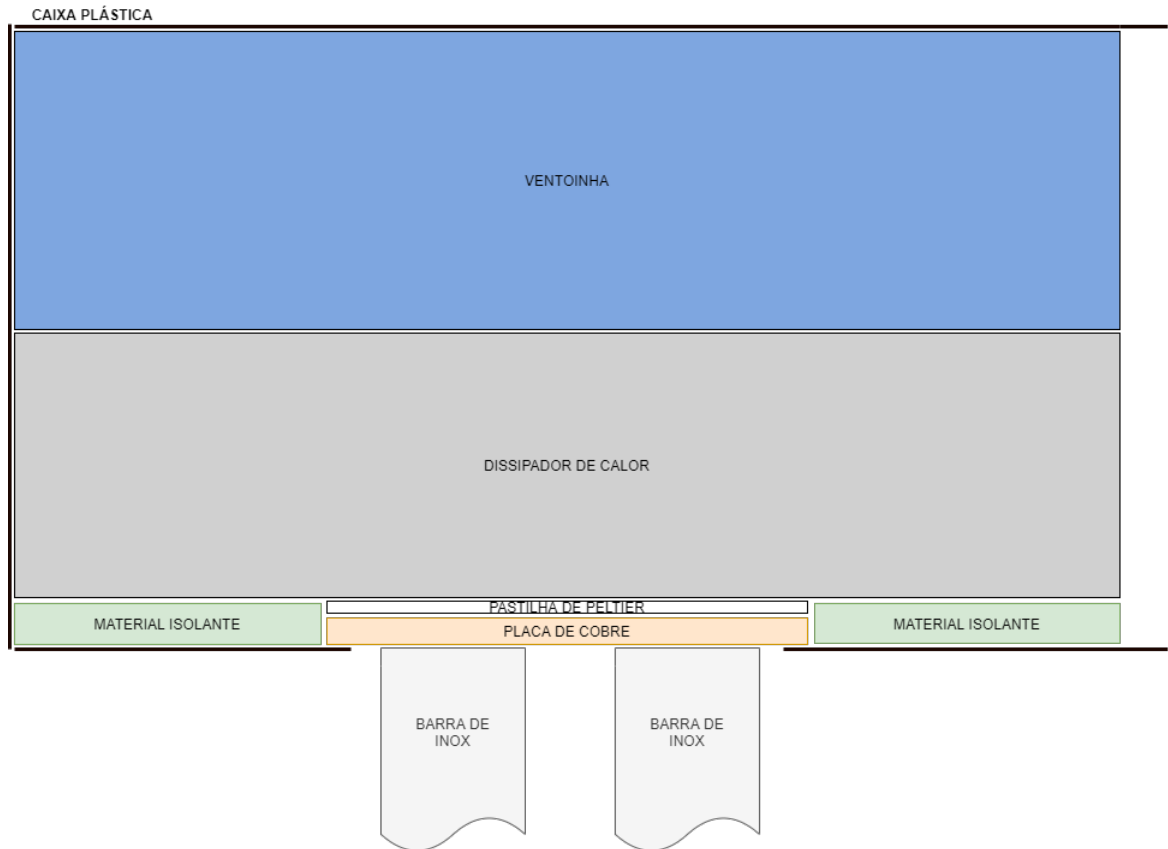


Figura 46: Esquema do perfil do dispositivo controlador de temperatura, escala aproximada.

Fonte: Autores.

5.2 Implementação de Software

A implementação do software seguiu a estratégia de se implementar primeiramente componentes que não possuem dependência para funcionar, seguindo para os componentes que tem como dependência os que foram implementados e assim por diante. De forma geral, foi seguido o diagrama de componentes da figura 40 no sentido da direita para esquerda, sendo implementadas em ordem: as camadas de banco de dados, de Negócio, Intermediária e de Interface.

Além disso, foram priorizados serviços essenciais para o funcionamento geral do projeto, i.e., os componentes de Aplicação Mobile e Autenticação, que garantem melhoras na utilização do usuário e segurança da informação, foram considerados menos prioritários que os demais componentes, que atuam desempenham a função primária do projeto, que é controlar e monitorar a temperatura de fermentação de cervejas.

Os links de acesso para os repositórios de código fonte por componente são listados

no apêndice A. Eles foram salvos na plataforma Github e serão mantidos públicos para consulta e oportunidade de continuidade do projeto, desde que seguindo os termos de licença especificados em cada repositório.

