

**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.

Área de Concentração:

Engenharia Elétrica

Orientador:

Prof. Dr. Carlos Eduardo Cugnasca

São Paulo  
2020

Dedicatória

# AGRADECIMENTOS

Thanks...

*“Epígrafe”*

-- Autor

# RESUMO

Resumo...

**Palavras-Chave** – Palavra, Palavra, Palavra, Palavra, Palavra.

# ABSTRACT

Abstract...

**Keywords** – Word, Word, Word, Word, Word.



## LISTA DE FIGURAS

1	Diagrama de Casos de Uso. . . . .	23
2	Camada de Interface. . . . .	24
3	Camada Intermediária. . . . .	25
4	Camada de Negócio - Servidor de Registros. . . . .	26
5	Camada de Negócio - Servidor de Receitas. . . . .	27
6	Camada de Negócio - Servidores de Análises, Processamento e Autenticação. . . . .	27
7	Camada de Persistência. . . . .	28
8	Diagrama de Entidade-Relacionamento. . . . .	30
9	Diagrama de Componentes. . . . .	31

## LISTA DE TABELAS

# LISTA DE SÍMBOLOS

Lista de símbolos...

# SUMÁRIO

<b>1</b>	<b>Introdução</b>	<b>13</b>
1.1	Objetivo . . . . .	13
1.2	Motivação . . . . .	13
1.3	Organização do Trabalho . . . . .	13
<b>2</b>	<b>Contexto</b>	<b>14</b>
2.1	História da Cerveja . . . . .	14
2.2	Fabricação de Cervejas . . . . .	14
2.2.1	Fermentação e Maturação . . . . .	14
2.3	Estado da Arte de Sistemas de Produção Artesanal . . . . .	14
2.3.1	Análise de Soluções Existentes . . . . .	14
<b>3</b>	<b>Metodologia do Trabalho</b>	<b>15</b>
<b>4</b>	<b>Projeto</b>	<b>16</b>
4.1	Especificação de Requisitos Técnicos . . . . .	16
4.1.1	Sistema Hardware . . . . .	16
4.1.1.1	Requisitos funcionais de Hardware . . . . .	16
4.1.1.2	Requisitos não funcionais de Hardware . . . . .	17
4.1.2	Sistema Software . . . . .	17
4.1.2.1	Requisitos funcionais de Software . . . . .	17
4.1.2.2	Requisitos não funcionais de Software . . . . .	18
4.2	Projeto de Controle . . . . .	18
4.2.1	Modelagem Térmica do Sistema . . . . .	18
4.3	Projeto de Hardware . . . . .	18

4.3.1	Sensores e Atuadores . . . . .	18
4.4	Projeto de Software . . . . .	18
4.4.1	Especificação dos Casos de Uso . . . . .	19
	UC - 1: Registro de Dispositivo . . . . .	20
	UC - 2: Cadastro de Receitas . . . . .	20
	UC - 3: Cadastro de Lotes . . . . .	21
	UC - 4: Envio das informações do Lote para Dispositivo . . . . .	21
	UC - 5: Envio das informações do Dispositivo para o Sistema . . . . .	22
	UC - 6: Visualização das Informações dos Lotes . . . . .	22
4.4.2	Modelo de Entidade-Relacionamento . . . . .	22
4.4.3	Arquitetura de Software . . . . .	23
4.4.3.1	Camada de Interface . . . . .	24
4.4.3.2	Camada Intermediária . . . . .	25
4.4.3.3	Camada de Negócio . . . . .	26
4.4.3.4	Camada de Persistência . . . . .	28
4.4.4	Definição de Tecnologias . . . . .	28
4.4.5	Projeto de Implantação . . . . .	29
<b>5</b>	<b>Implementação</b>	<b>32</b>
<b>6</b>	<b>Testes e Avaliação do Protótipo</b>	<b>33</b>
<b>7</b>	<b>Considerações Finais</b>	<b>34</b>
7.1	Conclusões do Projeto de Formatura . . . . .	34
7.2	Contribuições . . . . .	34
7.3	Perspectivas de Continuidade . . . . .	34
	<b>Referências</b>	<b>35</b>

