

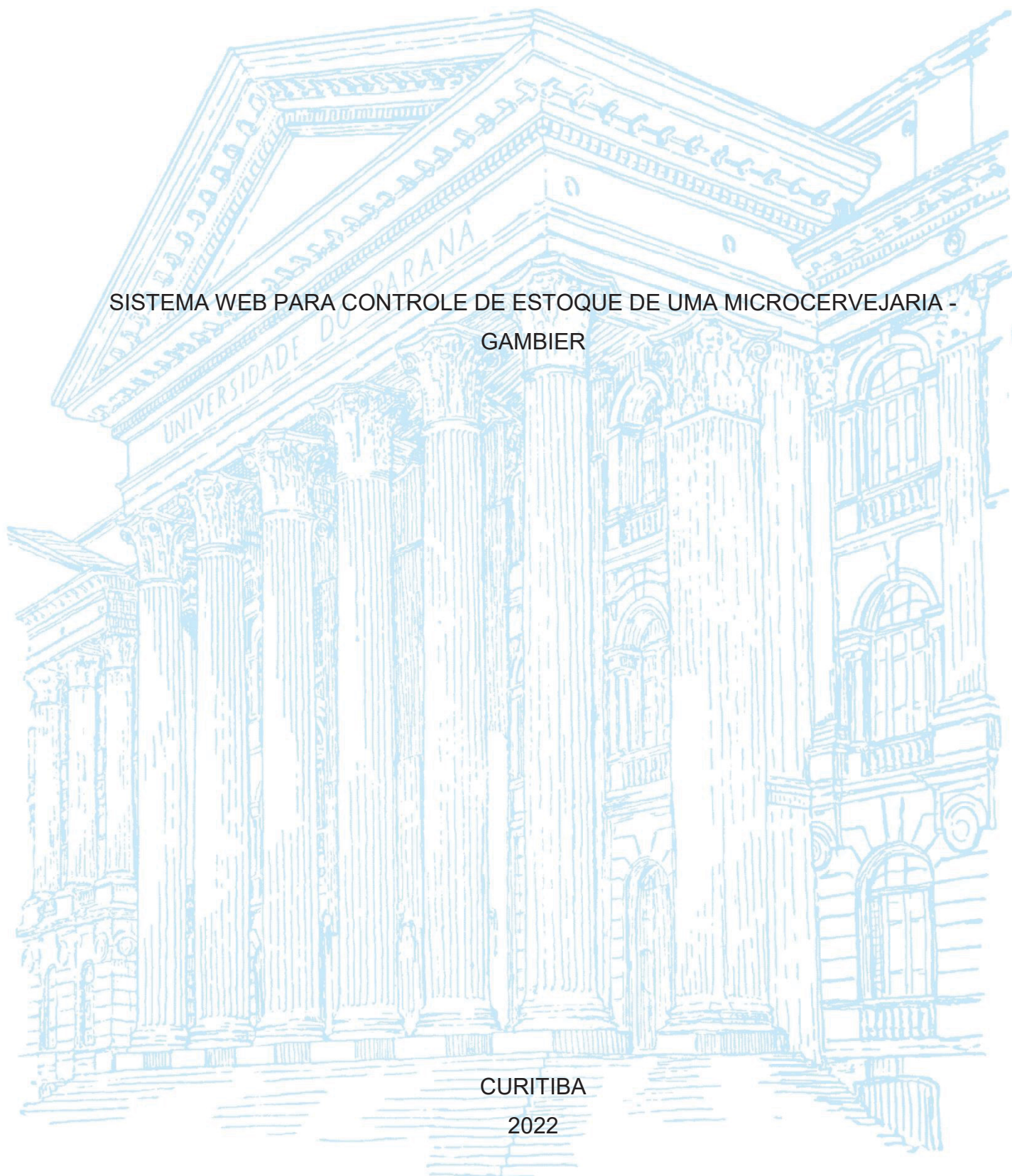
UNIVERSIDADE FEDERAL DO PARANÁ

LEVY CÉSAR BARBOSA SALES

SISTEMA WEB PARA CONTROLE DE ESTOQUE DE UMA MICROCERVEJARIA -  
GAMBIER

CURITIBA

2022



LEVY CÉSAR BARBOSA SALES

SISTEMA WEB PARA CONTROLE DE ESTOQUE DE UMA MICROCERVEJARIA -  
GAMBIER

Trabalho de Conclusão de Curso apresentado ao curso de Pós-Graduação em Engenharia de Software, Setor de Educação Profissional e Tecnológica da Universidade Federal do Paraná, Universidade Federal do Paraná, como requisito parcial à obtenção do título de Especialista em Engenharia de Software.

Orientador: Prof. Dr. Razer Anthom Nizer Rojas Montaña

CURITIBA

2022



MINISTÉRIO DA EDUCAÇÃO  
SETOR DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA  
UNIVERSIDADE FEDERAL DO PARANÁ  
PRÓ-REITORIA DE PESQUISA E PÓS-GRADUAÇÃO  
CURSO DE PÓS-GRADUAÇÃO ENGENHARIA DE  
SOFTWARE - 40001016231E1

## TERMO DE APROVAÇÃO

Os membros da Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação ENGENHARIA DE SOFTWARE da Universidade Federal do Paraná foram convocados para realizar a arguição da Monografia de Especialização de **LEVY CÉSAR BARBOSA SALES** intitulada: **SISTEMA WEB PARA CONTROLE DE ESTOQUE DE UMA MICROCERVEJARIA - GAMBIER**, que após terem inquirido o aluno e realizada a avaliação do trabalho, são de parecer pela sua APROVAÇÃO no rito de defesa. A outorga do título de especialista está sujeita à homologação pelo colegiado, ao atendimento de todas as indicações e correções solicitadas pela banca e ao pleno atendimento das demandas regimentais do Programa de Pós-Graduação.

Curitiba, 18 de Novembro de 2022.

  
RAZIR ANTON NIZER ROJAS MONTANO  
Presidente da Banca Examinadora

  
JAIME WOJCIECHOWSKI  
Avaliador Interno (UNIVERSIDADE FEDERAL DO PARANÁ)

Dedico este trabalho à todos aqueles que me ajudaram nessa trajetória,  
esposa, família, amigos, colegas e corpo docente da UFPR.

## **AGRADECIMENTOS**

Agradeço primeiramente à Deus por me capacitar a chegar até aqui. Ao meu professor orientador Dr. Razer Montañó pela disposição e orientação durante o desenvolvimento deste trabalho.

Agradeço à minha família e amigos que sempre me apoiaram nos desafios que a vida me proporcionou.

Por fim, um agradecimento especial ao corpo docente do curso de Especialização em Engenharia de Software da Universidade Federal do Paraná pelos conhecimentos compartilhados durante o curso e a disposição para auxiliar em todos os processos e etapas dessa formação.

“A persistência é o menor caminho do êxito. ”

(Charles Chaplin)

## RESUMO

Considerando o início de um novo negócio na família, decidiu-se iniciar este projeto com a intenção de prover um sistema completo e específico para o controle de estoque de uma microcervejaria e promover profissionalização organizacional em uma microempresa em seus estágios iniciais. O desenvolvimento do sistema foi com base em uma arquitetura de API REST no *backend* para o gerenciamento e disponibilização dos dados. Por se tratar de um sistema web, foi utilizado JavaScript juntamente com o React para a interação com usuário final e consumo da API REST. Além das tecnologias, o projeto foi desenvolvido com a aplicação dos princípios de desenvolvimento ágil de software, através do Scrum. Com base na definição dos requisitos e propósitos iniciais do sistema, o objetivo foi alcançado ao fim deste trabalho.

Palavras-chave: api rest, java, javascript, microcervejaria, controle de estoque, sistema web

## **ABSTRACT**

Considering the start of a new business in the family, it was decided to start this project with the intention of providing a complete and specific system for stock control of a microbrewery and promoting organizational professionalization in a microenterprise in its early stages. The development of the system was based on a REST API architecture in the backend for data management and availability. As it is a web system, JavaScript was used together with React to interact with the end user and consume the Rest API. In addition to the technologies, the project was developed with the application of agile software development principles, through Scrum. Based on the definition of the initial requirements and purposes of the system, the objective was reached at the end of this work.

Keywords: api rest, java, javascript, microbrewery, inventory management, web application

## LISTA DE FIGURAS

FIGURA 1 – Arquitetura REST .....	24
FIGURA 2 - Níveis de maturidade REST .....	25
FIGURA 3 - Planejamento das Sprints de desenvolvimento. ....	28
FIGURA 4 - Arquitetura do Sistema .....	38
FIGURA 5 - Tela de Autenticação no Sistema .....	39
FIGURA 6 - Tela inicial do Sistema – Parte Superior.....	39
FIGURA 7 - Tela Inicial - Parte Inferior.....	40
FIGURA 8 - Tela principal de Cerveja .....	41
FIGURA 9 - Modal de cadastro/edição de Cerveja .....	41
FIGURA 10 - Tela principal de Tipos de Cerveja .....	41
FIGURA 11 - Modal de cadastro/edição de Tipos de Cerveja.....	42
FIGURA 12 - Tela Principal de Clientes .....	42
FIGURA 13 - Modal de cadastro/edição de Clientes.....	43
FIGURA 14 - Tela principal de Pedidos .....	44
FIGURA 15 - Modal de lançamento de Pedidos.....	44
FIGURA 16 - E-mail de alerta de estoque baixo .....	45
FIGURA 17 - Modal de detalhes do Pedido / Alteração de Status .....	46
FIGURA 18 - Modal de detalhes do Pedido / Alteração de Status (2).....	46
FIGURA 19 - E-mail de confirmação de pedido .....	47
FIGURA 20 - Tela principal de Produção .....	48
FIGURA 21 - Modal de lançamento de Produção .....	48
FIGURA 22 - E-mail informativo de lançamento de Produção. ....	49
FIGURA 23 - Modal de detalhes da Produção .....	50
FIGURA 24 - Tela de Cervejas em estoque.....	50
FIGURA 25 - Tela de Cervejas em estoque com filtro ativo .....	51
FIGURA 26 - Tela de emissão de relatório de vendas diárias .....	51
FIGURA 27 - Relatório de vendas diárias .....	52
FIGURA 28 - Tela de emissão de relatório de vendas .....	53
FIGURA 29 - Relatório de vendas por cerveja .....	53
FIGURA 30 - Relatório de vendas por cliente .....	54
FIGURA 31 – Tela principal de usuário.....	55
FIGURA 32 - Modal de cadastro de novo usuário.....	55

FIGURA 33 - Modal de edição do usuário.....	56
FIGURA 34 - Modal de alteração de senha.....	56
FIGURA 35 - História de Usuário “Manutenção de Cerveja” .....	61
FIGURA 36 - História de Usuário "Manutenção de Tipos de Cerveja" .....	62
FIGURA 37 - História de Usuário "Manutenção de Clientes" .....	63
FIGURA 38 - História de Usuário "Manutenção de Pedido" .....	64
FIGURA 39 - História de Usuário "Manutenção de Produção" .....	65
FIGURA 40 - História de Usuário "Visualizar Estoque" .....	66
FIGURA 41 - História de Usuário "Gerar Relatório de Vendas Diárias" .....	66
FIGURA 42 - História de Usuário "Gerar Relatório de Vendas" .....	67
FIGURA 43 - Diagrama de Casos de Uso.....	68
FIGURA 44 - Diagrama de classes do sistema .....	69
FIGURA 45 - Diagrama de Sequência Login .....	70
FIGURA 46 - Diagrama de Sequência Alteração de Senha.....	71
FIGURA 47 - Diagrama de Sequência CRUD de Usuário.....	72
FIGURA 48 - Diagrama Sequência CRUD de Cerveja.....	73
FIGURA 49 - Diagrama de Sequência CRUD de Tipos de Cerveja .....	74
FIGURA 50 - Diagrama de Sequência CRUD de Cliente .....	75
FIGURA 51 - Diagrama de Sequencia Fluxo de Pedidos.....	76
FIGURA 52 - Diagrama de Sequência Alterar Status de Pedido.....	76
FIGURA 53 - Diagrama de Sequência de Fluxo de Produção .....	77
FIGURA 54 - Diagrama de Sequência Listar Estoque .....	77
FIGURA 55 - Diagrama de Sequência Emissão de Relatórios.....	78
FIGURA 56 - Diagrama físico de banco de dados .....	79
FIGURA 57 - Recursos disponíveis entidade ‘Cerveja’ .....	80
FIGURA 58 - Recursos disponíveis entidade ‘Tipos de Cerveja’ .....	80
FIGURA 59 - Recursos disponíveis entidade ‘Cliente’ .....	80
FIGURA 60 - Recursos disponíveis entidade ‘Usuário’ .....	81
FIGURA 61 - Recursos disponíveis entidade ‘Estado’ .....	81
FIGURA 62 - Recursos disponíveis entidade ‘Cidade’ .....	81
FIGURA 63 - Recursos disponíveis entidade ‘Pedido’ .....	81
FIGURA 64 - Recursos disponíveis fluxo de status de pedido .....	82
FIGURA 65 - Recursos disponíveis entidade ‘Produção’ .....	82
FIGURA 66 - Recursos disponíveis ‘Estoque’ .....	82

FIGURA 67 - Recursos disponíveis <i>dashboard/home page</i> .....	82
FIGURA 68 - Recursos disponíveis relatórios .....	83

## LISTA DE QUADROS

QUADRO 1 – Materiais e Métodos .....	31
--------------------------------------	----

## LISTA DE ABREVIATURAS OU SIGLAS

API	- Application programming interface
IDE	- Ambiente de Desenvolvimento Integrado
REST	- Representational State Transfer
SQL	- Structured Query Language
UML	- Unified Modeling Language
HATEOAS	- Hypermedia as the Engine of Application State

## SUMÁRIO

<b>1 INTRODUÇÃO</b> .....	<b>16</b>
1.1 JUSTIFICATIVA .....	17
1.2 OBJETIVOS .....	18
1.2.1 Objetivos específicos.....	18
1.3 ESTRUTURA DO DOCUMENTO.....	18
<b>2 FUNDAMENTAÇÃO TEÓRICA</b> .....	<b>20</b>
2.1 FUNDAMENTAÇÃO TEÓRICA DAS TECNOLOGIAS UTILIZADAS .....	21
2.1.1 <i>Unified Modeling Language</i> – UML .....	21
2.1.2 Java.....	22
2.1.3 Spring Boot.....	22
2.1.4 JavaScript.....	23
2.1.5 React.....	23
2.1.6 MySQL .....	23
2.1.7 API Rest .....	24
2.1.8 Desenvolvimento Ágil de Software .....	26
<b>3 MATERIAIS E MÉTODOS</b> .....	<b>28</b>
3.1 LINGUAGEM DE PROGRAMAÇÃO .....	29
3.2 BANCO DE DADOS.....	29
3.3 FLUXO DE TRABALHO .....	29
3.4 LISTA DE MATERIAIS .....	31
3.5 DESENVOLVIMENTO DO PROJETO .....	32
3.5.1 <i>Sprint</i> 1 .....	32
3.5.2 <i>Sprint</i> 2 .....	34
3.5.3 <i>Sprint</i> 3.....	34
3.5.4 <i>Sprint</i> 4 .....	35
3.5.5 <i>Sprint</i> 5.....	35
<b>4 APRESENTAÇÃO DOS RESULTADOS</b> .....	<b>37</b>
4.1 ARQUITETURA DO SISTEMA.....	37
4.2 SOFTWARE .....	38
<b>5. CONSIDERAÇÕES FINAIS</b> .....	<b>57</b>
5.1 RECOMENDAÇÕES PARA TRABALHOS FUTUROS.....	57
<b>REFERÊNCIAS</b> .....	<b>58</b>

<b>APÊNDICE 1 – HISTÓRIAS DE USUÁRIO .....</b>	<b>61</b>
<b>APÊNDICE 2 – DIAGRAMA DE CASOS DE USO .....</b>	<b>68</b>
<b>APÊNDICE 3 – DIAGRAMA DE CLASSES.....</b>	<b>69</b>
<b>APÊNDICE 4 – DIAGRAMAS DE SEQUÊNCIA.....</b>	<b>70</b>
<b>APÊNDICE 5 - DIAGRAMA FÍSICO DE BANCO DE DADOS .....</b>	<b>79</b>
<b>APÊNDICE 6 – DOCUMENTAÇÃO API REST (<i>BACKEND</i>) .....</b>	<b>80</b>

## 1 INTRODUÇÃO

Mesmo sem registros ou data que comprove sua origem, a cerveja é considerada uma das bebidas mais antigas do mundo, porém somente no início do século XIX, com a chegada da família real ao Brasil, é que houve o surgimento do setor no país, sendo que o surgimento das primeiras fábricas aconteceu somente na metade deste século, quando também surgiram as primeiras marcas de cerveja nacionais (MARTINS e col. , 2018).

Em um cenário mais contemporâneo, uma pesquisa da Kirin Beer University divulgada no anuário de 2015 da Associação Brasileira da Indústria da Cerveja (CervBrasil) cita o Brasil como o 3º maior produtor de cerveja do mundo e segue com crescimento considerável – segundo o Sistema de Controle de Produção de Bebidas da Receita Federal a produção nacional cresceu 64% entre 2005 e 2014 (SEBRAE, 2017).

Segundo Marcusso (2015), nas últimas décadas notou-se em todo o mundo uma grande expansão no número de cervejarias, impulsionado principalmente pela abertura de um novo segmento de mercado, as pequenas cervejarias, comumente chamadas de microcervejarias. No Brasil, o crescimento do mercado de microcervejarias se deu na década de 1990 devido à ações dos governos Collor e FHC e da maior divulgação da cultura cervejeira.

A demanda por cervejas especiais ou artesanais, produtos oriundos das microcervejarias, é influenciada pelo desejo de diferenciação e o consumo deste tipo de cerveja tem a preferência dos consumidores pela qualidade, diversidade de ingredientes e sabores, mais do que o preço em si. O aumento no consumo é explicado pelo aumento do poder aquisitivo da população, melhoria da distribuição de renda e a sofisticação do padrão de consumo (MARCUSO, 2015).

A alteração no padrão de consumo de cerveja foi confirmada quando, em 2010, pela primeira vez houve desaceleração do crescimento das cervejas populares, enquanto as microcervejarias mostravam crescimento (MARCUSO, 2015).

Sendo assim, apesar de que em 2014, segundo a Abrape, as microcervejarias representavam apenas 1% do setor no Brasil, há uma forte tendência de crescimento, principalmente pela valorização das cervejas artesanais pelos consumidores (SEBRAE, 2017).

Dessa forma, o objetivo deste trabalho é desenvolver uma aplicação web de controle e organização de estoque para uma microcervejaria, contemplando desde os aspectos diretamente ligados ao gerenciamento de estoque e vendas, assim como cadastro e manutenção de clientes e visão geral financeira do negócio.