5.2.1 Camada de Persistência

A Camada de Persistência foi rapidamente implementada instanciando-se um banco de dados na nuvem da Amazon Web Services (AWS) pelo serviço RDS e executando o script de Data Definition Language (DDL) que foi gerado a partir do diagrama apresentado anteriormente na figura 33. A plataforma utilizada para confeccionar o diagrama (<https://dbdiagram.io/>) oferece o download do código de geração do banco de dados a partir do diagrama. Foram então realizadas algumas modificações no script gerado, para incluir a criação de índices e estabelecer o relacionamento entre as tabelas. O arquivo de extensão *sql* para configuração do banco de dados está disponível no seguinte repositório de código: <https://github.com/josehcls/tcc-config>. O banco de dados foi provisionado pela plataforma de cloud computing AWS (Amazon Web Services)

5.2.2 Camada de Negócio

5.2.2.1 Servidores de Registro e Receitas

Após a configuração do banco de dados, foram implementados os componentes responsáveis por transacionar os registros de Usuários, Dispositivos, Receitas, Lotes e Perfis de Controle. Os servidores de Registro (<https://github.com/josehcls/tcc-register-service>) e Receitas (<https://github.com/josehcls/tcc-recipe-service>) foram desenvolvidos na linguagem Java e desempenham as funções básicas de transação de dados: listagem, criação, atualização e deleção.

O framework Spring foi utilizado para prover uma interface de programação REST, que serve de ponto de comunicação com o Servidor de Aplicação, ou API Gateway do projeto. Os servidores recebem solicitações de transação por meio dessa interface, realizam validações de consistências de dados e realizam as transações por meio de conexão direta ao banco de dados. Para validar as operações, foram criados testes automatizados durante o desenvolvimento, que podem ser executados para verificar a eficácia de cada um dos métodos implementados.

Após a implementação e testes gerais, foi confeccionado o arquivo Dockerfile para configurar a geração de um container Docker a partir do pacote jar (Java Archive), resultado

da compilação de cada um dos componentes. Esse container é utilizado para simplificar e otimizar a implantação desses componentes, de forma a permitir o provisionamento de uma instância do serviço em nuvem com facilidade e celeridade.

5.2.2.2 Servidor de Processamento

O Servidor de Processamento (<https://github.com/josehcls/tcc-processor>) foi desenvolvido para receber as mensagens enviadas pelo Dispositivo com os dados coletados, processá-las e salvar as informações no banco de dados. Ele foi implementado na linguagem Python utilizando a biblioteca paho-mqtt, o programa se inscreve no tópico de mensagens MQTT para qual o Dispositivo publica as mensagens, e um código de tratamento é acionado a cada nova mensagem recebida.

Na escala de um Dispositivo em operação, não se notou atraso significativo entre o recebimento da mensagem e disponibilização das informações no banco de dados, porém, uma possível melhoria desse sistema seria inserir a mensagem recebida em uma fila de processamento, utilizando softwares como o RabbitMQ, que então seria processada. Essa melhoria garantiria a retenção de mensagens em casos de queda do banco de dados e evitaria que o processamento e transação no banco de dados, que são as ações mais custosas desse fluxo, se sobrecarreguem e criem gargalos.

5.2.2.3 Servidor de Análises

Com a função de extrair e preparar as informações coletadas durante a fermentação para o usuário, foi implementada o Servidor de Análises (<https://github.com/josehcls/tcc-analytics>) utilizando a linguagem Python em conjunto com o framework Flask para providenciar uma interface de comunicação. Essa é uma aplicação simples que, ao receber a solicitação dos dados de um determinado lote, já finalizado ou vigente, carrega os dados armazenados em banco de dados e os manipula para fornecê-los em um formato mais adequado para a exibição em gráficos.

Essa manipulação consiste em agrupar as medidas referentes à mesma variável, como temperatura, densidade e pH, e construir objetos JSON contendo a lista dos pontos da série temporal. O formato foi determinado para possibilitar compatibilidade imediata com a ferramenta Chart.js, escolhida para exibir os gráficos na Interface Web.