<b>Apêndice A – Protótipos das Interfaces</b>	<b>36</b>
<b>Anexo A</b>	<b>37</b>

# **1 INTRODUÇÃO**

## **1.1 Objetivo**

O objetivo geral deste estudo é desenvolver um protótipo de um sistema que realize o monitoramento e controle do processo de fermentação de cervejas. O objetivo específico é 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**

## **1.3 Organização do Trabalho**

## **2 CONTEXTO**

### **2.1 História da Cerveja**

### **2.2 Fabricação de Cervejas**

#### **2.2.1 Fermentação e Maturação**

### **2.3 Estado da Arte de Sistemas de Produção Artesanal**

#### **2.3.1 Análise de Soluções Existentes**



### **3 METODOLOGIA DO TRABALHO**

## 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 100 °C), com precisão de 0,5 °C e intervalo de 1 minuto.

HW-F-2) O sistema deve monitorar o pH 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 50 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 50 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**

aaaa

## **4.3 Projeto de Hardware**

### **4.3.1 Sensores e Atuadores**

## **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.

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.

Adicionalmente, foram desenhados protótipos das interfaces que o usuário tem contato, e estão disponibilizadas no Apêndice A

#### **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 1.

### **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

## **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.



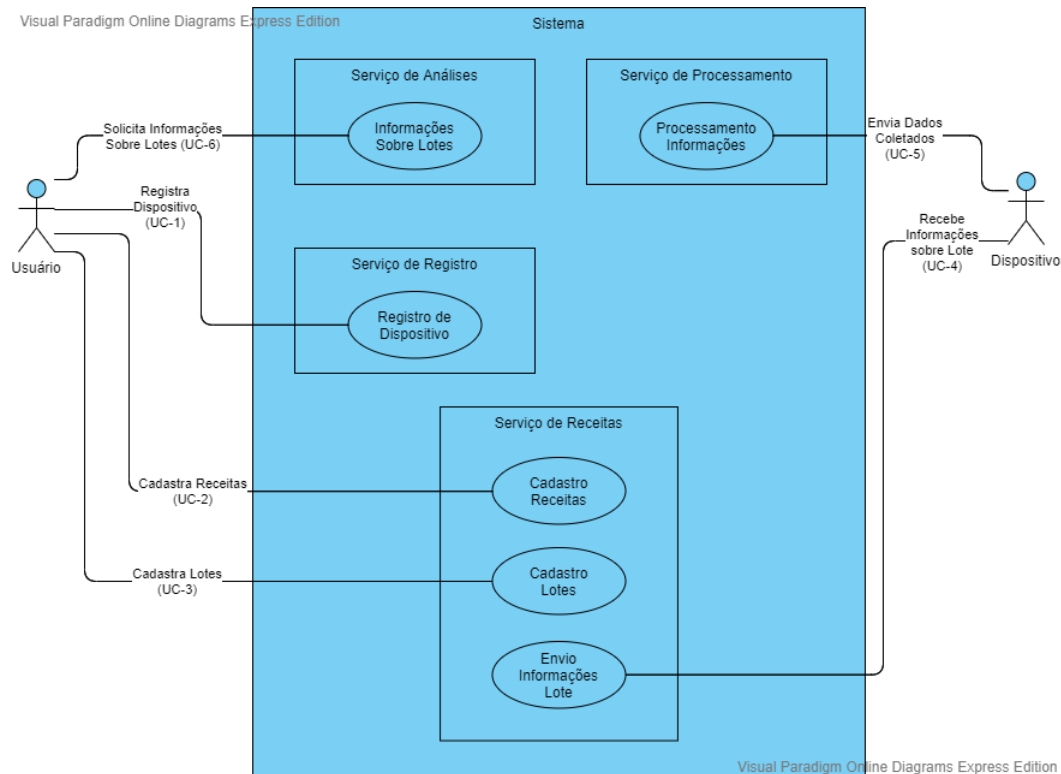


Figura 1: Diagrama de Casos de Uso.