## 1.1 JUSTIFICATIVA

Com o iminente crescimento do mercado no país, cada vez mais é necessário que as microcervejarias se organizem de forma que o negócio se torne o mais sustentável e lucrativo.

Em geral, as micro e pequenas empresas nacionais, onde estão inclusas as microcervejarias, não conseguem visualizar a importância de uma gestão de estoque.

Tratando de cerveja, um produto perecível, é notório que o produtor seja capaz de identificar o fluxo de consumo para ajustar seu fluxo de produção, evitando desperdícios e gastos desnecessários. É neste ponto que a gestão de estoque pode ser fundamental para garantir que a empresa possa melhorar sua cadeia de produção e com isso, reduzir os custos da mesma.

Oliveira e col. (2016) salientam que, um estoque adequado à demanda irá diminuir entre outros fatores, o custo da armazenagem, tópico este de suma importância quando se trata de uma microcervejaria.

Na perspectiva administrativa de um negócio, a gestão de estoque é um elemento imprescindível e pode ser crucial para uma boa gestão, obtendo-se redução de custos e reduzindo imprevistos, facilitando todo e qualquer planejamento da organização. Além de se implantar uma gestão de estoque, a mesma deve ser acompanhada para que possam trazer resultados positivos à empresa. (OLIVEIRA e col. 2016)

## 1.2 OBJETIVOS

Com base nos pontos apresentados anteriormente sobre a importância de se ter controle sob o estoque em um negócio, a seguir são descritos os objetivos gerais e específicos deste projeto.

Desenvolver uma aplicação web que servirá como controle de estoque para a microcervejaria Gambier.

### 1.2.1 Objetivos específicos

Os objetivos específicos do sistema são:

- a) Permitir o cadastro de cervejas por tipo;
- b) Permitir a consulta do estoque de cada cerveja cadastrada;
- c) Permitir o lançamento de entrada de estoque (produção) e saída de estoque (pedido/venda)
- d) Realizar e visualizar pedidos;
- e) Permitir login apenas de usuários permitidos e cadastrados;
- f) Enviar e-mails de estoque baixo, conforme definido pelo usuário;
- g) Enviar e-mails conforme pedidos são criados ou seus status são alterados;
- h) Dashboard em tela inicial com dados de estoque, fluxo de pedidos e outras informações relevantes;
- i) Permitir a emissão de relatórios periódicos de pedidos e entrada do estoque.

## 1.3 ESTRUTURA DO DOCUMENTO

Neste tópico do trabalho, a estrutura e as informações do desenvolvimento do projeto estão detalhadas a seguir.

O Capítulo 2 visa detalhar a fundamentação teórica do sistema, listando os pontos principais e visão aprofundada dos temas relevantes para a elaboração do mesmo. Além de aprofundar no assunto de controle de estoque, assuntos técnicos tais como a descrição das tecnologias utilizadas para desenvolvimento são abordados.

O Capítulo 3, Materiais e Métodos, possui a descrição da metodologia utilizada no desenvolvimento do sistema, incluindo as tecnologias utilizadas, cronograma de atividades e descrição das mesmas.

No Capítulo 4 o sistema contém o detalhamento e apresentação do sistema, contemplando as telas e todas as funcionalidades implementadas.

O Capítulo 5 trata da conclusão e considerações finais do trabalho, avaliando se o objetivo inicial foi cumprido e funcionalidades futuras.

Ao final do documento se encontram os apêndices e a documentação técnica do sistema.

## 2 FUNDAMENTAÇÃO TEÓRICA

A prática de fabricação de cervejas artesanais no país é recente, menciona Ferreira (2011) e a ideia de apreço não só por beber uma boa cerveja, como também em fazê-la acaba originando as cervejarias artesanais e também aos cervejeiros caseiros, denominados de *homebrewers* (MORADO, 2009).

Com a valorização deste produto, muitos fabricantes caseiros se viram com a oportunidade para abertura de microcervejarias, que têm crescido no mercado, prezando pelo refinamento do produto e variedade, mesmo que o público alvo seja mais seletivo e que prezam pela qualidade da cerveja. (SILVA, 2017)

De acordo com Novaes (2011), não importa o tipo de empresa, o importante é que os estoques precisam ser controlados de forma que a empresa consiga desenvolver melhor suas atividades.

Martelli e Dandaro (2015), afirmam que o gerenciamento de estoque surgiu para suprir a necessidade das empresas em controlar todo o fluxo de materiais, como o período de cada em armazenamento, a quantidade mantida, a periodicidade de reposição, entre outros.

Borges e col. (2010), definem que um bom gerenciamento de estoques ajuda na redução dos valores monetários envolvidos, de forma a mantê-los os mais baixos possíveis, mas dentro dos níveis de segurança e dos volumes para o atendimento da demanda.

Na visão de Ballou (2001), um dos objetivos primários do controle de estoque é garantir que o produto esteja disponível no tempo e nas quantidades necessárias. O autor ainda complementa que, administrar cuidadosamente o nível de estoque é economicamente sensato.

Jó (2012), define que o estoque e a manutenção do mesmo se tornam uma vantagem competitiva na rapidez da distribuição, sendo de suma importância em vários pontos: melhorar o atendimento ao cliente, economia, contra incerteza na demanda e tempo de entrega.

A gestão de estoques é uma prática de cunho estratégico que visa aproveitar melhor os recursos e aumentar os resultados positivos, gerando reflexos significativos na eficiência operacional e financeira de uma empresa. Dessa forma, a gestão de estoques tem reflexos diretos na eficiência e no financeiro de uma empresa, sendo necessário para melhorar a eficiência no atendimento à demanda. (JÓ, 2012).

De forma sucinta, a gestão de estoques é considerada elemento fundamental para a redução e o controle dos custos totais das empresas (OSÓRIO, 2009).

## 2.1 FUNDAMENTAÇÃO TEÓRICA DAS TECNOLOGIAS UTILIZADAS

Nessa seção estão listadas as tecnologias utilizadas durante o desenvolvimento do sistema de controle de estoque para a microcervejaria Gambier.

### 2.1.1 *Unified Modeling Language* – UML

A UML é uma linguagem gráfica para visualização, especificação, construção e documentação de artefatos de sistemas complexos de software. Na UML, é utilizado uma forma padrão para a preparação de planos de arquitetura de projetos de sistema (RUMBAUGH e col., 2005).

A UML é adequada para a modelagem de sistemas, sendo uma linguagem muito expressiva, abrangendo todas as visões necessárias ao desenvolvimento e implantação de um sistema. Por se tratar apenas de uma linguagem, a UML é somente parte de um método para desenvolvimento de software, sendo usada independente do processo, apesar de ser perfeitamente utilizada em processos orientados à casos de uso, centrado na arquitetura, iterativo e incremental (RUMBAUGH e col., 2005).

Rumbaugh e col. (2005), define os blocos de construção da UML em três tipos:

1. Itens;
2. Relacionamentos;
3. Diagramas.

Segundo os autores, os itens são as abstrações identificadas como cidadãos de primeira classe em um modelo, sendo que os relacionamentos reúnem estes itens e os diagramas agrupam as coleções interessantes de itens.

Dos três tipos listados anteriormente, a apresentação gráfica de um conjunto de elementos (itens) e seus relacionamentos, é um diagrama.

Os diagramas são desenhados para permitir a visualização de um sistema sob diferentes perspectivas, constituindo uma projeção de um determinado sistema, representando uma visão parcial dos elementos que o compõem (RUMBAUGH e col., 2005).

### 2.1.2 Java

O Java foi criado no início dos anos 90 em um projeto financiado pela Sun Microsystems e chefiado por James Gosling (DEITEL, 2016).

O principal objetivo do Java é a capacidade de sua utilização para escrever programas que podem ser executados em uma grande variedade de sistemas computacionais e dispositivos controlados por computador, popularmente chamado de “escreve uma vez, execute em qualquer lugar” (DEITEL, 2016)

Atualmente, o Java é utilizado para desenvolver aplicativos corporativos de grande porte, desenvolver funcionalidades para servidores web (os conteúdos vistos nos navegadores são fornecidos por um computador) e em aplicativos voltados ao consumo popular (por exemplo em smartphones, tvs, etc.). Em outras palavras, o Java é amplamente utilizado para implementar aplicativos e softwares baseados na internet para todos os dispositivos que se comunicam através de uma rede (DEITEL, 2016)

### 2.1.3 Spring Boot

O Spring facilita a construção de aplicações Java, provendo o que é necessário para cobrir o desenvolvimento em Java em um ambiente empresarial.

Spring é um *framework* dividido em módulos, onde as aplicações que definem quais os módulos serão utilizados. Dentro desses módulos, no contêiner principal, há os arquivos de configuração e o mecanismo de injeção de dependência. Sendo assim, o Spring suporta diferentes arquiteturas, incluindo mensageria, dados e comunicação com banco de dados e web, incluindo o Spring MVC (arquitetura dividida em: *model*, *view* e *controller*) baseado em servlets (SPRING FRAMEWORK, 2022).

No contexto web, há o Spring Boot que permite a construção de aplicações web de forma rápida e descomplicada, removendo muito do código *boilerplate* e de configuração. (SPRING, 2021)

O Spring Boot fornece um modelo de programação web moderno que agiliza o desenvolvimento das aplicações do lado do servidor, APIs REST, entre outros.

#### 2.1.4 JavaScript

Segundo Flanagan (2011), JavaScript é a linguagem mais onipresente da história, considerando que a maioria dos sites modernos e todos os outros navegadores modernos possuem interpretadores JavaScript. A linguagem faz parte da tríade de tecnologias que os desenvolvedores web devem conhecer, juntamente com HTML, utilizado para especificar o conteúdo de páginas web e CSS, para especificar a apresentação dessas páginas (FLANAGAN, 2011).

#### 2.1.5 React

O React é uma biblioteca JavaScript utilizada para auxiliar e facilitar a criação das interfaces de usuário.

Baseado em componentes, permite que eles sejam criados encapsulados e assim gerenciem o seu próprio estado e combinados, assim sendo utilizados para criar interfaces mais complexas. Toda a lógica de programação destes componentes, é feita inteiramente em JavaScript (REACT, 2021).

A biblioteca React é feita para ser usado conforme necessidade, ou seja, você pode usar menos ou mais da biblioteca de acordo com o que a interface que está sendo criada necessita (REACT, 2021).

#### 2.1.6 MySQL

O MySQL é um servidor e gerenciador de banco de dados (SGDB) relacional que teve origem na década de 90, criado pelos desenvolvedores David Axmark, Allan Larsson e Michael Widenius (MILANI, 2006).

O MySQL tem alto poder de execução e de armazenamento, sendo possível a execução de mais de um bilhão de consultas por dia em um site, ou até mesmo realizar o processamento de milhares de transações por minuto. (MILANI, 2006)

### 2.1.7 API Rest

Para prover os dados para que uma aplicação funcione adequadamente, são desenvolvidas API REST.

A definição de REST é um conjunto de princípios para a utilização dos protocolos web, utilizando a arquitetura web, representada na FIGURA 1, em prol da aplicação.

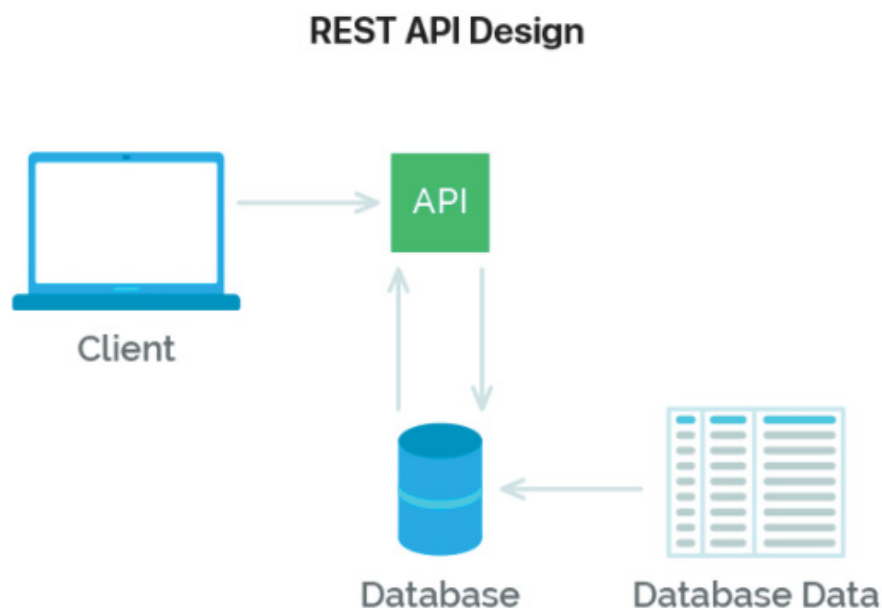
Baseado no protocolo HTTP, o REST utiliza de diversos métodos HTTP para acessar os recursos que expõe os dados para o consumo de uma aplicação. Entre os principais métodos, os mais comuns e utilizados são GET, POST, PUT, DELETE, PATCH (TILKOV, 2008).

Em uma chamada HTTP, há sempre na resposta um código de status para indicar se a requisição HTTP foi concluída corretamente. Estes status são definidos pela seção 10 da RFC 2616 (MOZILLA, 2022).

Essas respostas são agrupadas em cinco divisões:

- Respostas de informação – 100 a 199;
- Resposta de sucesso – 200 a 299;
- Redirecionamentos – 300 a 399;
- Erros de servidor – 500 a 599.

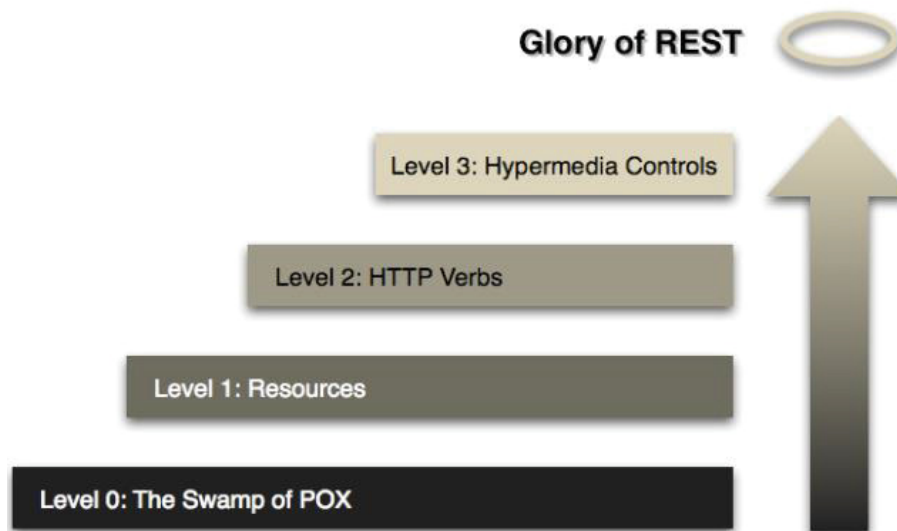
FIGURA 1 – Arquitetura REST



FONTE: Alifyz F. Pires (2018).

Em se tratando de APIs REST, Leonard Richardson propôs um modelo de maturidade baseado em quatro níveis (FIGURA 2) para que uma API seja considerada REST (FOWLER, 2010).

FIGURA 2 - Níveis de maturidade REST



FONTE: FOWLER (2010).

O modelo de design do Nível 0 é o nível mais distante do que de fato o REST propõe. Neste modelo, não há gerenciamento dos códigos de resposta HTTP, sendo comum em aplicações deste nível de maturidade, códigos incoerentes com o resultado da requisição. Como exemplo, uma consulta *get* em uma lista de objetos onde nenhum objeto é encontrado, porém o código HTTP retornado é um 200 OK. Desta forma, a aplicação impede que elementos de *gateway* e *proxy* atuem de forma adequada.

No segundo nível, o nível dos Recursos, a API é modelada e organizada tendo os recursos como referência, sendo que, neste nível, não é necessário conhecer a funcionalidade de cada requisição/método, apenas conhecer o recurso, que se trata do objeto mapeado (cerveja, cliente, por exemplo), a que se tem acesso.

Como exemplo, no nível 0 haveria um *get* em */buscarCerveja* com a ação de visualizar/listar a cerveja, já no nível 1 de maturidade, esse mesmo *get* seria representado apenas por */cerveja*.

O terceiro nível, resolve um problema citado e encontrado no nível 0. O nível de maturidade Verbos HTTP, altera o papel do HTTP em si, que deixa de exercer um papel apenas de transporte para também exercer um papel semântico na API. Simplificando, neste nível, os verbos HTTP passam a ser utilizados com o propósito no qual foram criados. Utilizando o mesmo exemplo do nível 0, neste nível de maturidade teríamos a resposta *404-Not Found*, que de fato é a mais adequada neste cenário.

O quarto e último nível, chamado de HATEOAS, "*Hypermedia As the Engine Of Application State*", tem como principal elemento a representação *hypermedia*, que permite um documento ou requisição apresentar seu estado atual e quais os seus relacionamentos com outros futuros estados. Para ilustrar, em uma requisição do tipo *get* em */cervejas*, onde o retorno é uma única cerveja, com o HATEOAS haveria além da representação do objeto cerveja, a representação *hypermedia* dos relacionamentos futuros, como deletar e editar.

### 2.1.8 Desenvolvimento Ágil de Software

Em 2001, um grupo de dezessete autores e representantes das metodologias ágeis, tais como: Scrum, Extreme Programming (XP), entre outros, discutiram um padrão de desenvolvimento de projetos dentre as técnicas e metodologias existentes e o resultado foi a criação do Manifesto para Desenvolvimento Ágil de Software (AGILE MANIFESTO, 2001).

Neste manifesto, foi estabelecido um *framework* comum para processos ágeis, contemplando os seguintes itens:

1. Indivíduos e interações mais que processos e ferramentas;
2. Software funcionando mais que documentação abrangente;
3. Colaboração do cliente mais que negociação de contrato;
4. Responder à mudança mais do que seguir um plano.

Quando se fala em desenvolvimento ágil de software, entre diversos modelos e meios de se aplicar, o Scrum é um dos mais populares e mais utilizados ao redor do mundo.

O Manifesto Ágil defende que responder as mudanças é mais importante do que seguir um plano, sendo assim, o Scrum encara as mudanças como parte natural

do processo de desenvolvimento, dessa forma, novas funcionalidades solicitadas pelo cliente, por exemplo, podem ser introduzidas no próximo *sprint* (MACHADO; MEDINA, 2009).

Vieira (2014) descreve o Scrum como um *framework* para organizar e gerenciar trabalhos complexos, sendo um conjunto de valores, princípios e práticas que fornecem a base para que se adicione práticas particulares de gestão à organização e que sejam relevantes para a empresa.