5.2.3 Camada de Aplicação

O Servidor de Aplicação (<https://github.com/josehcls/tcc-app>) atua como uma camada intermediária entre os microsserviços da Camada de Negócio e o cliente, que interage pela Interface Web. A aplicação consiste em diversos pontos de acesso que correspondem a métodos que realizam uma ou mais chamadas aos microsserviços, agregando os resultados quando necessário. Essa camada permite que os demais serviços fiquem isolados do cliente e que todo o controle de acesso e medidas de segurança fiquem centralizadas nessa camada, evitando a necessidade de se re-implementar essas medidas em cada serviço. O serviço foi desenvolvido utilizando a linguagem Python e o framework Flask para provisionamento da API. Assim como os outros serviços, essa aplicação é provisionada por meio um container Docker, que instala as dependências da aplicação e executa a aplicação Flask através do servidor HTTP Gunicorn.

5.2.4 Interface Web

Seguindo o fluxo de interações da interface exposto no capítulo de projeto pela figura 41, a Interface Web (<https://github.com/josehcls/tcc-front-end>) foi desenvolvida utilizando, principalmente, a biblioteca React para JavaScript. Com ajuda dessa e de outras ferramentas atuais, foi possível desenvolver uma plataforma simples e funcional com agilizada. O React, como outras ferramentas recentes, permite a abstração de boa parte da diagramação e funcionalidades de sites, o que simplifica em grande escala o desenvolvimento. Para a exibição de gráficos, foi utilizada a biblioteca Chart.js, que simplifica o processo de construção de gráficos como o da figura 47 que compara a evolução de temperatura e densidade.

5.2.5 Implantação

A partir dos containers que definidos por cada aplicação de backend, a implantação de todos os sistemas podem ser orquestrados em um único comando utilizando a ferramenta Docker Compose definindo uma aplicação multi-container. Existem hoje tecnologias mais robustas como o Kubernetes para fazer o provisionamento e monitoramento de vários containers, mas o Docker Compose foi escolhido pela sua simplicidade. Dessa forma, os serviços backend e o broker de mensagens MQTT são provisionados em uma máquina virtual oferecida pelo serviço de computação em nuvem AWS.

A configuração dos serviços é feita de forma que apenas o broker e o Servidor de

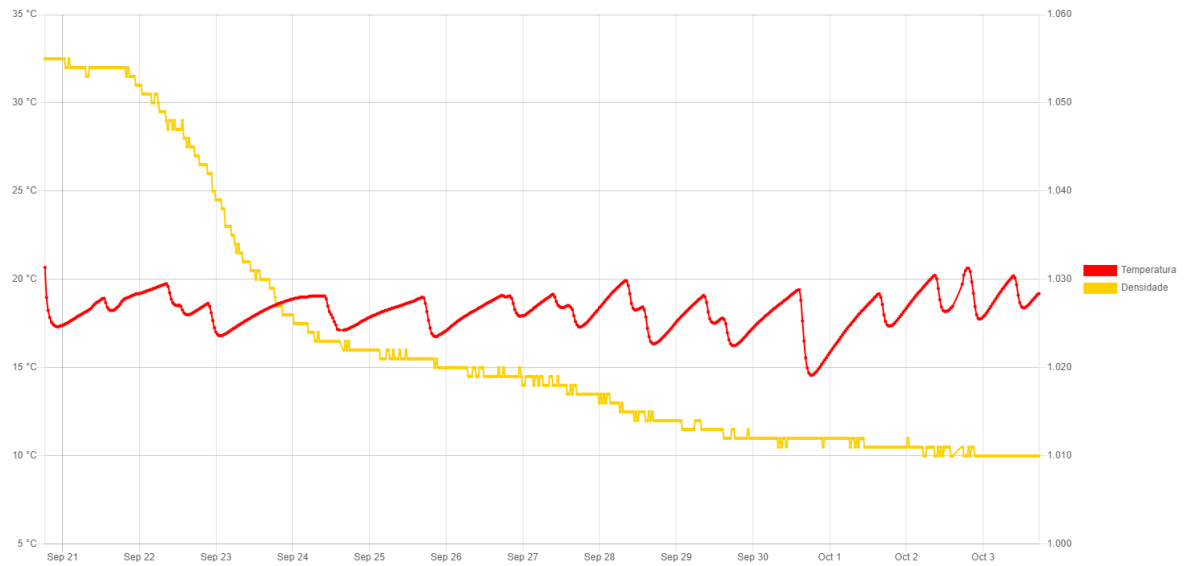


Figura 47: Exemplo de gráfico de evolução das variáveis.