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 microserviços 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 8, com destaque em cor para cada um dos schemas determinados.

#### 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 microserviços. 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 9.

#### 4.4.3.1 Camada de Interface

A Camada de Interface (figura 2) 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.

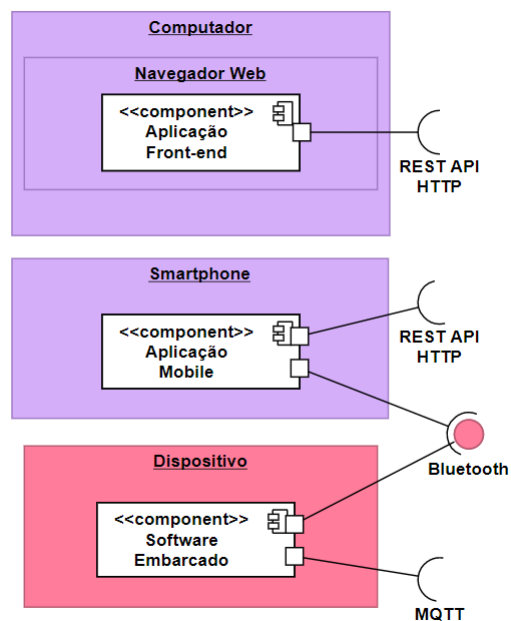


Figura 2: Camada de Interface.

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 va-

lidação 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 3). 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 microsserviços, 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.

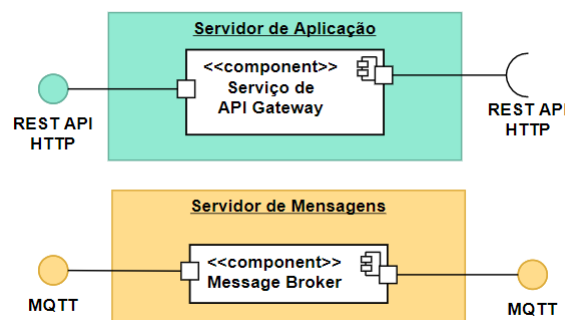


Figura 3: Camada Intermediária.

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 4) 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.

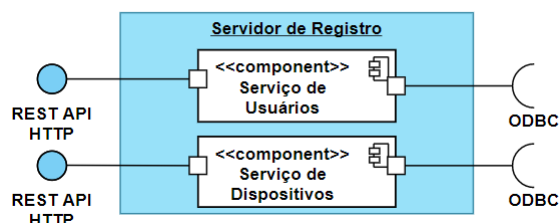


Figura 4: Camada de Negócio - Servidor de Registros.

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 5). 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.

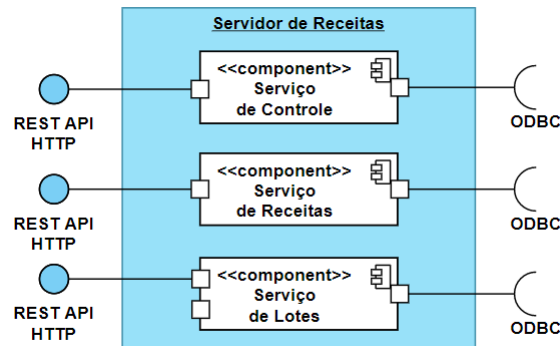


Figura 5: Camada de Negócio - Servidor de Receitas.

A figura 6 ilustra os demais serviços da Camada de Negócios. O Serviço de Análises emprega 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.