O processo do Scrum se inicia com a visão do produto/software a ser desenvolvido, sendo que nesta etapa, devem ser estabelecidos uma lista de requisitos que representam essa visão, sendo que após essa definição, é há a ordenação dos processos de desenvolvimento, conforme prioridade. (SCHWABER, 2004)

O processo do desenvolvimento é feito por meio da divisão por *sprints*, que representam um intervalo pré-definido de dias, geralmente 30 dias, onde em cada *sprint* há a entrega de uma ou mais funcionalidades do software. Por convenção, durante as *sprints*, a equipe se reúne diariamente para o scrum diário, onde cada membro da equipe deve comentar o que foi feito desde o último scrum diário, o que fará até o próximo e se há algum impedimento em prosseguir com suas atividades. (SCHWABER, 2004)

### 3 MATERIAIS E MÉTODOS

Neste capítulo são abordadas como as tecnologias e métodos, descritas no capítulo anterior, foram utilizadas para o desenvolvimento e implementação do software.

Por não se tratar de uma equipe de desenvolvimento, algumas ferramentas do Scrum, entre elas *daily meeting*, retrospectiva, etc, não foram aplicadas no planejamento e desenvolvimento deste sistema.

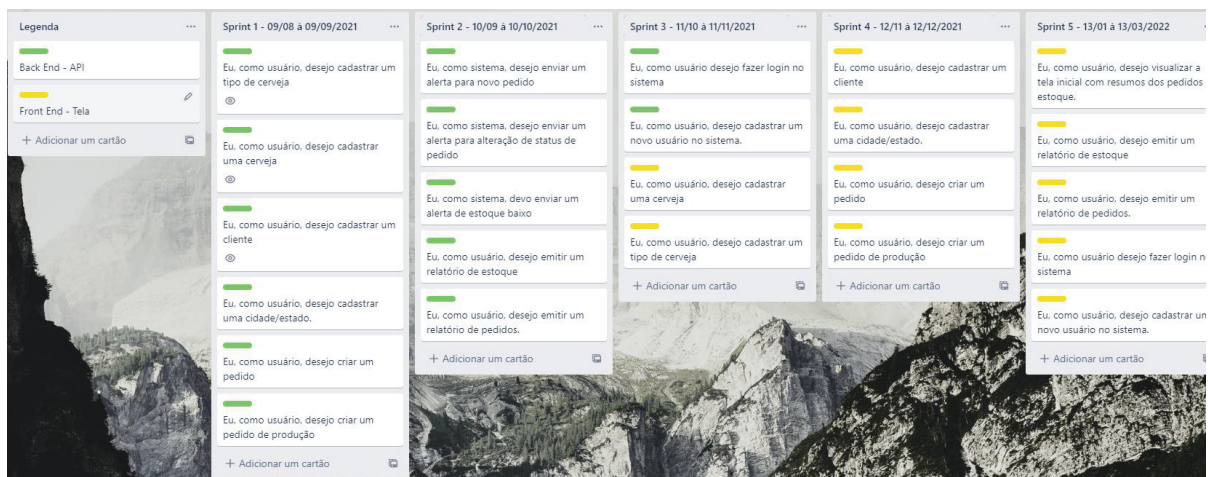
A metodologia Scrum foi aplicada no planejamento do projeto, utilizando a ferramenta web *Trello*, que se trata de um gerenciador de tarefas em listas em que nele foi possível adicionar todas as tarefas de *backlog* à serem desenvolvidas, e posteriormente planejar em *sprints* todas as etapas do projeto, e com isso, acompanhar o progresso e redefinir prioridades, ajustes de funcionalidade, etc. Esse planejamento, pode ser visto com detalhes na FIGURA 3.

Diferente do que se define inicialmente no Scrum, nem todas as sprints definidas no projeto geram um produto ou *feature* já para produção. No caso deste sistema, as sprints iniciaram com todo o *backlog* do *backend* e as sprints finais trataram do *frontend*, onde é finalizado o “produto” em si.

Diferente do habitual quando se utiliza Scrum (*sprints* mais curtas), por se tratar de somente um desenvolvedor, foi definida um mês de duração para cada *sprint*.

Com um total de cinco *sprints*, as tarefas das *sprints* foram divididas entre *backend* (API REST) e *frontend* (telas e interação com usuário final), conforme legenda.

FIGURA 3 - Planejamento das Sprints de desenvolvimento.



FONTE: O autor (2022).

### 3.1 LINGUAGEM DE PROGRAMAÇÃO

Para o *backend* (API REST) foi utilizado Java em conjunto com o *framework* Spring Boot, na versão 2.5.3. A linguagem especificamente compõe grande parte do mercado e tem uma comunidade ativa, o que facilita a busca e obtenção de matérias de apoio na internet.

O Spring Boot atualmente é o *framework* para desenvolvimento web mais popular e além disso, consiste em um ecossistema completo, oferecendo soluções em segurança, acesso e persistência ao banco de dados, entre outras.

Antes da implementação, foram criados diagramas para facilitar e ajudar no desenvolvimento do sistema, entre eles, o diagrama de classes (APÊNDICE 1) e os diagramas de sequência (APÊNDICE 3).

Para o *frontend*, que será responsável pela interface com o usuário e o consumo da API REST desenvolvida no *backend*, foi utilizado a linguagem JavaScript na versão 16.14.2 juntamente da biblioteca React, utilizando a versão 17.0.2.

O JavaScript está presente em todos os sistemas web modernos e o React é uma das bibliotecas mais populares e mais utilizadas da linguagem. Além das já citados, para definir os elementos da página e estilizá-los, foi utilizado HTML, CSS e algumas bibliotecas de CSS, como Bootstrap.

### 3.2 BANCO DE DADOS

Devido à sua facilidade de operação e grande utilização no mercado, o banco de dados relacional MySQL foi utilizado para o armazenamento dos dados de todo o software.

Para a arquitetura do banco, suas tabelas e detalhes, foi criado um diagrama relacional de dados (APÊNDICE 2), utilizando o software MySQL Workbench, versão 8.0, disponibilizado atualmente pela Oracle e que se trata de uma IDE para facilitar o gerenciamento, criação e operação de um banco de dados.

### 3.3 FLUXO DE TRABALHO

Para um software deste porte é necessário que exista um versionamento de código, e para isso foi utilizado a ferramenta GIT, através da plataforma GitHub.

O versionamento foi feito de forma que, em cada tarefa realizada conforme as sprints, era criada uma nova *branch*, que posteriormente, quando finalizada, era criado um *pull request* para que o código fosse revisto e um *merge* realizado com a *branch master*.

Para a criação de *branchs* e *commits* no repositório principal, foi utilizado um padrão de nomenclatura para que facilmente fosse identificado o que foi realizado em cada tarefa.

A seguir, a nomenclatura utilizada e a descrição do uso de cada uma delas:

- *bugfix*: Correções de bugs que devem ser corrigidos assim que identificados;
- *feature*: Implementação de uma nova funcionalidade ao sistema;
- *hotfix*: Semelhante ao *bugfix*, porém não se trata de um erro, e é utilizada quando é necessário alguma correção, seja ela performance, refatoração, etc.

### 3.4 LISTA DE MATERIAIS

No QUADRO 1 é apresentado a lista dos materiais utilizados durante a confecção do projeto.

QUADRO 1 – Materiais e Métodos

Material	Descrição	Versão
Java	Linguagem de programação utilizada no backend	11.0.12
Spring Boot	Framework para a criação da API REST	2.5.3
JavaScript	Linguagem de programação utilizada no frontend	16.14.2
React	Biblioteca para a criação das interfaces com o usuário.	17.0.2
MySQL	Sistema gerenciador de banco de dados	-
Intellij	IDE utilizado para desenvolvimento do backend (API REST)	2021.2.3
Visual Studio Code	IDE utilizado para desenvolvimento do frontend	1.66.2
MySQL Workbench	IDE utilizado para o gerenciamento do banco de dados	8.0
Github	Ferramenta para controle de versão	-
Astah	IDE para confecção de diagramas	8.5
Notebook Dell G3	Dispositivo utilizado para desenvolvimento	-
Windows	Sistema operacional do dispositivo de desenvolvimento.	10 Pro

FONTE: O autor (2022).

### 3.5 DESENVOLVIMENTO DO PROJETO

Nesta seção é descrito como ocorreu o desenvolvimento do projeto, conforme *sprints* definidas previamente utilizando o Scrum como ferramenta de metodologia ágil.

Na fase inicial do desenvolvido, foram desenvolvidas as histórias de usuário, apresentadas no Apêndice 1, para facilitar e tornar o levantamento de requisitos do sistema mais dinâmico.

Todo o desenvolvimento do projeto, contido nas *sprints* a seguir, foram desenvolvidos seguindo os diagramas específicos de caso de uso, conforme Apêndice 3.

#### 3.5.1 *Sprint* 1

Na primeira fase do projeto foram desenvolvidos os diagramas de classe (APÊNDICE 4) e de casos de uso (APÊNDICE 2), tal como os requisitos e escopo do projeto, seguindo os fundamentos da UML.

Foi criado o projeto utilizando o Spring *Initializr* e o banco de dados utilizando o MySQL Workbench, conforme diagrama de banco de dados (APÊNDICE 6)

Ainda nesta fase, foram criados na API REST os principais serviços de CRUD do sistema, documentados no Apêndice 7 e detalhados abaixo:

Atividades da *Sprint* 1 – Tipos de Cerveja:

1. Cadastrar novo tipo de cerveja;
2. Remover tipo de cerveja;
3. Listar tipos de cerveja.
4. Atualizar tipos de cerveja;
5. Busca de tipo de cerveja por nome;
6. Busca de tipo de cerveja por Id.

Atividades da *Sprint* 1 – Cerveja:

1. Cadastrar nova cerveja;
2. Remover cerveja;
3. Listar cervejas;
4. Atualizar cerveja;

5. Busca de cerveja por nome;
6. Busca de cerveja por Id;
7. Busca de cerveja por tipo de cerveja.

Atividades da *Sprint 1* – Cliente:

1. Cadastrar novo cliente;
2. Remover cliente;
3. Listar clientes;
4. Atualizar cliente;
5. Busca de cliente por nome;
6. Busca de cliente por Id;
7. Busca de cliente por tipo de cliente;
8. Cadastro de endereço (CRUD Cidade e Estado);

Atividades da *Sprint 1* – Pedido:

1. Cadastrar novo pedido;
2. Listar pedidos;
3. Busca de pedido por id;
4. Alteração do *status* do pedido;
5. Alteração do *status* de pagamento do pedido;
6. Busca de pedido por status;
7. Implementação da atualização de estoque quando pedido criado.

Atividades da *Sprint 1* – Pedido de Produção:

1. Cadastrar novo pedido de produção;
2. Listar pedidos de produção;
3. Busca de pedido de produção por id;
4. Implementação da atualização de estoque quando pedido de produção criado.

Para todas as entidades envolvidas nas implementações descritas acima, foram feitas todas as validações de campos no *backend*, assim como o tratamento de exceções e relacionamento entre tabelas.

### 3.5.2 *Sprint 2*

Com todas os principais CRUDs (*features* de *create*, *update* e *delete*) criados, nesta *Sprint* foram desenvolvidas algumas funcionalidades não-CRUD do sistema, conforme os requisitos já definidos.

Atividades da *Sprint 2*:

1. Envio de e-mail para cliente e usuários do sistema quando pedido é criado;
2. Envio de e-mail para cliente e usuários do sistema quando o status do pedido é alterado;
3. Envio de e-mail alerta para usuários quando o estoque de determinada cerveja está baixo (porcentagem de estoque baixo definido pelo usuário);
4. Permitir ao usuário o *download* de relatórios periódicos de fluxo de estoque;
5. Permitir ao usuário o *download* de relatórios periódicos de pedidos.

### 3.5.3 *Sprint 3*

Nesta terceira fase do projeto, foram finalizadas as funcionalidades do *backend* e iniciado a criação das interfaces com o usuário (telas).

Atividades da *Sprint 3 – Backend*:

1. Cadastro de usuários no sistema;
2. Autenticação de usuário no sistema;
3. Alteração de senha de usuário;
4. Alteração de informações do usuário;

Atividades da *Sprint 3 – Frontend*:

1. Criação da interface para cadastro de cerveja;
2. Criação da interface para listagem de cervejas;
3. Criação da interface para edição de cervejas;
4. Criação da interface para exclusão de cerveja;
5. Criação de interface modal para cadastro de tipo de cerveja;
6. Criação de interface modal para edição de tipos de cerveja;
7. Criação de interface modal para edição de tipos de cerveja;
8. Criação de interface modal para exclusão de tipos de cerveja.

### 3.5.4 Sprint 4

Com o *backend* finalizado e as primeiras interfaces de usuário criadas, na *Sprint 4* do projeto foi dada continuidade no desenvolvimento do *frontend*, conforme descrito abaixo:

Atividades da *Sprint 4*:

1. Criação da interface para cadastro de cliente;
2. Criação da interface para listagem de clientes;
3. Criação da interface para edição de cliente;
4. Criação da interface para exclusão de cliente;
5. Criação da interface modal para cadastro de cidade e estado;
6. Criação da interface modal para edição de cidade e estado;
7. Criação da interface modal para edição de cidade e estado;
8. Criação da interface modal para exclusão de cidade e estado;
9. Criação da interface para criação de pedidos de produção;
10. Criação da interface para listagem de pedidos de produção;
11. Criação da interface para criação de pedidos;
12. Criação da interface para listagem de pedidos;
13. Criação das funcionalidades para alteração de status dos pedidos;
14. Criação das funcionalidades para alteração de status de pagamento dos pedidos;

### 3.5.5 Sprint 5

Na última etapa do projeto, as interfaces do usuário pendentes foram criadas, finalizando o sistema.

O fluxo completo das funcionalidades pode ser visualizado nos diagramas de sequência, disponíveis no Apêndice 5.

Nesta fase, a API do *backend* foi devidamente documentada, conforme Apêndice 7, e foi inserido uma massa de dados bem maior para que os testes fossem finalizados.

Atividades da *Sprint 5*:

1. Criação da página inicial, exibindo seis quadros com estatísticas gerais de vendas, estoque, etc e dois gráficos com dados periódicos que podem ser alterados pelo usuário;
2. Criação da interface para criação e download de relatórios periódicos de estoque;
3. Criação da interface para criação e download de relatórios periódicos de pedidos;
4. Criação da interface de *login* do sistema;
5. Criação da interface de alteração de senha do sistema;
6. Criação da interface para criação e alteração de usuário no sistema.

## 4 APRESENTAÇÃO DOS RESULTADOS

Neste capítulo é apresentada a arquitetura do sistema, bem como suas telas e funcionalidades.

O sistema é executado em uma interface *web*, para que os administradores do negócio possam ter o controle e gerenciamento do estoque, assim como das vendas, pedidos e clientes.

A seguir, é apresentada a arquitetura do sistema e em sequência, é possível visualizar detalhadamente todas as funcionalidades do sistema.

### 4.1 ARQUITETURA DO SISTEMA

O sistema foi desenvolvido com o *backend* e *frontend* desacoplados, isso é, em projetos diferentes. A comunicação entre eles, se dá pelo modelo REST, utilizando os verbos HTTP para consultas, persistência e deleção de dados.

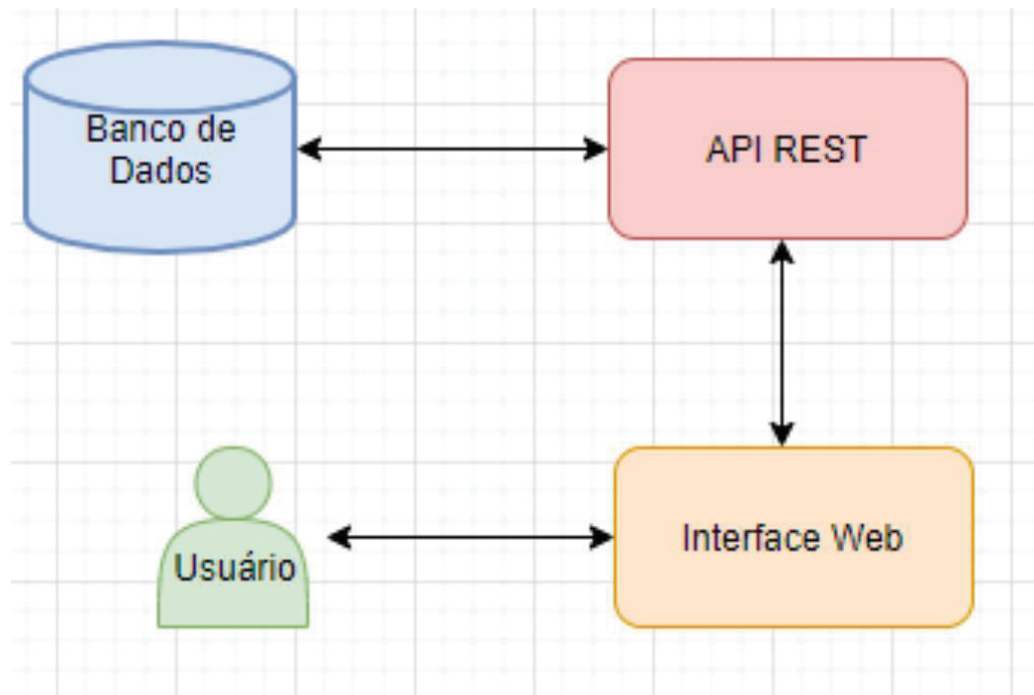
O *backend* foi implementado na arquitetura REST, onde foi desenvolvido uma API contendo todos os requisitos e funcionalidades para a operação do sistema, conforme documentado no APÊNDICE 6.

A FIGURA 4, representa o desenho da arquitetura do sistema como um todo, que consiste em três camadas:

- Interface Web;
- REST API;
- Banco de dados.

Nessa arquitetura, a comunicação entre a camada de interface com o usuário (web) e a camada onde de fato são aplicadas e desenvolvidas as regras de negócio (REST API) são feitas através de chamadas HTTP.

FIGURA 4 - Arquitetura do Sistema



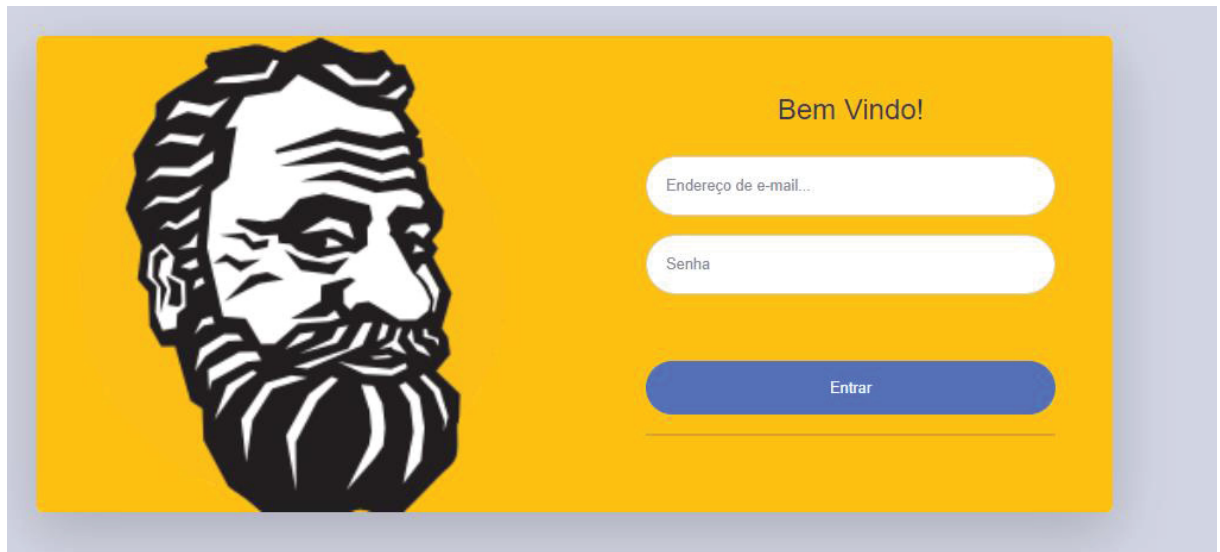
FONTE: O autor (2022).