Fonte: Autores.

Aplicação estejam expostos a acesso externo, protegendo os demais serviços. Essa configuração pode ser conferida no arquivo *docker-compose.yaml* no repositório de código: <https://github.com/josehcls/tcc-config>.

A Interface Web foi provisionada pela plataforma Vercel, que oferece o serviço de hospedagem gratuito, sob condições, e algumas facilidades como construir e disponibilizar o acesso à plataforma web a partir do código no repositório do GitHub.

6 TESTES E AVALIAÇÃO DO PROTÓTIPO

O intuito original dos testes de validação era aferir o funcionamento do protótipo durante uma fermentação real, realizada em pequena escala exclusivamente para avaliação desse projeto. Contudo, devido a atrasos no cronograma nas últimas etapas de montagem do protótipo e documentação, não foi possível preparar esse teste. Decorrente disso, o protótipo foi testado em situações não ideais, mas ainda assim relativamente semelhantes ao processo real, e os dados coletados foram observados na Interface Web e comparados com valores inferidos manualmente,

Primeiramente foram testados os sensores do protótipo. Para o sensor de temperatura, utilizou-se um recipiente com água comparando a temperatura lida em diferentes temperaturas com um termômetro analógico. Considerando a resolução da escala do termômetro analógico de 1°C e a resolução de duas casas decimais do sinal digital, todas as medidas feitas estavam de acordo e dentro da margem de erro. O sensor de pH foi validado após a calibração, usando as soluções tampão de pH 4, 7, e 10, com erro máximo obtido de 0,1 unidade de medição, o que é suficiente para a aplicação.

Por fim, o sensor de densidade foi testado em água, mantendo uma distância de 18 cm entre os dois pontos de medição, e comparado com o resultado de um densímetro analógico, obtendo um erro de aproximadamente 0,3%, que foi bem abaixo do valor esperado, dado a acurácia nominal de 5% do sensor, seria interessante coletar mais medições com diferentes fluidos para melhor aferimento do erro do sensor. Caso o sensor apresentasse um resultado pouco confiável, ele poderia ser usado apenas para conferir uma noção qualitativa de evolução do processo pela tendência de mudança da densidade, porém, ele aparenta ser adequado para cálculo de índices relevantes ao processo como gradação alcoólica, eficiência e atenuação, que se baseiam nos valores de densidade relativa antes e depois da fermentação.

O atuador de temperatura, composto pela pastilha de Peltier com o dissipador de calor e barra de aço inoxidável para troca de calor com o mosto foi também testado em água, utilizando um fermentador de plástico com capacidade de 20 litros. Foi montado um

isolamento simples ao entorno do fermentador, utilizando poliestireno expandido (comumente denominado "isopor") e espuma vinílica acetinada (EVA). A partir da temperatura ambiente de aproximadamente 23°C , o atuador foi ligado em potência máxima para medir a potência real de retirada de calor do sistema, obtendo-se algo próximo de $0,3^{\circ}\text{C}$ por hora, para resfriar os 20 litros de água. Acredita-se que esse desempenho é insuficiente, e que um melhor isolamento térmico e possíveis melhorias na montagem do sistema de troca de calor possam melhorá-lo. Contudo, não se avançou nesse aspecto.

7 CONCLUSÃO

7.1 Considerações Finais

Ao final do presente trabalho, tem-se em mãos um dispositivo que coleta dados de temperatura, densidade e pH e os envia para um servidor online e que aplica um algoritmo de controle sobre a temperatura, e também um sistema de software que recebe os dados coletados e os exibe em poucos segundos em um gráfico online. Apesar do controle ainda necessitar de melhorias de montagem e isolamento térmico para atingir a potência esperada, e o software desenvolvido não ter atendido completamente o processo, como implementação de controle de acesso e o fluxo para configuração do dispositivo previsto nos casos de uso; acredita-se que foi desenvolvido um protótipo viável com um caminho evidente de melhorias.

Contudo, lamenta-se de não se ter dedicado o empenho desejado na etapa de teste e validação do protótipo. Essa etapa foi prejudicada por atrasos no cronograma, principalmente no momento intermediário do projeto. Com certeza, um controle mais rígido das tarefas e disciplina com o cronograma teria afetado positivamente o projeto, especialmente nessa etapa.