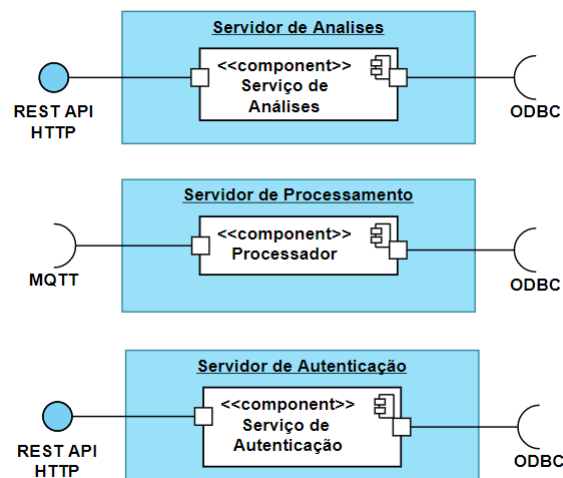


Figura 6: Camada de Negócio - Servidores de Análises, Processamento e Autenticação.

#### 4.4.3.4 Camada de Persistência

A Camada de Persistência (figura 7) 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.

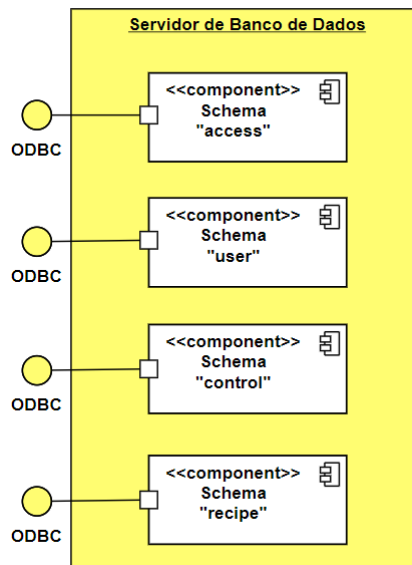


Figura 7: Camada de Persistência.

#### 4.4.4 Definição de Tecnologias

As tecnologias a serem utilizadas na implementação e implantação deste projeto foram definidas levando em consideração sua adequação, familiaridade dos autores com ela e facilidade de uso. Dessa forma, foi determinada a utilização da linguagem de programação Python para implementar os microsserviços e as lógicas necessárias no API Gateway. Utilizando o Python, é possível prover de forma simples e com poucos recursos uma API REST utilizando o framework Flask e sendo executado com o servidor Gunicorn. A segurança da aplicação segue padrão de mercado OAuth 2.0 com utilização da biblioteca oauthlib.

A interface da Aplicação Web utiliza a biblioteca React do JavaScript, utilizando também a biblioteca chart.js para construção de gráficos. O Aplicativo Mobile é desenvolvido em Java, para utilização em smartphones Android. O banco de dados relacional definido é o PostgreSQL.

Com finalidade de reduzir custos em primeiro momento e antecipando uma necessidade de escalabilidade ao sistema, a tecnologia de Kubernetes é empregada para preparar

e instanciar cada componente em um contêiner virtual, possibilitando a organização e controle de diversos componentes em uma mesma máquina virtual.