## 4.2 SOFTWARE

O sistema apresenta inicialmente a interface de *login* e autenticação no sistema. Para que um usuário possa acessar o sistema, o mesmo deve fornecer e-mail e senha, previamente cadastrados no sistema. (FIGURA 5).

Por se tratar de um sistema de controle e administração com poucos usuários, o cadastro de novos usuários somente pode ser feito por um usuário já logado.

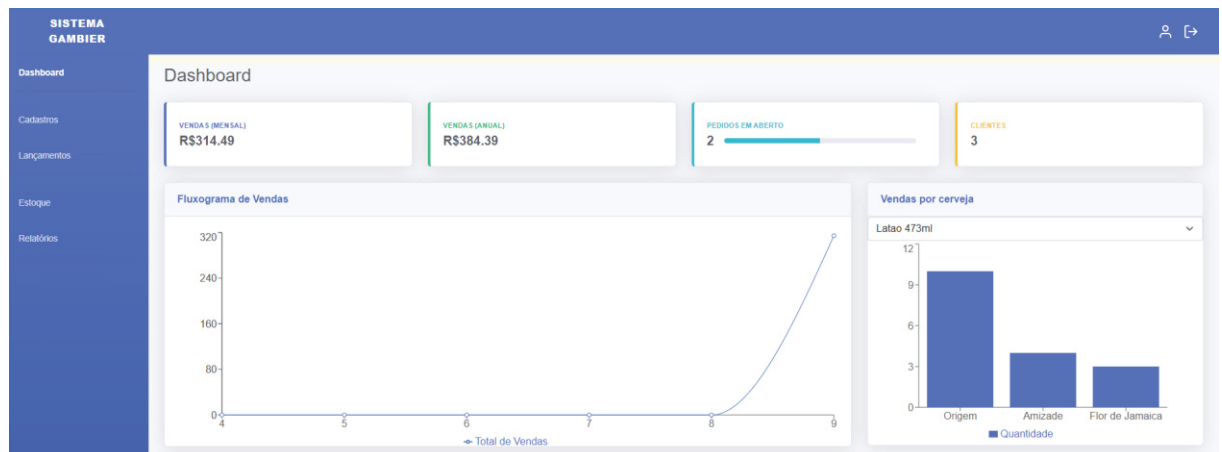
FIGURA 5 - Tela de Autenticação no Sistema



FONTE: O autor (2022).

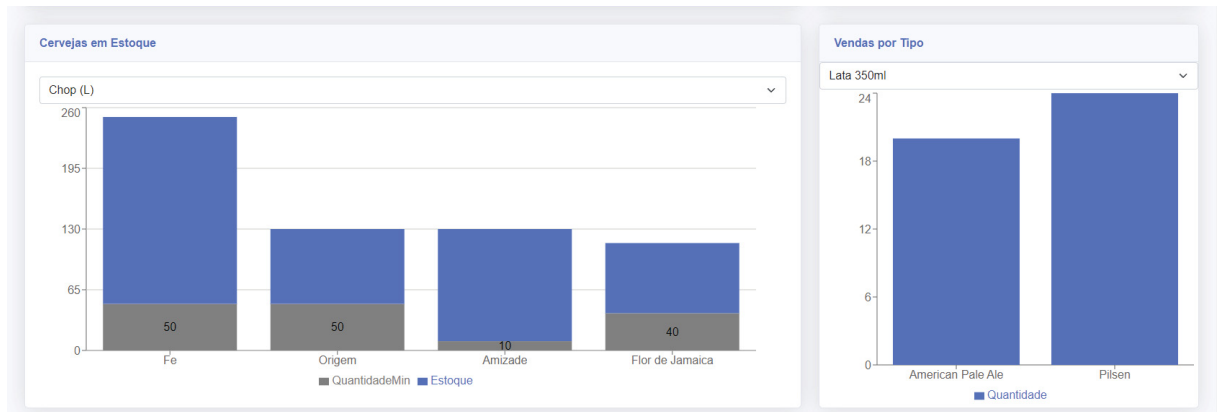
Ao executar o login, o sistema exibe a tela inicial contendo um menu lateral, com as principais ações que o usuário pode executar e também uma série de *dashboards* e resumos importantes para os administradores.

FIGURA 6 - Tela inicial do Sistema – Parte Superior



FONTE: O autor (2022).

FIGURA 7 - Tela Inicial - Parte Inferior



FONTE: O autor (2022).

No menu lateral, presente em todas as telas do sistema, o usuário tem as seguintes opções:

- Cadastro;
- Lançamentos;
- Estoque;
- Relatórios.

Na opção cadastro, há três sub opções, representando as entidades de domínio do sistema – cerveja, tipos de cerveja e clientes.











Nas figuras a seguir, estão representadas todas as telas dessas três opções, contendo as telas principais, onde são listados todos os dados existentes e também o modal que é utilizado para cadastro e edição dos dados.

Em todas as telas de listagem, há um campo de busca, onde conforme o usuário digita, a lista atualiza automaticamente conforme o que está sendo digitado.

FIGURA 8 - Tela principal de Cerveja

Cervejas

[Nova Cerveja](#)

SKU	Nome	Tipo	Embalagem	Preço	Ações
APAO0473	Origem	American Pale Ale	Latão (473ml)	R\$16.99	 
APAO0350	Origem	American Pale Ale	Lata (350ml)	R\$8.99	 
EPAF1000	Fe	New England Pale Ale	Chopp (1L)	R\$12.79	 
PILA0000	Amizade	Pilsen	Long Neck	R\$5.99	 
PILA0350	Amizade	Pilsen	Lata (350ml)	R\$3.99	 











1 2

FONTE: O autor (2022).

FIGURA 9 - Modal de cadastro/edição de Cerveja

Cervejas

[Nova Cerveja](#)

SKU	Nome	Preço	Ações
APAO0473	Origem	R\$16.99	 
APAO0350	Origem	R\$8.99	 
EPAF1000	Fe	R\$12.79	 
PILA0000	Amizade	R\$5.99	 
PILA0350	Amizade	R\$3.99	 

1 2

**Nova Cerveja**

SKU\*

Nome\*  Tipo\*

Embalagem\*  Preço

% Alcool\*  Quantidade Mínima\*

Descrição







[Cancelar](#) [Salvar](#)

FONTE: O autor (2022).

FIGURA 10 - Tela principal de Tipos de Cerveja

Tipos de Cerveja

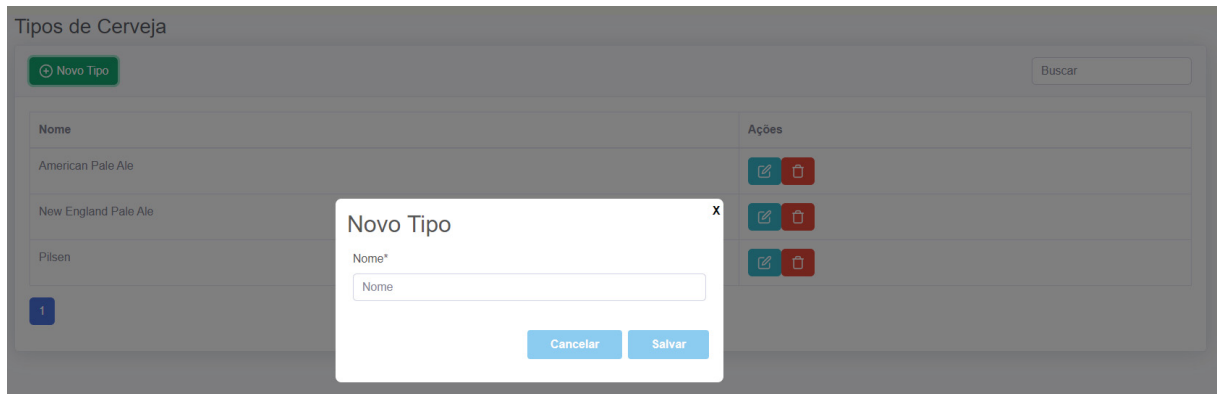
[Novo Tipo](#)

Nome	Ações
American Pale Ale	 
New England Pale Ale	 
Pilsen	 

1

FONTE: O autor (2022).

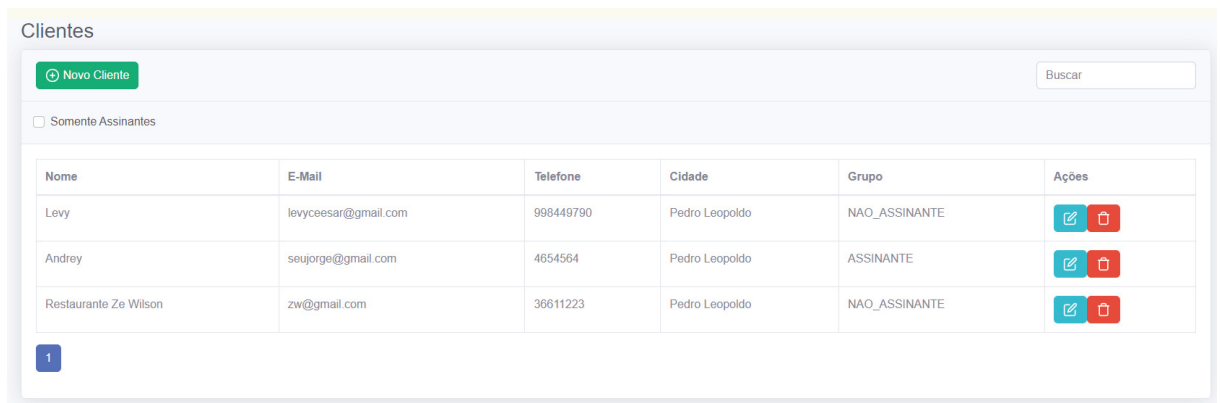
FIGURA 11 - Modal de cadastro/edição de Tipos de Cerveja



FONTE: O autor (2022).

Na tela de listagem de clientes, há um *checkbox* em que o usuário da aplicação pode filtrar somente os assinantes.

FIGURA 12 - Tela Principal de Clientes



FONTE: O autor (2022).

FIGURA 13 - Modal de cadastro/edição de Clientes

The image shows a web application interface for managing clients. On the left, a sidebar titled 'Clientes' contains a green '+ Novo Cliente' button and a checkbox for 'Somente Assinantes'. Below this is a list of clients with names: Levy, Andrey, and Restaurante Ze Wilson. A blue button with the number '1' is at the bottom of the sidebar. The main area displays a 'Novo Cliente' modal form with the following fields:

- Nome\* (text input)
- E-Mail\* (text input, value: gambier@gambier.com.br)
- Data de Nascimento (text input, value: Ex: 9.90)
- Tipo Pessoa\* (dropdown menu, value: Pessoa Física)
- CPF/CNPJ\* (text input)
- Grupo\* (dropdown menu, value: Assinante)
- Endereço section:
  - Rua (text input)
  - Número (text input)
  - Complemento (text input)
  - Bairro (text input)
  - Cidade\* (dropdown menu)
  - Estado\* (dropdown menu)
  - CEP (text input)

At the bottom right of the modal are two blue buttons: 'Cancelar' and 'Salvar'.

FONTE: O autor (2022).

A segunda opção do menu lateral, expande um sub menu constando as opções de pedido e produção. Semelhante as telas anteriores, cada opção exibe sua tela principal e sua listagem.

Na tela de pedidos, há um filtro em que o usuário pode alternar a exibição da lista preenchendo um ou mais campos. Dessa forma, os usuários da aplicação têm a opção de filtrar por um intervalo de datas (“data início” e “data fim”), nome do cliente, status do pedido, status do pagamento e pelo usuário que cadastrou o pedido.











FIGURA 14 - Tela principal de Pedidos

Pedidos

[Novo Pedido](#)

Data Início\*  Data Fim\*  Cliente

Status  Status de Pagamento  Usuário

Data	Cliente	Status	Status Pagamento	Entrega	Ações
2022-11-01	Levy	ENTREGUE	PAGO	2022-10-14	 
2022-09-30	Levy	ENTREGUE	A RECEBER	2022-10-02	 
2022-09-01	Levy	ENTREGUE	PAGO	2022-10-01	 
2022-10-20	Levy	CRIADO	A RECEBER		 
2022-10-01	Levy	CONFIRMADO	A RECEBER		 

FONTE: O autor (2022).

Conforme os fluxos já apresentados em outras telas, ao selecionar “Novo Pedido”, o sistema abre um modal para o lançamento de um novo pedido.

FIGURA 15 - Modal de lançamento de Pedidos


Pedidos

[Novo Pedido](#)

Data Início\*  Status

Cliente  Telefone  E-mail

Adicionar Itens











Cerveja  Quantidade  

Resumo do Pedido

Nome	Embalagem	Preço Unitário	Quantidade	Total
Total R\$0.00				

[Cancelar](#) [Salvar](#)

Cliente  Usuário

Data	Cliente	Entrega	Ações
2022-11-01	Levy	2022-10-16	 
2022-09-30	Levy	2022-10-02	 
2022-09-01	Levy	2022-10-01	 
2022-10-20	Levy		 
2022-10-01	Levy		 

FONTE: O autor (2022).

Após um pedido, se o estoque de uma cerveja for menor que a quantidade mínima definida, um e-mail de alerta é enviado aos usuários.

FIGURA 16 - E-mail de alerta de estoque baixo

Sistema Gambier - Alerta de Estoque Baixo!!! ☰ Caixa de entrada x

SistemaGambier por sendgrid.net  
para mim, henriquemail ▼

## Hora de produzir!!!

A(s) cerveja(s): Origem\_LATAO\_473, estão com estoque abaixo do mínimo!

Reponha o estoque e não perca nenhuma venda!



FONTE: O autor (2022).

Diferente das demais, na listagem de pedidos, há um ícone de detalhes, onde o usuário pode visualizar os detalhes do pedido e também alterar o status e status de pagamento do mesmo.

Os botões de alteração de status e alteração de status de pagamento, são exibidos conforme o status atual do pedido.

FIGURA 17 - Modal de detalhes do Pedido / Alteração de Status

The modal displays the following information:

- Cliente:** Levy
- Telefone:** 998449790
- E-mail:** levyceesar@gmail.com
- Status:** CONFIRMADO
- Status de Pagamento:** A\_RECEBER
- Data:** 2022-10-01
- Total R\$0.00**
- Items:**

Cerveja	Embalagem	Preço Unitário	Quantidade	Total
Origem	LATAO_473	R\$16.99	10	R\$169.9
Origem	LATA_350	R\$8.99	12	R\$107.88

Buttons at the bottom: Fechar, Cancelar Pedido, Entregar Pedido, Pagamento Recebido.

FONTE: O autor (2022).

FIGURA 18 - Modal de detalhes do Pedido / Alteração de Status (2)

The modal displays the following information:

- Cliente:** Levy
- Telefone:** 998449790
- E-mail:** levyceesar@gmail.com
- Status:** ENTREGUE
- Status de Pagamento:** PAGO
- Data:** 2022-11-01
- Data de Entrega:** 2022-10-25
- Total R\$0.00**
- Items:**

Cerveja	Embalagem	Preço Unitário	Quantidade	Total
Origem	LATA_350	R\$6.99	10	R\$69.9

Button at the bottom: Fechar.

FONTE: O autor (2022).

Quando o status de um pedido é alterado para “Confirmado”, o cliente recebe um e-mail informativo de confirmação de pedido.

FIGURA 19 - E-mail de confirmação de pedido

Sistema Gambier - O seu pedido foi confirmado!  Caixa de entrada x



SistemaGambier por\_sendgrid.net  
para mim ▾



# GAMBIER

## Pedido confirmado!

Levy, seu pedido foi confirmado e já está sendo separado.

Quantidade	Cerveja	Tipo	Embalagem	Valor
2	Origem	American Pale Ale	LATAO_473	R\$ 33,98
<b>Total</b>				<b>R\$ 33,98</b>

### Status do Pedido

CONFIRMADO

### Status de Pagamento

A\_RECEBER

FONTE: O autor (2022).

Na tela principal de produção, assim como em “Pedido” é possível que o usuário filtre os dados por um intervalo de datas (“data início” e “data fim”) e também pelo usuário que fez o lançamento.

Como, por regra de negócio, não é possível editar uma produção, na listagem dessa tela, há apenas os ícones de detalhes e exclusão.









FIGURA 20 - Tela principal de Produção

Produção

**Novo Lançamento**

Data Início\*  Data Fim\*  Usuário

**Filtrar**

Quantidade Total	Data de Lançamento	Custo Total	Responsável	Ações
10	2021-10-01	100	Levy	 
100	2022-04-25	400	Levy	 
200	2021-07-01	400	Henrique	 
100	2022-04-01	350	Henrique	 

FONTE: O autor (2022).


Ao selecionar “Novo Lançamento”, o sistema abre um modal para o lançamento de uma nova produção.

FIGURA 21 - Modal de lançamento de Produção


Produção

**Novo Lançamento**

Data Início  Usuário

Quantidade Total  Data  Custo Total  

**Adicionar Itens**

Cerveja  Quantidade  

**Resumo**

Cerveja	Quantidade
Cerveja	

**Cancelar** **Salvar**

FONTE: O autor (2022).

Quando uma produção é lançada, o usuário da aplicação recebe um e-mail informativo sobre o lançamento efetuado.

FIGURA 22 - E-mail informativo de lançamento de Produção.

Sistema Gambier - Produção Cadastrada! Caixa de entrada x

 **SistemaGambier** por [sendgrid.net](mailto:sendgrid.net)  
para mim, henriquemail ▾



# GAMBIER

## Produção cadastrada!

Levy botou a mão na massa e cadastrou uma nova produção.