Por outro lado, as etapas de projeto e prototipação em etapas foram consideravelmente proveitosas como uma aplicação prática e extensa dos conteúdos que foram estudados durante a graduação. Ter atuado nesse trabalho, que se iniciou com a busca de entendimento sobre um processo, seguindo por levantamento de idéias de melhoria e projeto de uma solução e finalizando pela implementação configurou um grande aprendizado de engenharia.

No mais, considerando o objetivo do projeto de interesse pessoal dos autores, espera-se dar continuidade ao trabalho após a entrega desse documento, e que esse relatório possa estimular outros alunos interessados e ajudá-los na elaboração de seus projetos.

7.2 Perspectivas de Continuidade

Assim como foi mencionado, foram notadas nesse documento algumas melhorias que podem ser feitas sobre o escopo do trabalho atual, como melhorar a aplicação do controle de temperatura, refinar a montagem do dispositivo, finalizar a implementação da Interface Web com controle de acessos e dados segregados por usuário, e refinar as interações entre hardware e software focando nos requisitos não funcionais levantados. Acredita-se que com poucas semanas adicionais, boa parte disso poderia ser realizado.

Sob a ótica acadêmica, uma continuação interessante ao trabalho seria estudar a viabilidade de estruturar o algoritmo de controle levando em consideração a densidade do mosto em fermentação. Isso poderia permitir inclusive reduzir o tempo de fermentação em alguns dias, como discutido brevemente por White e Zainasheff 2010, e possui grande valor para cervejarias, mesmo de grande porte que poderiam adaptar a estratégia a sua escala. Contudo, para essa abordagem, acredita-se ser necessário um maior entendimento do processo bioquímico e da natureza das leveduras.

Sob a ótica de mercado, aplicando-se um refinamento sobre o protótipo deve ser possível transformá-lo em um produto, assim como foi imaginado no início desse projeto. Os devidos estudo sobre mercado deveriam ser realizados, e seria interessante mais entrevistas com produtores para validar o produto, mas acredita-se sê-lo viável, ainda que para um mercado de nicho.

REFERÊNCIAS

- [Bezerra 2015]BEZERRA, E. *Princípios de Análise de Projeto de Sistemas com UML*. 3. ed. Rio de Janeiro: Elsevier, 2015. ISBN 9788535226263.
- [Boulton e Quain 2001]BOULTON, C.; QUAIN, D. *Brewing Yeast and Fermentation*. [S.l.]: Wiley, 2001. ISBN 9780632054756.
- [Kunze 2004]KUNZE, W. *Technology Brewing and Malting*. 3. ed. Berlin: VLB Berlin, 2004.
- [Lewis 2001]LEWIS, M. J. *Brewing*. 2. ed. New York: Kluwer Academic / Plenum Publishers, 2001.
- [Munroe 2006]MUNROE, J. H. Fermentation. In: PRIEST, F.; STEWART, G. (Ed.). *Handbook of Brewing*. 2. ed. New York: CRC Press, 2006. p. 487–524.
- [White e Zainasheff 2010]WHITE, C.; ZAINASHEFF, J. *Yeast: the practical guide to beer fermentation*. Colorado: Brewers Publications, 2010.

APÊNDICE A – CÓDIGO FONTE

Os códigos fonte do projeto foram hospedados em repositórios virtuais na plataforma Github. O link de acesso a cada repositório é listado a seguir.

Aplicação Web - Front-End do projeto: <https://github.com/josehcls/tcc-front-end>

Dispositivo - Software embarcado do dispositivo: <https://github.com/josehcls/tcc-device>

Servidor de Aplicação - Serviço de API Gateway: <https://github.com/josehcls/tcc-app>

Servidor de Registro - Serviços de Usuário e Registro: <https://github.com/josehcls/tcc-register-service>

Servidor de Análises - Serviço de Análises: <https://github.com/josehcls/tcc-analytics>

Servidor de Receitas - Serviços de Receitas, Lotes e Controle: <https://github.com/josehcls/tcc-recipe-service>

Servidor de Processamento - Serviço Processador: <https://github.com/josehcls/tcc-processor>

Configuração - Arquivos gerais, como Docker-compose para implantar os servidores e DDL do Banco de Dados: <https://github.com/josehcls/tcc-config>

Monografia - Arquivos utilizados na geração desta monografia (LaTeX): <https://github.com/josehcls/tcc-monografia>