#### **4.4.5 Projeto de Implantação**

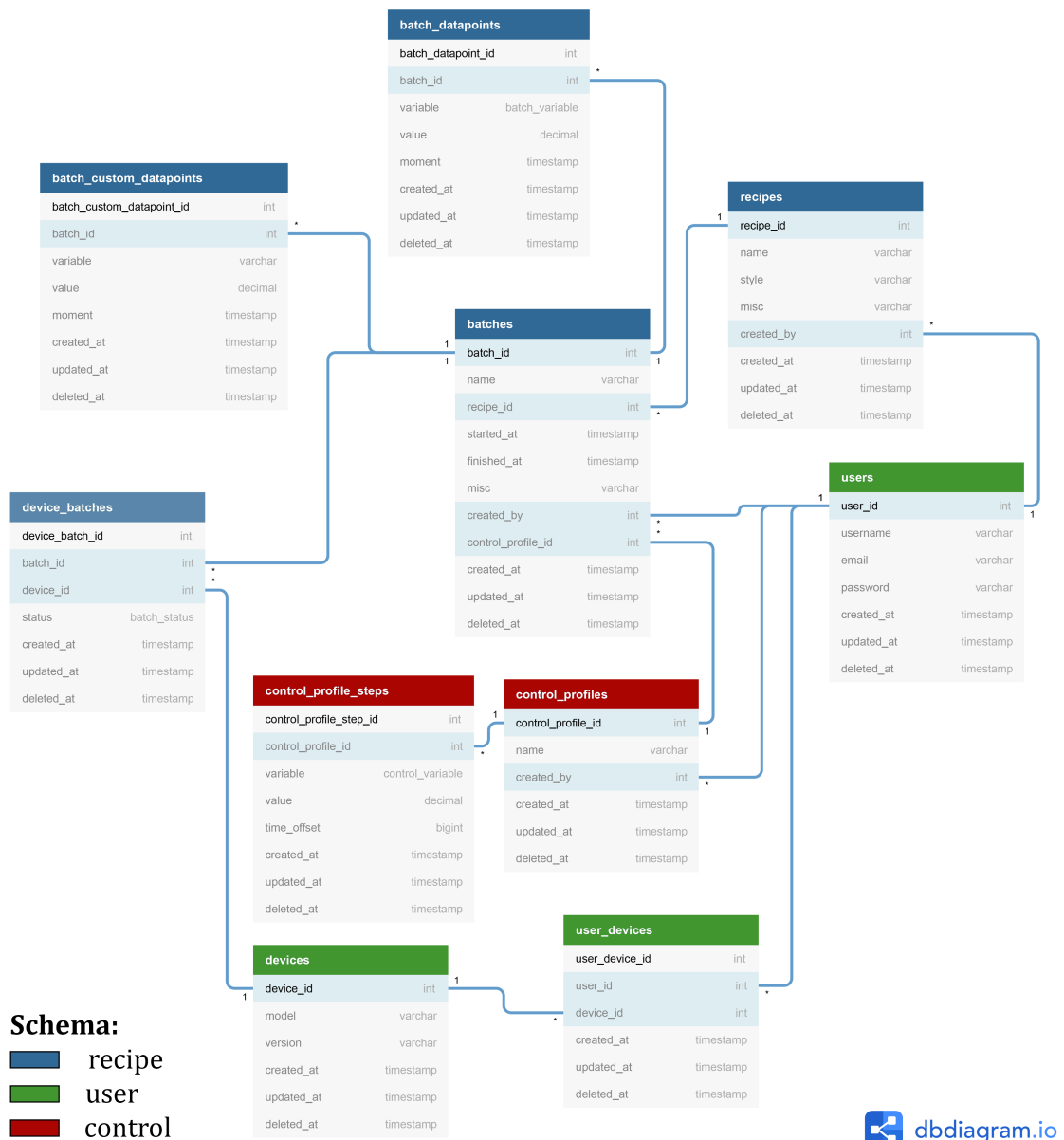


Figura 8: Diagrama de Entidade-Relacionamento.