Cerveja	Tipo	Embalagem	Quantidade
Origem	American Pale Ale	LATAO_473	\$100

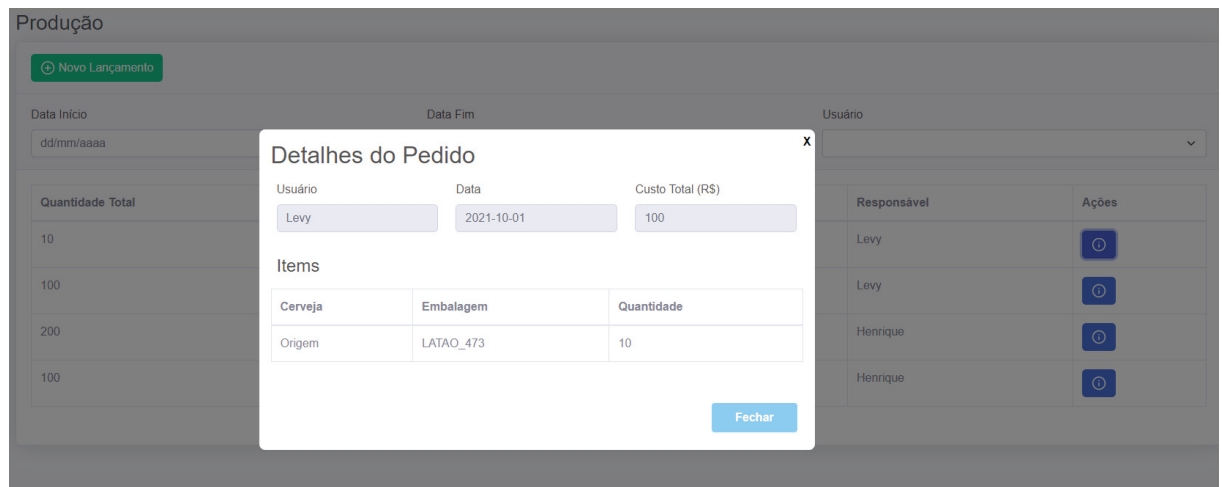
**Confira abaixo o estoque atualizado:**

Cerveja	Tipo	Embalagem	Quantidade em Estoque
Origem	American Pale Ale	LATAO_473	110

FONTE: O autor (2022).

Ao clicar no ícone de detalhes, o sistema exibe um modal com os detalhes da produção.

FIGURA 23 - Modal de detalhes da Produção



FONTE: O autor (2022).

A terceira opção do menu, “Estoque”, exibe a listagem de cervejas em estoque, tal como seus detalhes.

Essa tela serve apenas como informação, com o usuário podendo filtrar as cervejas em estoque conforme digita no campo “buscar”. Quando o filtro está ativo, ou seja, há algum texto no campo, a quantidade total do estoque atualiza automaticamente conforme o filtro.

FIGURA 24 - Tela de Cervejas em estoque

Cerveja	Tipo	Embalagem	Quantidade Min	Quantidade em Estoque
Origem	American Pale Ale	Lafão (473ml)	30	10
Origem	American Pale Ale	Lata (350ml)	10	100
Fe	New England Pale Ale	Chopp (1L)	5	200
Amizade	Pilsen	Long Neck	5	100
Amizade	Pilsen	Lata (350ml)	5	100
			TOTAL	510

FONTE: O autor (2022).

FIGURA 25 - Tela de Cervejas em estoque com filtro ativo

Estoque

On

Cerveja	Tipo	Embalagem	Quantidade Min	Quantidade em Estoque
Origem	American Pale Ale	Latão (473ml)	30	10
Origem	American Pale Ale	Lata (350ml)	10	100
			TOTAL	110

FONTE: O autor (2022).

O administrador do negócio, também pode selecionar no menu lateral a opção “Relatórios” e um sub menu será aberto, onde é possível definir entre emitir um relatório de vendas e de vendas diárias.

No relatório de vendas diárias, o usuário seleciona um intervalo de datas e opcionalmente pode selecionar um cliente específico.

FIGURA 26 - Tela de emissão de relatório de vendas diárias

Relatório de Vendas Diárias

Data Início\*  Data Fim\*  Cliente

FONTE: O autor (2022).

O relatório de vendas diárias consiste em um resumo das vendas (somente pedidos confirmados ou entregues e pagos).

FIGURA 27 - Relatório de vendas diárias



Data	Quantidade Vendas	Total Faturado
01/09/2022 00:00	1	R\$89,90
01/11/2022 00:00	1	R\$89,90

FONTE: O autor (2022).

Os relatórios de vendas, são relatórios globais em que o usuário pode emitir todo o histórico de vendas de uma determinada cerveja ou cliente, selecionando a opção que preferir.

FIGURA 28 - Tela de emissão de relatório de vendas

Relatórios de Vendas

Por cerveja

Por cliente

[Gerar Relatório](#)

FONTE: O autor (2022).

FIGURA 29 - Relatório de vendas por cerveja

Relatório de Vendas				
Cerveja	Embalagem	Tipo	Quantidade Vendida	Faturamento
Origem	LATAO_473	American Pale Ale	40.00	R\$379,60
Origem	LATA_350	American Pale Ale	12.00	R\$107,88

FONTE: O autor (2022).

FIGURA 30 - Relatório de vendas por cliente



The image shows a screenshot of a web application interface. At the top, there is an orange header bar with the text "Relatório de Vendas" in white. Below the header, there is a table with four columns. The columns are labeled "Cliente", "Tipo", "Grupo", and "N° Pedidos". The first row of data shows "Levy" under "Cliente", "FISICA" under "Tipo", "NAO\_ASSINANTE" under "Grupo", and "4" under "N° Pedidos".

Cliente	Tipo	Grupo	N° Pedidos
Levy	FISICA	NAO_ASSINANTE	4

FONTE: O autor (2022).

No menu superior o sistema apresenta um ícone de usuário, que ao ser selecionado redireciona para uma página mostrando as informações do usuário logado, tais quais a lista de usuários cadastrados no sistema e também as ações que podem ser realizadas.

FIGURA 31 – Tela principal de usuário

Nome: Levy  
E-mail: levyceesar@gmail.com

[✎ Editar](#) [✎ Alterar Senha](#)

### Usuários

Nome	Email
Levy	levyceesar@gmail.com
Henrique	henriquemail@gambier.com

[+ Novo Usuário](#)

FONTE: O autor (2022).

Nesta tela, o usuário tem as seguintes ações disponíveis:

- Adicionar novo usuário;
- Editar seus dados;
- Alterar senha;

FIGURA 32 - Modal de cadastro de novo usuário

Nome: Levy  
E-mail: levyceesar@gmail.com

[✎ Editar](#) [✎ Alterar Senha](#)

### Usuários

Nome	Email
Levy	levyceesar@gmail.com
Henrique	henriquemail@gambier.com

[+ Novo Usuário](#)

#### Novo Usuário

Nome\*

E-Mail\*

Senha

[Cancelar](#) [Salvar](#)

FONTE: O autor (2022).

FIGURA 33 - Modal de edição do usuário

The image shows a user management interface. At the top, there is a user profile for 'Levy' with email 'levyceesar@gmail.com'. Below this are two buttons: 'Editar' and 'Alterar Senha'. A table titled 'Usuários' lists the user 'Levy' with the name 'Henrique'. A green button 'Novo Usuário' is at the bottom left. A modal window titled 'Editar Usuário' is open, containing two input fields: 'Nome\*' with the value 'Levy' and 'E-Mail\*' with the value 'levyceesar@gmail.com'. At the bottom of the modal are 'Cancelar' and 'Salvar' buttons.

FONTE: O autor (2022).

FIGURA 34 - Modal de alteração de senha

The image shows the same user management interface as Figure 33. The modal window is now titled 'Alteração de Senha'. It contains two input fields: 'Senha Atual' and 'Nova Senha'. At the bottom of the modal are 'Cancelar' and 'Alterar' buttons.

FONTE: O autor (2022).

## 5. CONSIDERAÇÕES FINAIS

O desenvolvimento do sistema para controle de estoque da microcervejaria Gambier cumpriu com o objetivo inicial proposto.

Foram utilizadas diversas ferramentas e metodologias vistas durante o curso de especialização em Engenharia de Software, destacando a linguagem de programação Java, que durante o curso foi abordada em todos os aspectos de desenvolvimento (web, mobile e desktop), a arquitetura de software com o padrão UML e as metodologias ágeis através do Scrum, estas duas últimas, de suma importância para o planejamento e execução do sistema como um todo.

### 5.1 RECOMENDAÇÕES PARA TRABALHOS FUTUROS

A arquitetura do projeto, incluindo as classes, entidades, relacionamentos e banco de dados foram pensados para que no futuro, possa ser adicionado ao sistema uma plataforma de e-commerce sem que haja necessidade de alterações bruscas na estrutura do mesmo e sem risco de prejudicar os dados já em uso.

Além disso, para trabalhos futuros há também a possibilidade de inclusão de gráficos e estatísticas mais detalhadas, assim como tornar o sistema um ERP, adicionando controle do fluxo financeiro da empresa.

## REFERÊNCIAS

- COIMBRA, C C; MARTINS, L F; PANDOLFI, M A C. **Análise dos Indicadores do Mercado Cervejeiro Brasileiro**. In: IV Simpósio de Tecnologia Fatec Taquaritinga, 2018, Taquaritinga. Fatec, 2018.
- LACERDA, V S S; PEREIRA, D H; OLIVEIRA, P M; SOUZA, R S; SILVA, F M. **Os desafios para gestão de micro e pequenas empresas: um estudo de caso**. In: XIII Simpósio de Excelência em Gestão e Tecnologia, Rezende, 2016, Rezende. AEDB, 2016.
- MARCUSSO, E F. **As microcervejarias no Brasil atual: Sustentabilidade e Territorialidade**. Trabalho de Pós-Graduação (Sustentabilidade na Gestão Ambiental) – Universidade Federal de São Carlos, Sorocaba, 2018.
- SEBRAE, Estudo do SEBRAE reúne oportunidades para microcervejarias: Microcervejarias, 2017. Disponível em <<https://www.sebrae.com.br/sites/PortalSebrae/ufs/sc/artigos/estudo-do-sebrae-reune-oportunidades-para-microcervejarias>> Acesso em: 12 out. 2021
- NOVAES, A. G. **Logística e gerenciamento de cadeia de distribuição: estratégia, operação e avaliação**. 1ª edição. Rio de Janeiro: Campus, 2001
- SILVA, G. F. L. **Proposta para gerenciamento de estoque de uma micro cervejaria**. Departamento de Engenharia de Produção – Graduação em Engenharia de Produção da Universidade Estadual de Maringá. Maringá, 2017.
- DANDARO, F ; MARTELLI, L. L.. **Planejamento e Controle de estoque nas organizações**. Revista Gestão Industrial, v.11, n 02: p170-185, 2015.
- BORGES C. E.; BORGES C. T.; CAMPOS S. M.. **Implantação de um sistema para o controle de estoques em uma gráfica/editora de uma universidade**. Revista Eletrônica Produção & Engenharia, v. 3, n. 1, p. 236-247, Jul./Dez. 2010.
- BALLOU, R. H. **Gerenciamento da Cadeia de Suprimentos/Logística Empresarial**. 5ª edição. Porto Alegre: Bookman, 2001.
- JÓ, M. Y. **Estudo comparativo de modelo de gestão de estoque em empresas de bebidas**. Especialização. DECONT – Pós-Graduação em Contabilidade da Universidade Federal do Paraná. Curitiba, 2012.

BOOCH, Grady; JACOBSON, Ivar; RUMBAUGH James. **Uml - Guia do Usuário**. Rio de Janeiro: Elsevier, 2005.

AGILE MANIFESTO, Manifesto for Agile Software Development, 2001. Disponível em < <http://agilemanifesto.org/iso/ptbr/manifesto.html>> Acesso em: 25 jan. 2022.

MACHADO e MEDINA, SCRUM – Método ÁgilÇ uma mudança cultural na Gestão de Projetos de Desenvolvimento de Software <[http://uniesp.edu.br/sites/\\_biblioteca/revistas/20170531154126.pdf](http://uniesp.edu.br/sites/_biblioteca/revistas/20170531154126.pdf)> Acesso em: 25 jan. 2022

DEITEL, H; DEITEL, P. **Java: Como Programar** 10. Ed. São Paulo: Pearson, 2016

SPRING Framework Overview. Disponível em: <https://docs.spring.io/spring-framework/docs/current/reference/html/overview.html> Acesso em: 29 abril. 2022

SPRING. Disponível em: <<https://spring.io/web-applications>> Acesso em: 29 out. 2021

FLANAGAN, D. **JavaScript: O guia definitivo** 6. Ed. São Paulo: Bookman, 2011

REACT. Disponível em: <<https://pt-br.reactjs.org>> Acesso em: 30 out. 2021

REACT, What is React. Disponível em: <<https://pt-br.reactjs.org/tutorial/tutorial.html#what-is-react>> Acesso em: 30 out. 2021

MILANI, A. **MySQL: Guia do Programador**. São Paulo: Novatec, 2006

TILKOV, S. Uma rápida introdução ao REST. 29 out. 2008 seção Artigos. Disponível em: <<https://www.infoq.com/br/articles/rest-introduction/>> Acesso em: 29 out. 2021

MOZILLA, Códigos de status de respostas HTTP. 7 out. 2022 seção HTTP. Disponível em: <<https://developer.mozilla.org/pt-BR/docs/Web/HTTP/Status/>> Acesso em: 29 out. 2022

PIRES, A F. Consumindo API REST com Retrofit + Kotlin no Android. 16 dez. 2018 seção Artigos. Disponível em: < <https://medium.com/@alifyzpires/consumindo-api-rest-com-retrofit-kotlin-no-android-abba52820c/>> Acesso em: 30 out. 2021

FOWLER, M. Richardson Maturity Model; 18 mar. 2010. Disponível em: <https://martinfowler.com/articles/richardsonMaturityModel.html> Acesso em: 01 mai. 2022

VIEIRA, Denisson. Scrum: A Metodologia Ágil Explicada de Forma Definitiva. 26 jun. 2014. Disponível em: Acesso em: 17 out. 2021.


SCHWABER K., Agile Project Management With Scrum, Microsoft Press, 2004


SABBAGH, Rafael. Scrum: Gestão ágil para projetos de sucesso. São Paulo: Casa do Código, 2013.

## APÊNDICE 1 – HISTÓRIAS DE USUÁRIO

No início do projeto, para servir como base e direcionamento na implementação do sistema, foram desenvolvidas as histórias de usuário, essas que auxiliam no levantamento de requisitos do sistema.

FIGURA 35 - História de Usuário “Manutenção de Cerveja”

 **US - Manutenção de Cerveja**  
na lista [Histórias de Usuário](#)

 **Descrição** Editar

**Como:** um administrador do sistema  
**Quero:** adicionar, alterar, consultar e excluir cervejas  
**Para:** que as cervejas e suas especificações estejam sempre atualizadas.

**Cenário 1:** Adicionar nova cerveja  
**Dado que** desejo adicionar uma nova cerveja  
**E** o sistema abre o modal de cadastro após a opção ser selecionada  
**Quando** envio os dados referentes a cerveja  
**Então** o sistema grava a nova cerveja  
**E** apresenta a mensagem de "Cerveja salva com sucesso"

**Cenário 2:** Alterar uma cerveja existente  
**Dado que** desejo alterar uma cerveja existente  
**E** o sistema abre o modal com os dados da cerveja escolhida após a opção ser selecionada para que possa ser modificada  
**Quando** envio os dados com as opções modificadas  
**Então** o sistema altera os dados da cerveja escolhida  
**E** apresenta a mensagem de "Cerveja editada com sucesso"

**Cenário 3:** Consultar cervejas  
**Dado que** desejo consultar as cervejas cadastradas  
**Quando** clico na opção do menu "Cervejas"  
**Então** o sistema abre a lista com todas as cervejas que estão cadastradas  
**E** posso visualizar a lista de todas as cervejas

**Cenário 4:** Excluir uma cerveja  
**Dado que** excluir uma cerveja  
**E** seleciono o botão de excluir de uma cerveja específica na lista  
**Quando** o sistema abre um pop-up com a mensagem de confirmação  
**Então** clico na opção "sim"  
**E** apresenta a mensagem de "Cerveja excluída com sucesso"

FIGURA 36 - História de Usuário "Manutenção de Tipos de Cerveja"



## US - Manutenção de Tipos de Cerveja

na lista [Histórias de Usuário](#)



Descrição
Editar

**Como:** um administrador do sistema

**Quero:** adicionar, alterar, consultar e excluir tipos de cerveja

**Para:** que os tipos de cerveja estejam sempre atualizados.

**Cenário 1:** Adicionar novo tipos de cerveja

**Dado que** desejo adicionar um novo tipo de cerveja

**E** o sistema abre o modal de cadastro após a opção ser selecionada

**Quando** envio os dados referentes ao tipo de cerveja

**Então** o sistema grava o novo tipo de cerveja

**E** apresenta a mensagem de "Tipo de Cerveja salvo com sucesso"

**Cenário 2:** Alterar um tipo de cerveja existente

**Dado que** desejo alterar um tipo de cerveja existente

**E** o sistema abre o modal com os dados do tipo de cerveja escolhido após a opção ser selecionada para que possa ser modificado

**Quando** envio os dados com as opções modificadas

**Então** o sistema altera os dados do tipo de cerveja escolhido

**E** apresenta a mensagem de "Tipo de Cerveja editado com sucesso"

**Cenário 3:** Consultar tipos de cerveja

**Dado que** desejo consultar os tipos de cerveja cadastrados

**Quando** clico na opção do menu "Tipos de Cerveja"

**Então** o sistema abre a lista com todos os tipos de cerveja que estão cadastrados

**E** posso visualizar a lista de todos os tipos de cerveja

**Cenário 4:** Excluir um tipo de cerveja

**Dado que** excluir um tipo de cerveja

**E** seleciono o botão de excluir de um tipo de cerveja específico na lista

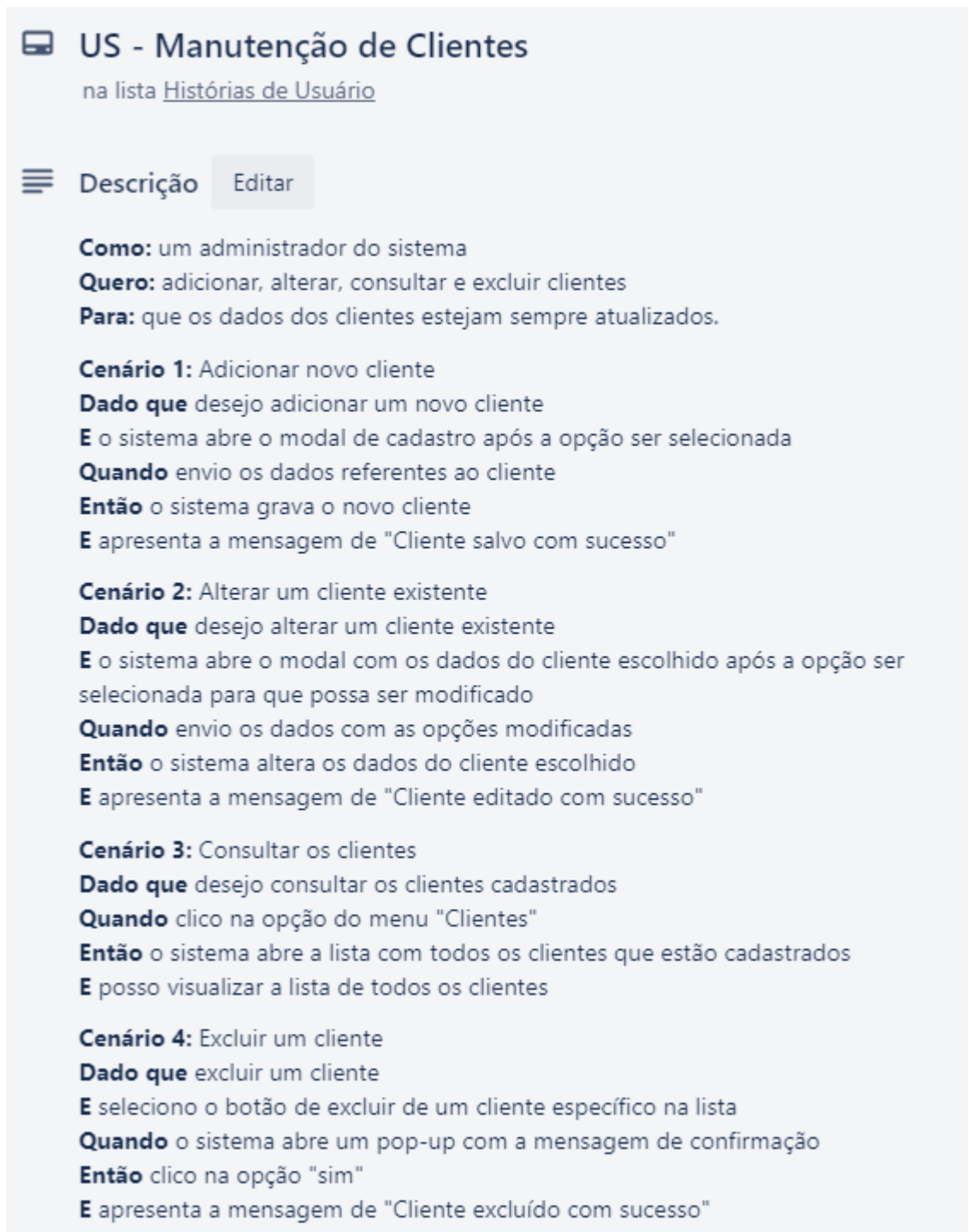
**Quando** o sistema abre um pop-up com a mensagem de confirmação

**Então** clico na opção "sim"

**E** apresenta a mensagem de "Tipo de Cerveja excluído com sucesso"

FONTE: O autor (2022).

FIGURA 37 - História de Usuário "Manutenção de Clientes"



**US - Manutenção de Clientes**  
na lista [Histórias de Usuário](#)

Descrição Edit

**Como:** um administrador do sistema  
**Quero:** adicionar, alterar, consultar e excluir clientes  
**Para:** que os dados dos clientes estejam sempre atualizados.

**Cenário 1:** Adicionar novo cliente  
**Dado que** desejo adicionar um novo cliente  
**E** o sistema abre o modal de cadastro após a opção ser selecionada  
**Quando** envio os dados referentes ao cliente  
**Então** o sistema grava o novo cliente  
**E** apresenta a mensagem de "Cliente salvo com sucesso"

**Cenário 2:** Alterar um cliente existente  
**Dado que** desejo alterar um cliente existente  
**E** o sistema abre o modal com os dados do cliente escolhido após a opção ser selecionada para que possa ser modificado  
**Quando** envio os dados com as opções modificadas  
**Então** o sistema altera os dados do cliente escolhido  
**E** apresenta a mensagem de "Cliente editado com sucesso"

**Cenário 3:** Consultar os clientes  
**Dado que** desejo consultar os clientes cadastrados  
**Quando** clico na opção do menu "Clientes"  
**Então** o sistema abre a lista com todos os clientes que estão cadastrados  
**E** posso visualizar a lista de todos os clientes

**Cenário 4:** Excluir um cliente  
**Dado que** excluir um cliente  
**E** seleciono o botão de excluir de um cliente específico na lista  
**Quando** o sistema abre um pop-up com a mensagem de confirmação  
**Então** clico na opção "sim"  
**E** apresenta a mensagem de "Cliente excluído com sucesso"

FONTE: O autor (2022).

FIGURA 38 - História de Usuário "Manutenção de Pedido"

## US - Manutenção de Pedido

na lista [Histórias de Usuário](#)

Descrição Editar

**Como:** um administrador do sistema  
**Quero:** adicionar, alterar consultar e alterar status dos pedidos  
**Para:** que eu possa cadastrar e administrar as vendas

**Cenário 1:** Adicionar novo pedido  
**Dado que** desejo adicionar um novo pedido  
**E** o sistema abre o modal de cadastro após a opção ser selecionada  
**Quando** envio os dados referentes ao pedido  
**Então** o sistema grava o pedido  
**E** apresenta a mensagem de "Pedido salvo com sucesso"

**Cenário 2:** Alterar um pedido existente  
**Dado que** desejo alterar um pedido existente  
**E** o sistema abre o modal com os dados do pedido escolhido após a opção ser selecionada para que possa ser modificada  
**Quando** envio os dados com as opções modificadas  
**Então** o sistema altera os dados do pedido escolhido  
**E** apresenta a mensagem de "Pedido editado com sucesso"

**Cenário 3:** Consultar pedidos  
**Dado que** desejo consultar os pedidos criados  
**Quando** clico na opção do menu "Pedidos"  
**Então** o sistema abre a lista com todos os pedidos que estão cadastrados  
**E** posso visualizar a lista de todos os pedidos

**Cenário 4:** Consultar detalhes de um pedido  
**Dado que** desejo visualizar os detalhes de um pedido  
**E** o sistema abre a lista de todos os pedidos  
**Quando** clico na opção de detalhes de um pedido específico  
**Então** o sistema exibe um modal com os detalhes do pedido selecionado  
**E** eu posso visualizar com detalhes o pedido

**Cenário 5:** Alterar status do pedido  
**Dado que** desejo alterar o status de um pedido  
**E** o sistema exibe um modal com os detalhes do pedido selecionados juntamente de qual o próximo status a alterar  
**Quando** clico na opção de alterar status  
**Então** o sistema altera o status do pedido  
**E** apresenta a mensagem de "Status do Pedido alterado com sucesso!"

**Cenário 6:** Alterar status de pagamento do pedido  
**Dado que** desejo alterar o status de pagamento de um pedido  
**E** o sistema exibe um modal com os detalhes do pedido selecionados juntamente de qual o próximo status a alterar  
**Quando** clico na opção de alterar status de pagamento  
**Então** o sistema altera o status de pagamento do pedido  
**E** apresenta a mensagem de "Status de pagamento alterado com sucesso!"

FONTE: O autor (2022).

FIGURA 39 - História de Usuário "Manutenção de Produção"



**US - Manutenção de Produção**  
na lista [Histórias de Usuário](#)

**Descrição** [Editar](#)

**Como:** um administrador do sistema  
**Quero:** adicionar e consultar as produções  
**Para:** que eu possa incrementar o estoque e armazenar o histórico de produção de cerveja

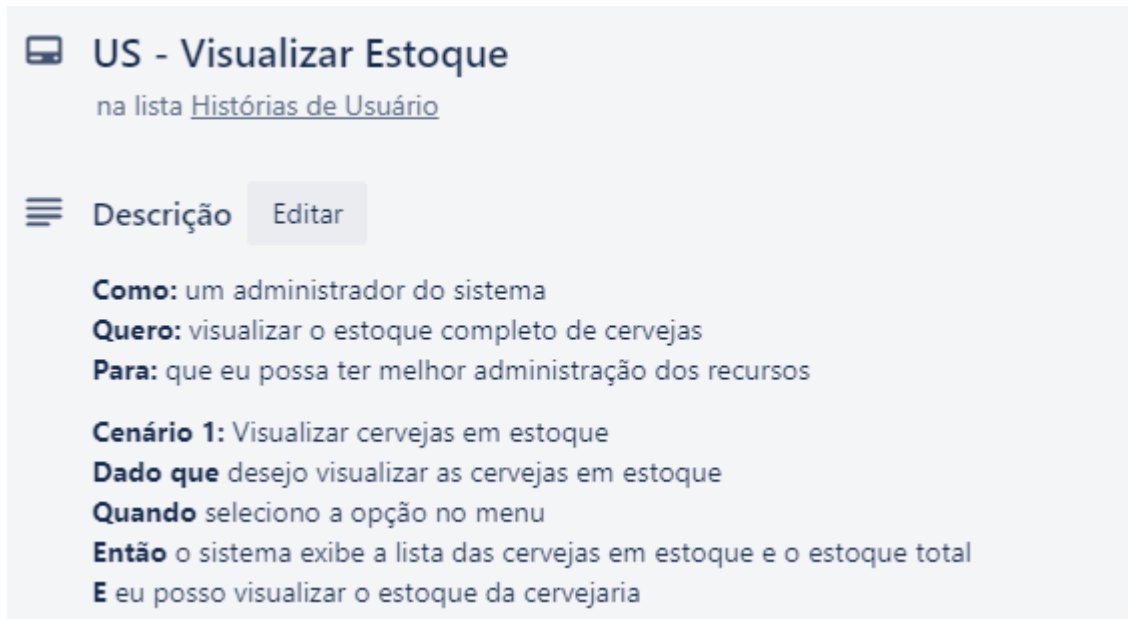
**Cenário 1:** Adicionar nova produção  
**Dado que** desejo adicionar uma nova produção  
**E** o sistema abre o modal de cadastro após a opção ser selecionada  
**Quando** envio os dados referentes à produção  
**Então** o sistema grava a produção  
**E** apresenta a mensagem de "Produção lançada com sucesso"

**Cenário 2:** Consultar produções  
**Dado que** desejo consultar as produções cadastradas  
**Quando** clico na opção do menu "Produção"  
**Então** o sistema abre a lista com todas as produções que estão cadastradas  
**E** posso visualizar a lista de todas as produções

**Cenário 3:** Consultar detalhes de uma produção  
**Dado que** desejo visualizar os detalhes de uma produção  
**E** o sistema abre a lista de todas as produções  
**Quando** clico na opção de detalhes de uma produção específica  
**Então** o sistema exibe um modal com os detalhes da produção selecionada  
**E** eu posso visualizar com detalhes a produção

FONTE: O autor (2022).

FIGURA 40 - História de Usuário "Visualizar Estoque"



**US - Visualizar Estoque**  
na lista [Histórias de Usuário](#)

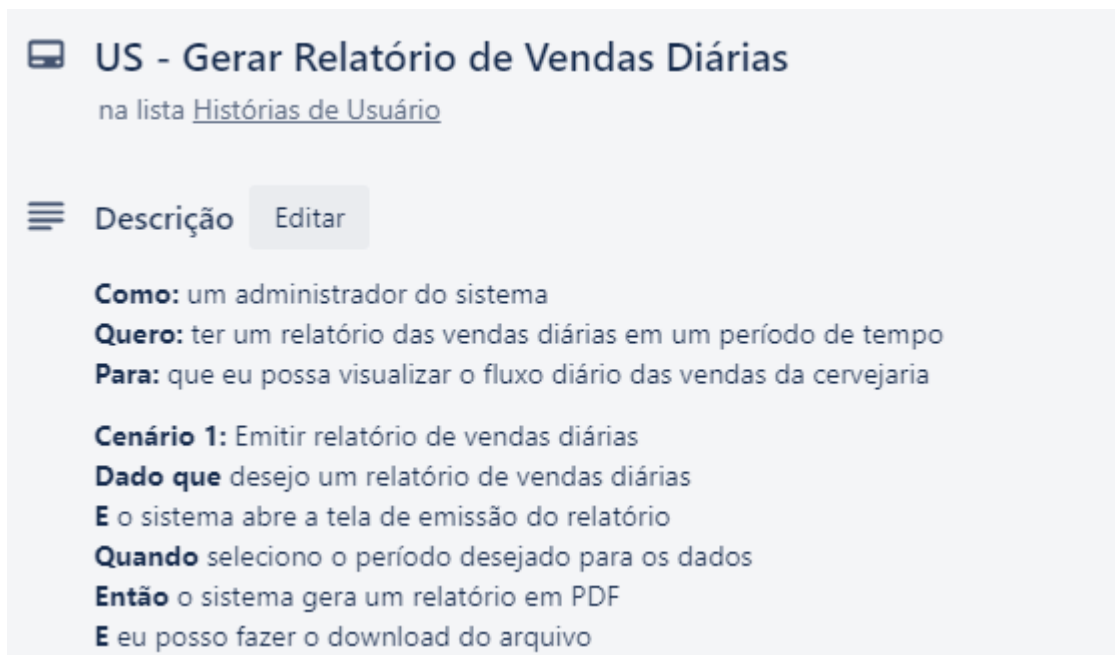
Descrição

**Como:** um administrador do sistema  
**Quero:** visualizar o estoque completo de cervejas  
**Para:** que eu possa ter melhor administração dos recursos

**Cenário 1:** Visualizar cervejas em estoque  
**Dado que** desejo visualizar as cervejas em estoque  
**Quando** seleciono a opção no menu  
**Então** o sistema exibe a lista das cervejas em estoque e o estoque total  
**E** eu posso visualizar o estoque da cervejaria

FONTE: O autor (2022).

FIGURA 41 - História de Usuário "Gerar Relatório de Vendas Diárias"



**US - Gerar Relatório de Vendas Diárias**  
na lista [Histórias de Usuário](#)

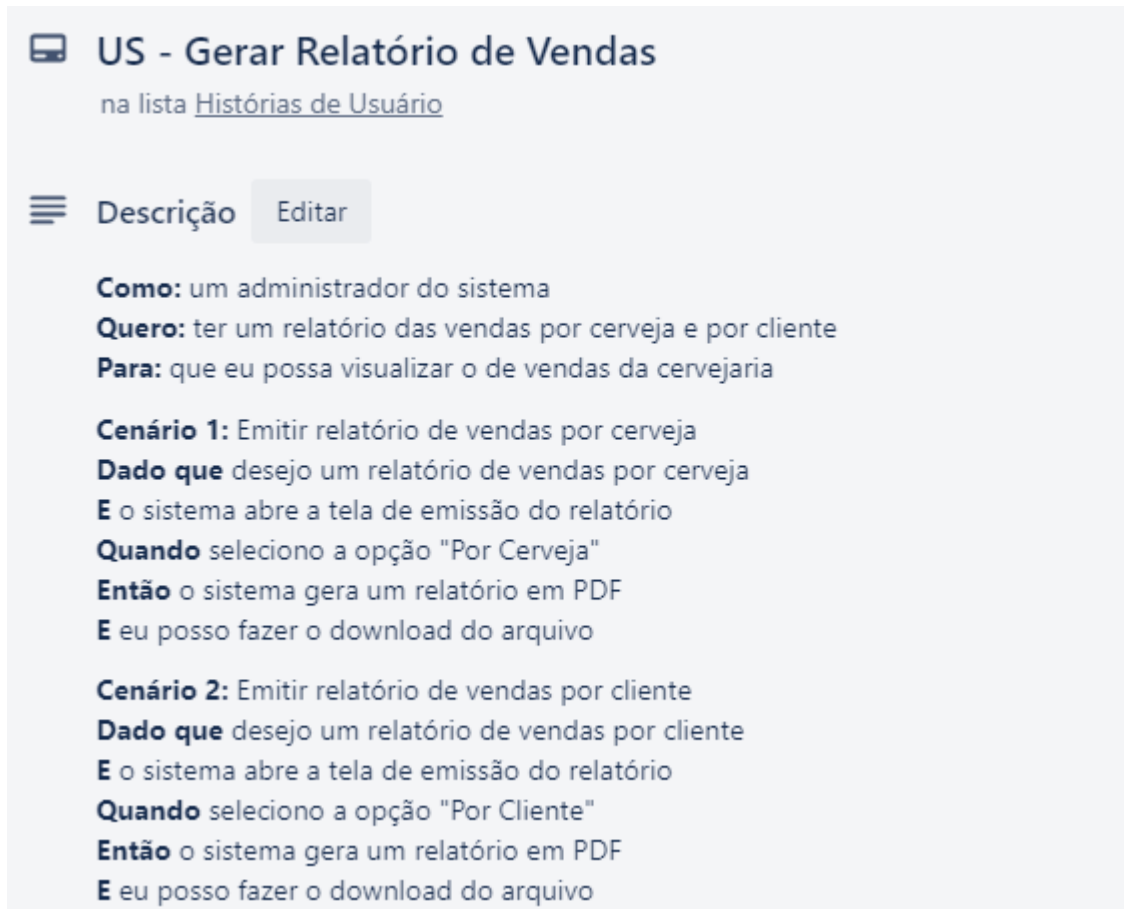
Descrição

**Como:** um administrador do sistema  
**Quero:** ter um relatório das vendas diárias em um período de tempo  
**Para:** que eu possa visualizar o fluxo diário das vendas da cervejaria

**Cenário 1:** Emitir relatório de vendas diárias  
**Dado que** desejo um relatório de vendas diárias  
**E** o sistema abre a tela de emissão do relatório  
**Quando** seleciono o período desejado para os dados  
**Então** o sistema gera um relatório em PDF  
**E** eu posso fazer o download do arquivo

FONTE: O autor (2022).