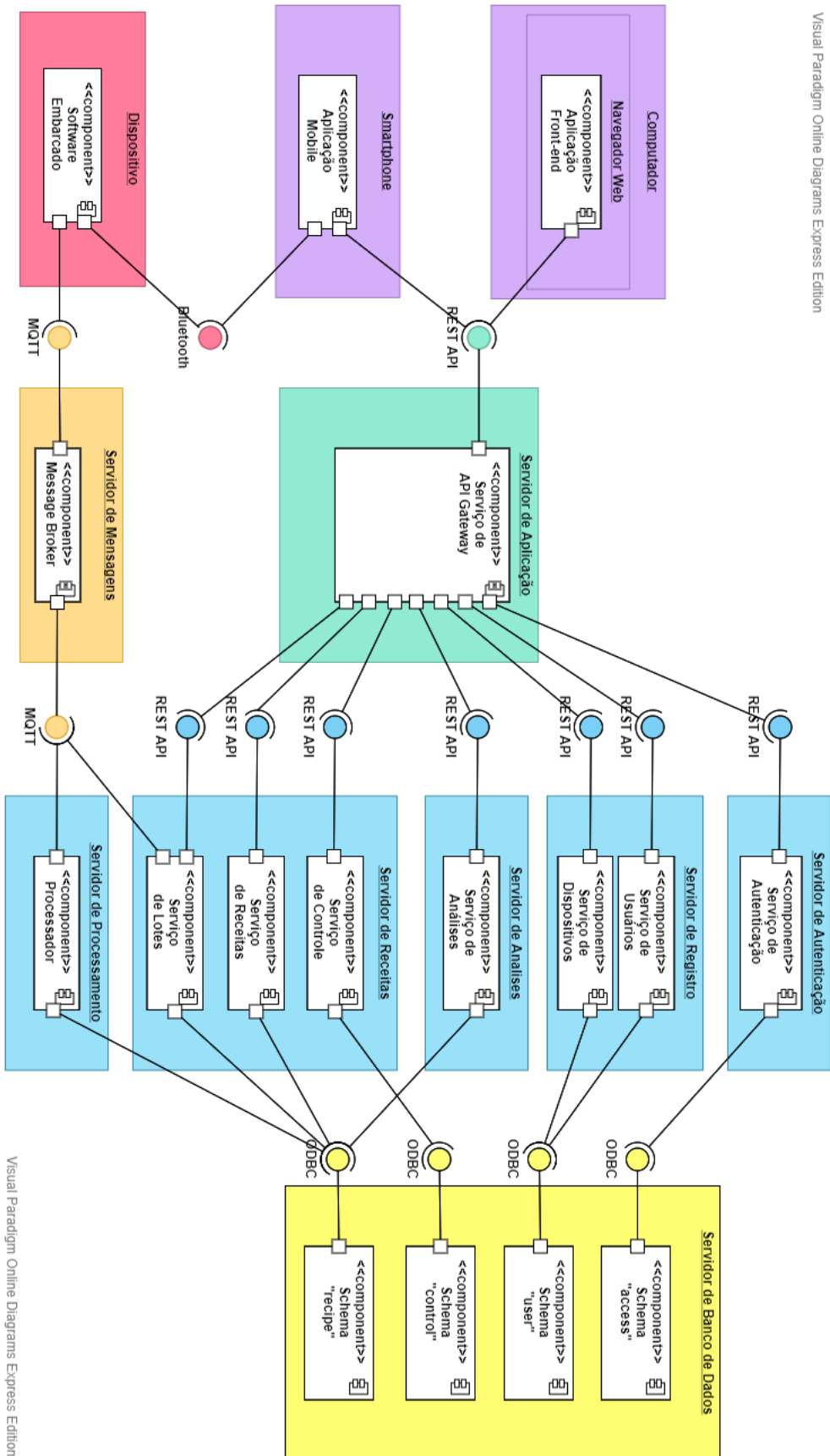


Figura 9: Diagrama de Componentes.

## 5 IMPLEMENTAÇÃO

## **6 TESTES E AVALIAÇÃO DO PROTÓTIPO**

## **7 CONSIDERAÇÕES FINAIS**

### **7.1 Conclusões do Projeto de Formatura**

### **7.2 Contribuições**

### **7.3 Perspectivas de Continuidade**

Testando referência bibliográfica como descrito em [1].

## REFERÊNCIAS

- [1] COCKERHAM, C. C.; ZENG, Z.-B. Design III With Marker Loci. *Genetics*, v. 143, p. 1437–1456, 1996.

## APÊNDICE A – PROTÓTIPOS DAS INTERFACES

## ANEXO A