FIGURA 42 - História de Usuário "Gerar Relatório de Vendas"



**US - Gerar Relatório de Vendas**  
na lista [Histórias de Usuário](#)

**Descrição** [Editar](#)

**Como:** um administrador do sistema  
**Quero:** ter um relatório das vendas por cerveja e por cliente  
**Para:** que eu possa visualizar o de vendas da cervejaria

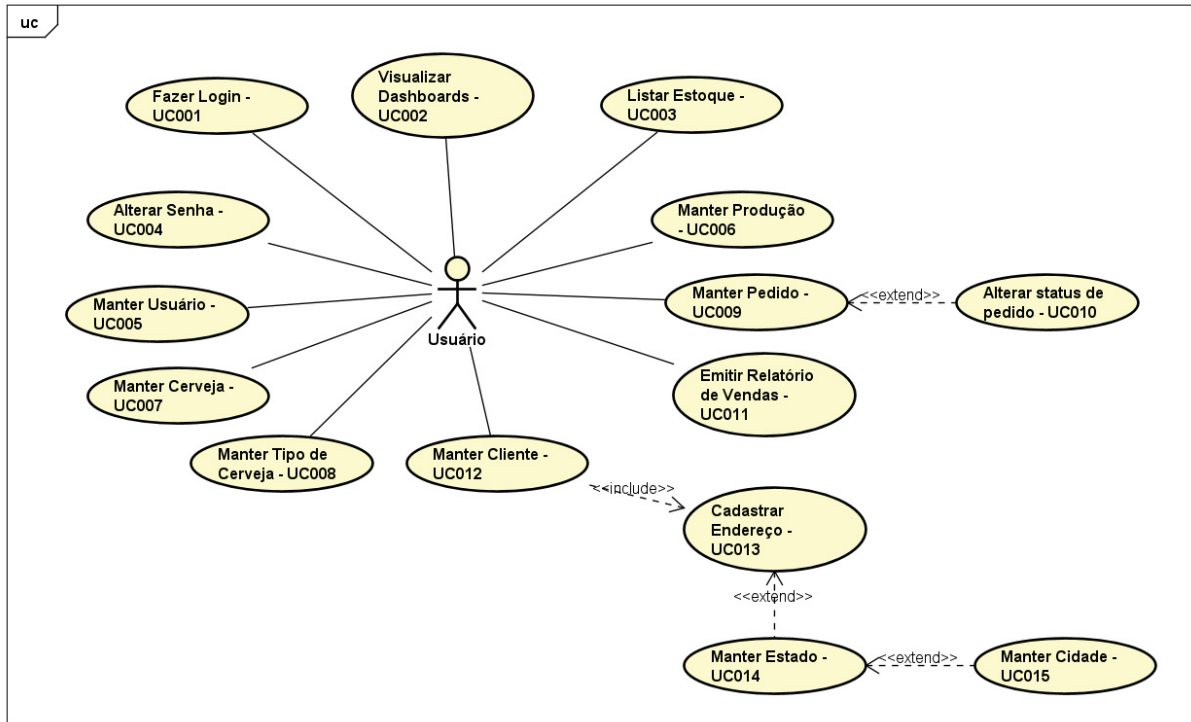
**Cenário 1:** Emitir relatório de vendas por cerveja  
**Dado que** desejo um relatório de vendas por cerveja  
**E** o sistema abre a tela de emissão do relatório  
**Quando** seleciono a opção "Por Cerveja"  
**Então** o sistema gera um relatório em PDF  
**E** eu posso fazer o download do arquivo

**Cenário 2:** Emitir relatório de vendas por cliente  
**Dado que** desejo um relatório de vendas por cliente  
**E** o sistema abre a tela de emissão do relatório  
**Quando** seleciono a opção "Por Cliente"  
**Então** o sistema gera um relatório em PDF  
**E** eu posso fazer o download do arquivo

FONTE: O autor (2022).

## APÊNDICE 2 – DIAGRAMA DE CASOS DE USO

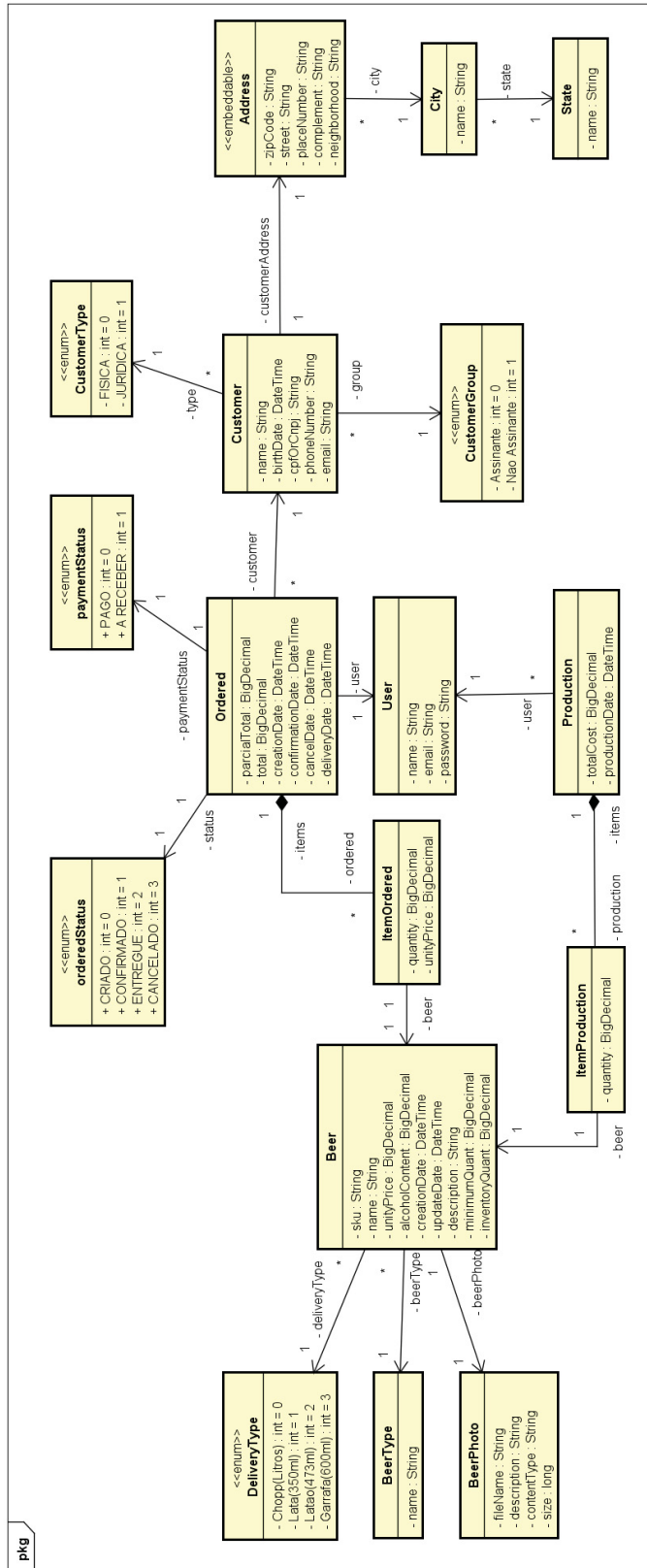
FIGURA 43 - Diagrama de Casos de Uso



FONTE: O autor (2022).

## APÊNDICE 3 – DIAGRAMA DE CLASSES

FIGURA 44 - Diagrama de classes do sistema

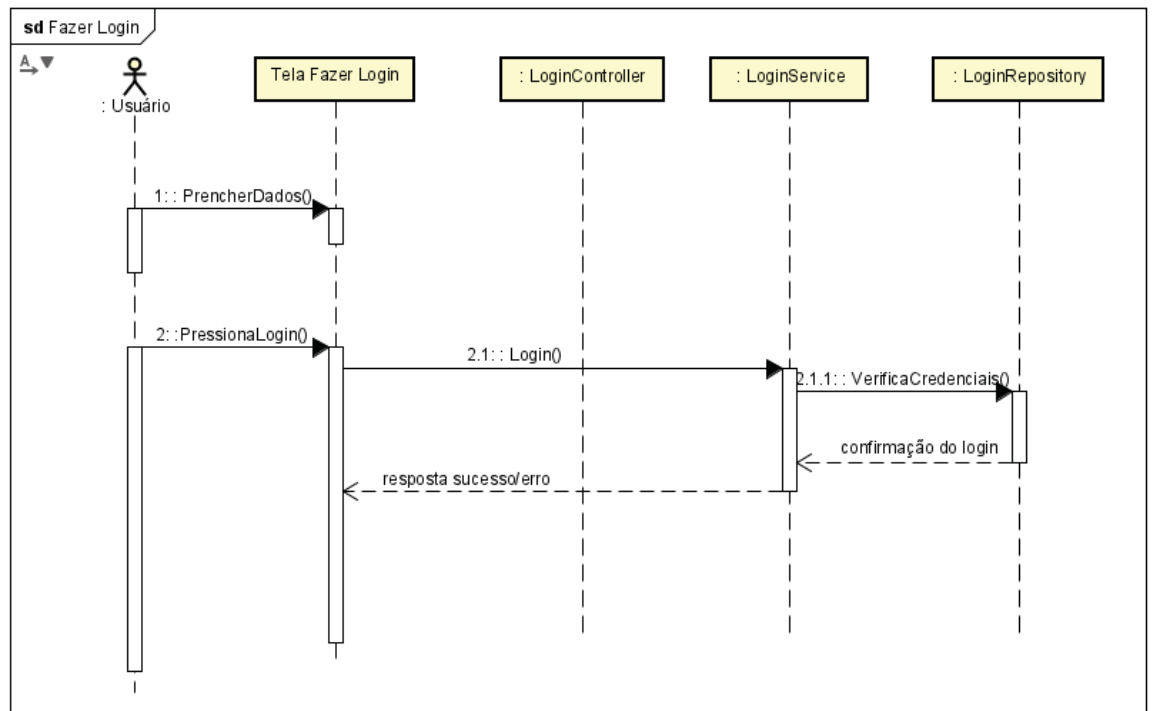


FONTE: O autor (2022).

## APÊNDICE 4 – DIAGRAMAS DE SEQUÊNCIA

No diagrama abaixo, representado pela FIGURA 43, é possível visualizar o fluxo da ação de realizar o *login* de um usuário no sistema.

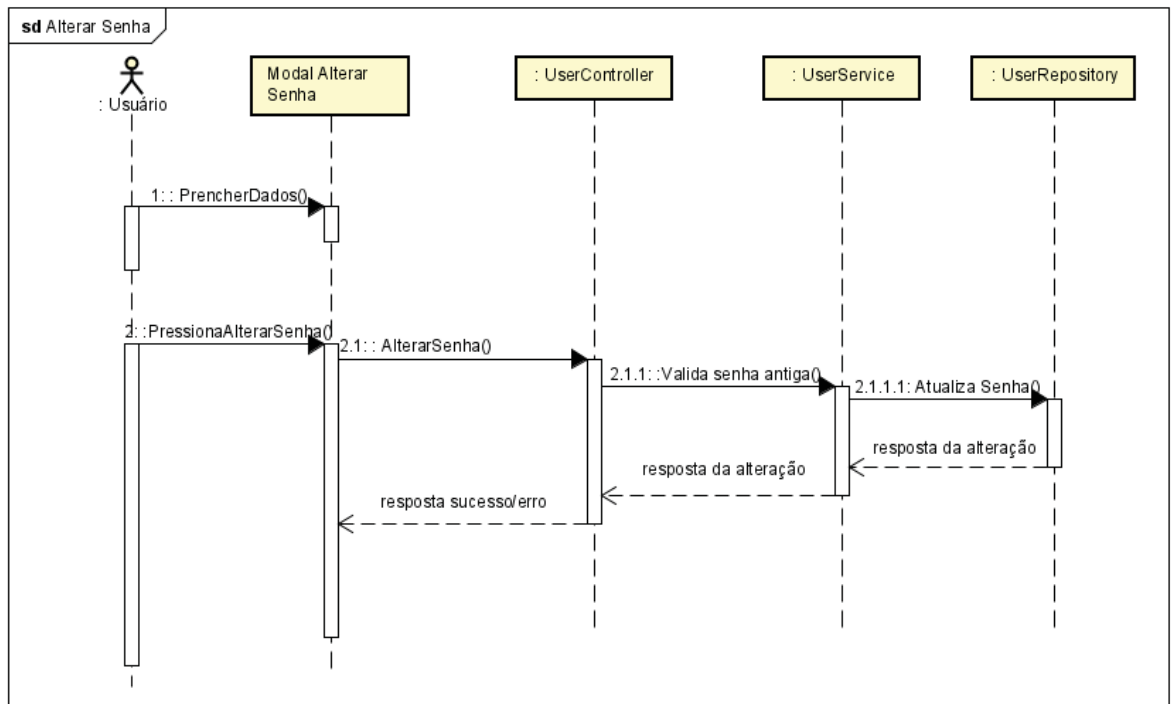
FIGURA 45 - Diagrama de Sequência Login



FONTE: O autor (2022).

A FIGURA 44, a seguir, representa o fluxo de alteração de senha.

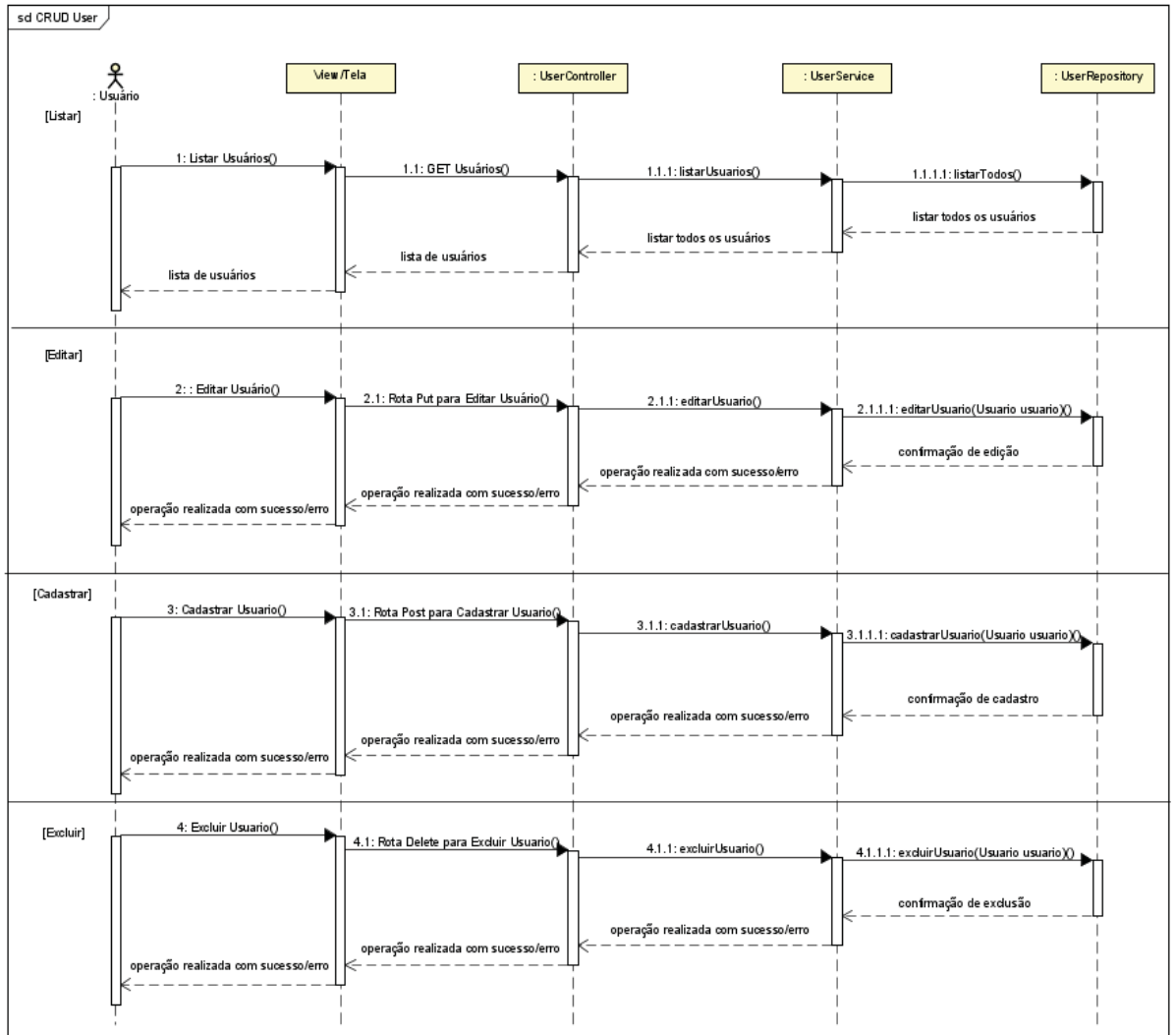
FIGURA 46 - Diagrama de Sequência Alteração de Senha



FONTE: O autor (2022).

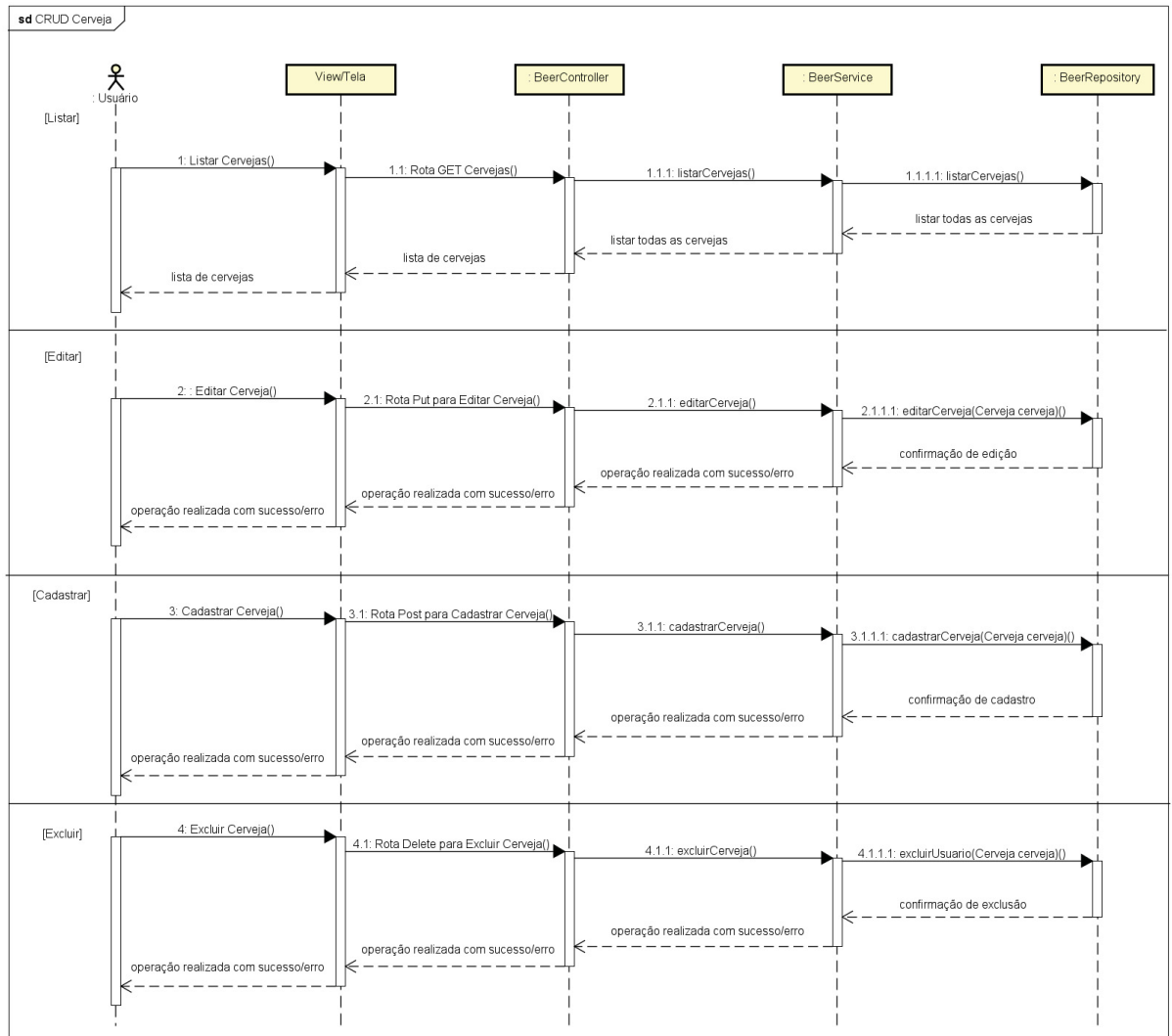
As operações CRUD (create, delete, update) de usuários, cervejas, tipos de cerveja e clientes, estão representados nas FIGURAS 45, 46, 47, 48 respectivamente.

FIGURA 47 - Diagrama de Sequência CRUD de Usuário



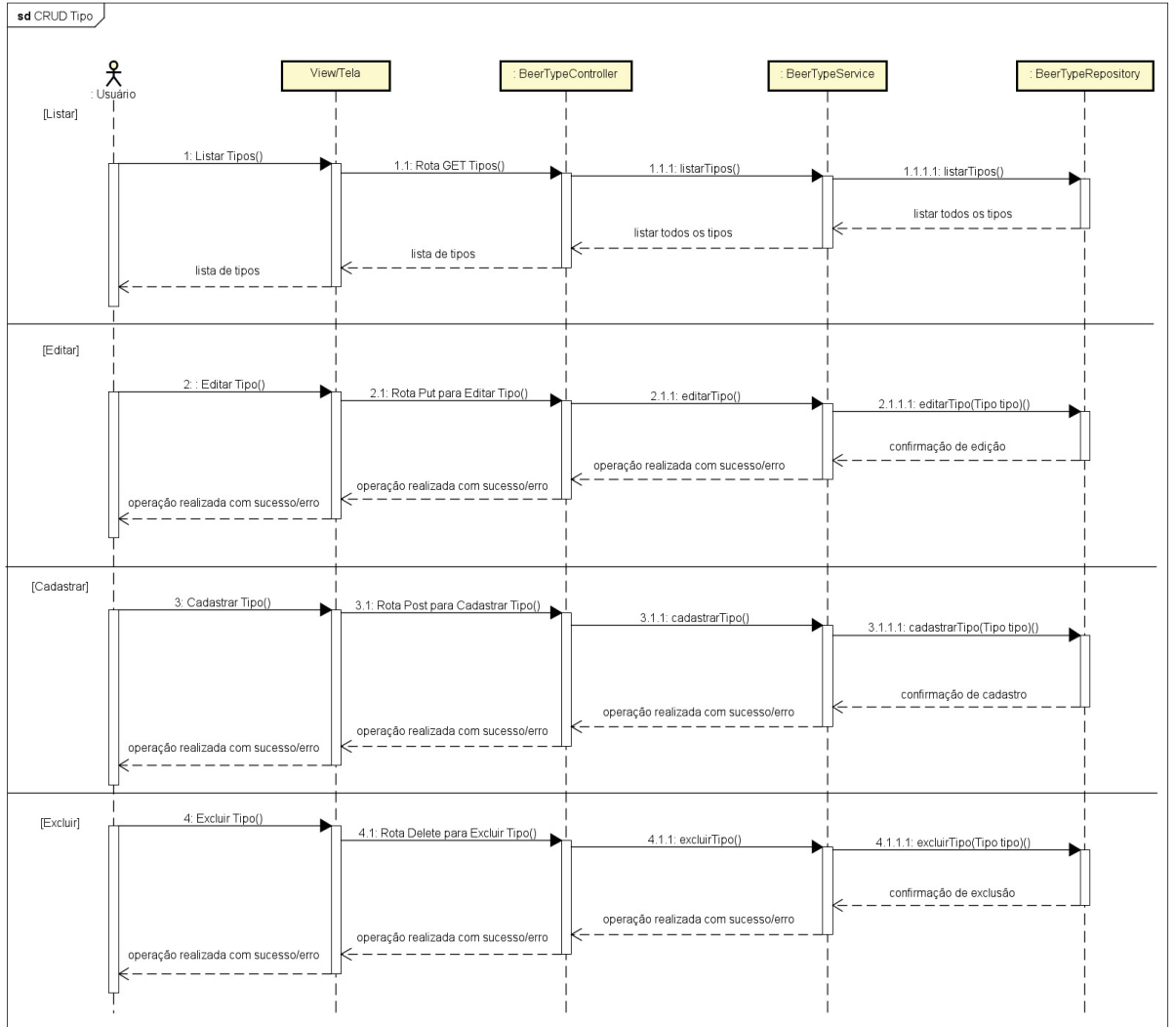
FONTE: O autor (2022).

FIGURA 48 - Diagrama Sequência CRUD de Cerveja



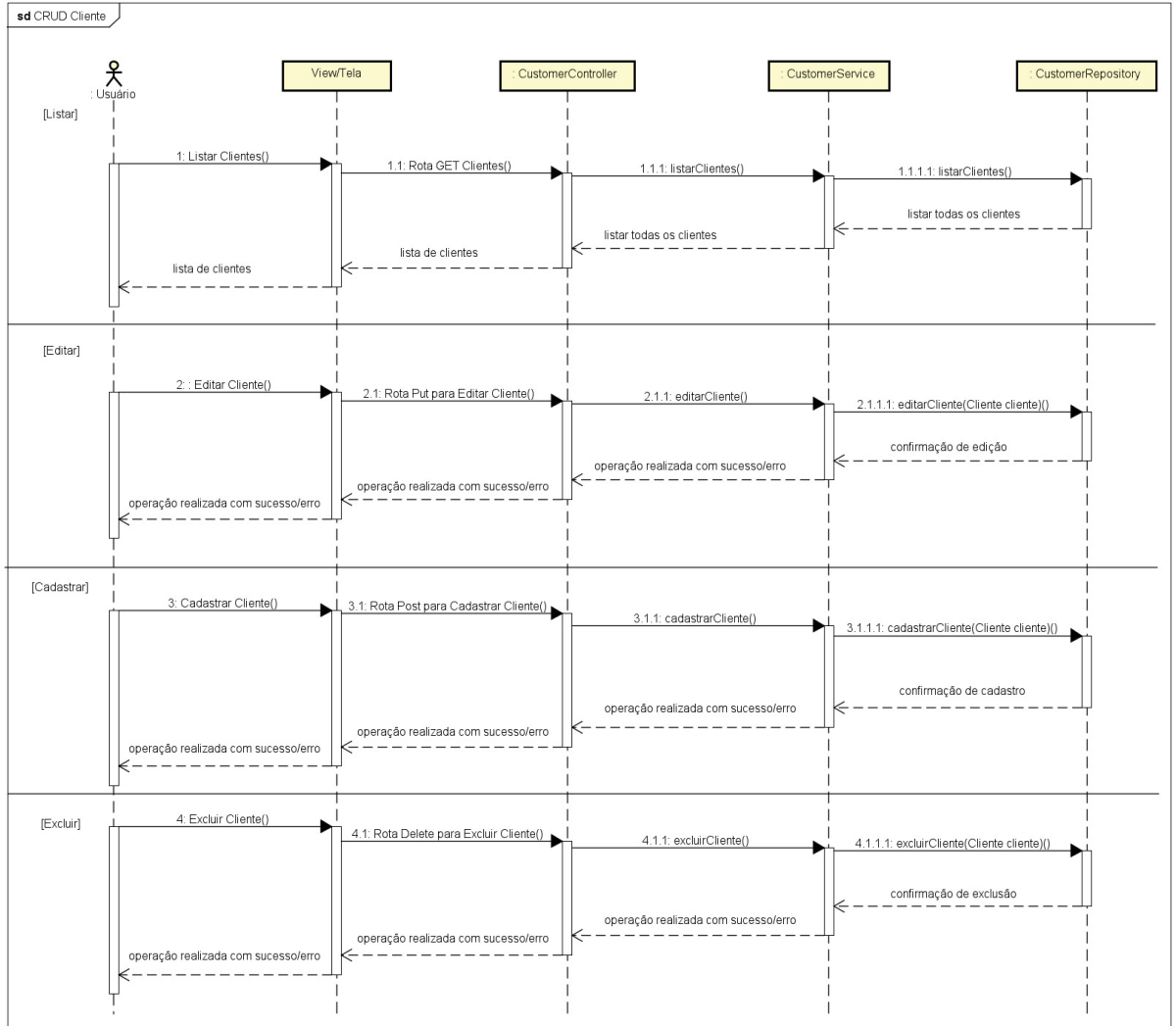
FONTE: O autor (2022).

FIGURA 49 - Diagrama de Sequência CRUD de Tipos de Cerveja



FONTE: O autor (2022).

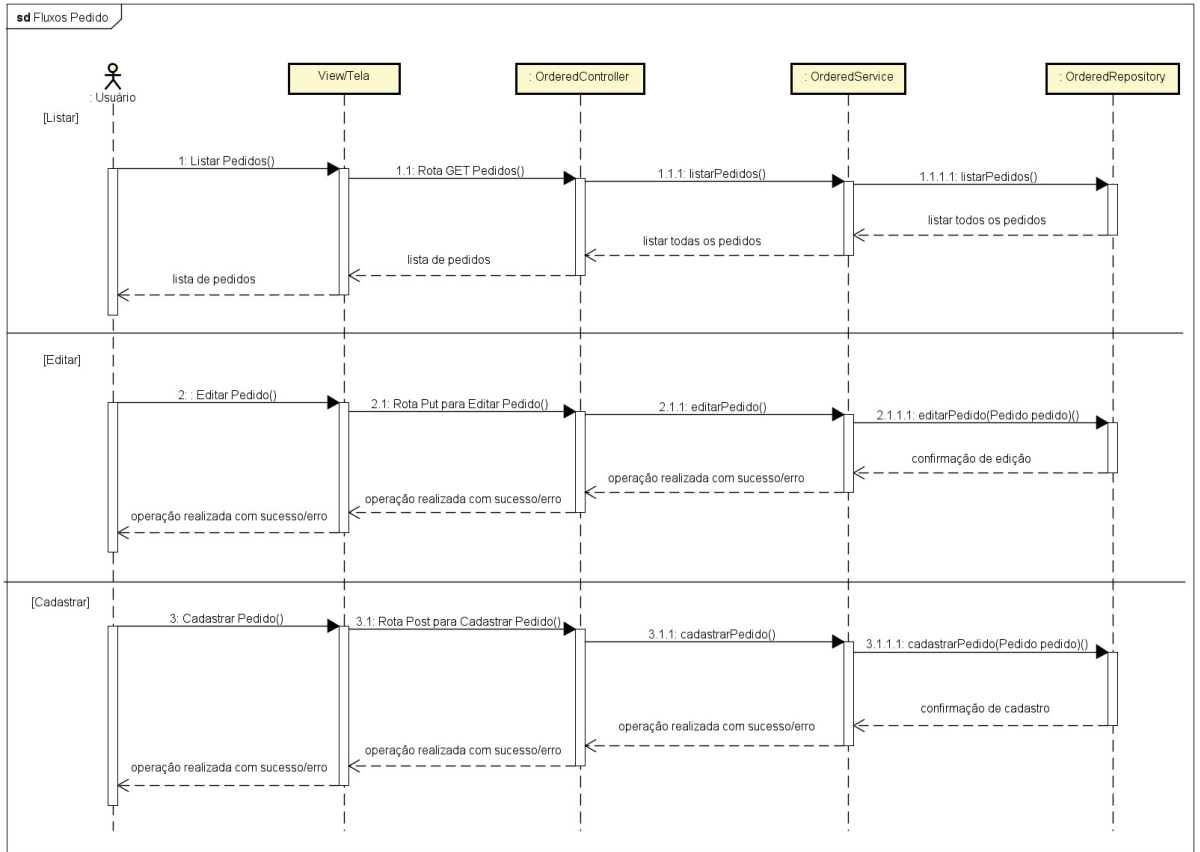
FIGURA 50 - Diagrama de Sequência CRUD de Cliente



FONTE: O autor (2022).

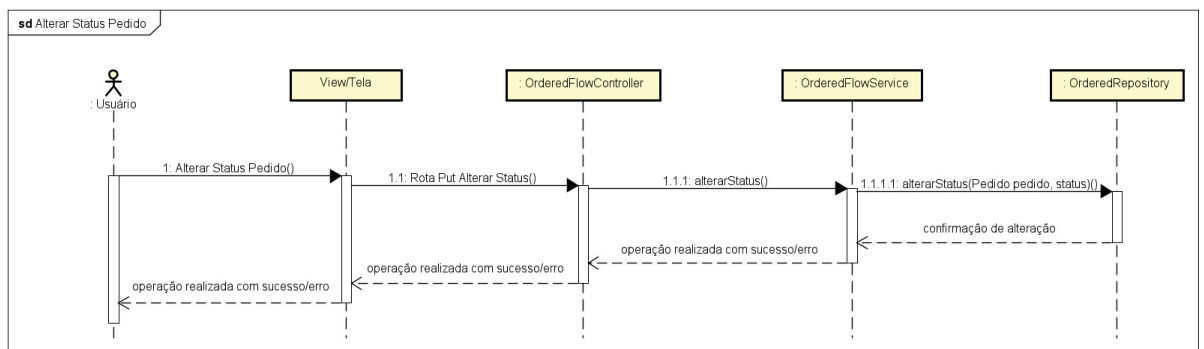
As FIGURAS 49 e 50, representam respectivamente os fluxos de criação, deleção, edição de pedidos e alteração de status.

FIGURA 51 - Diagrama de Sequencia Fluxo de Pedidos



FONTE: O autor (2022).

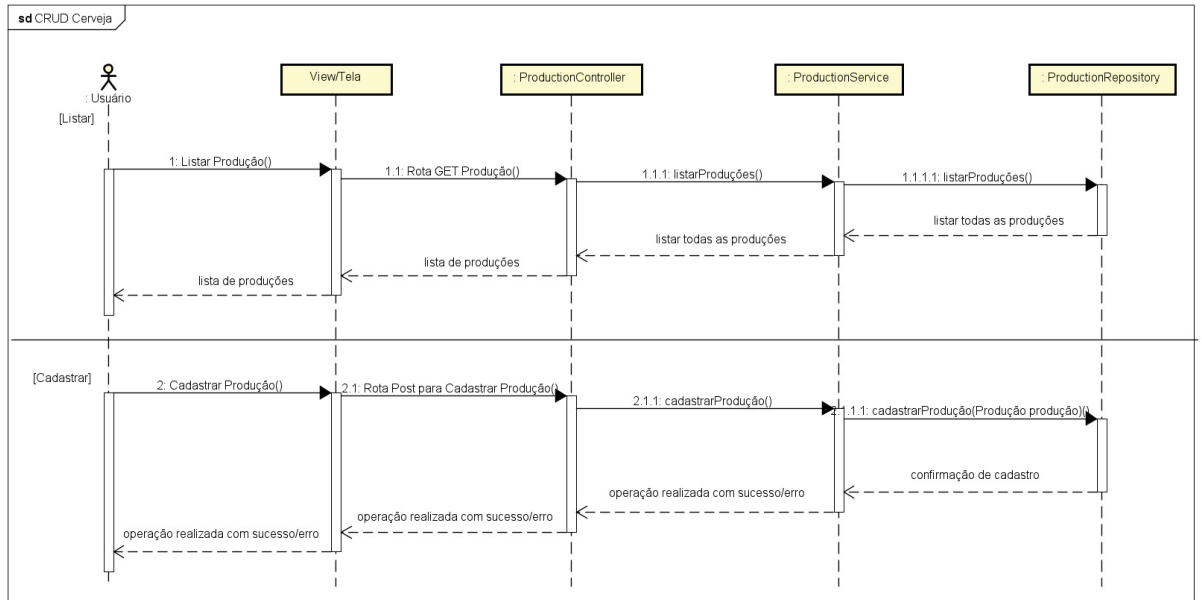
FIGURA 52 - Diagrama de Sequência Alterar Status de Pedido



FONTE: O autor (2022).

Os fluxos de criação e listagem de lançamento de produção, estão representados no diagrama de seqüência a seguir pela FIGURA 51.

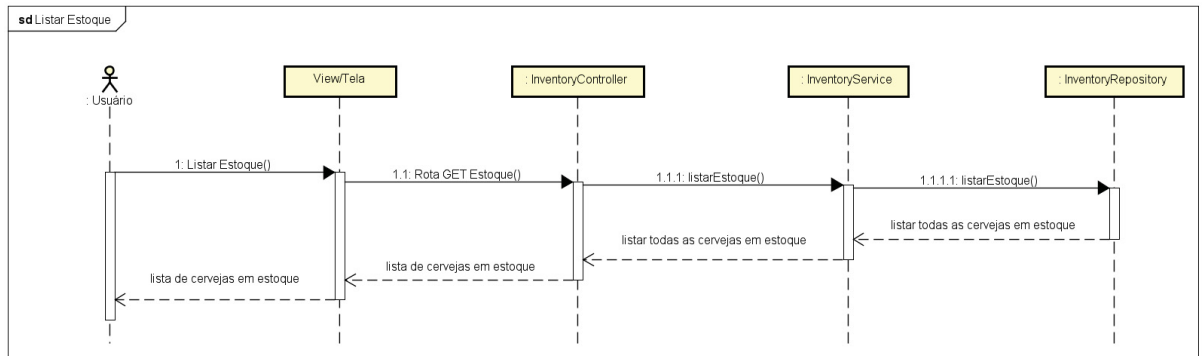
FIGURA 53 - Diagrama de Sequência de Fluxo de Produção



FONTE: O autor (2022).

Na FIGURA 52, é representado o fluxo de listagem de cervejas em estoque.

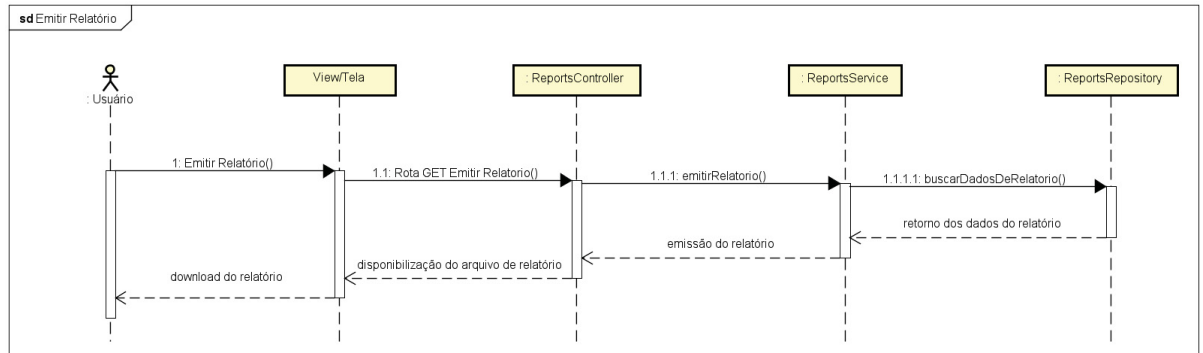
FIGURA 54 - Diagrama de Sequência Listar Estoque



FONTE: O autor (2022).

As emissões de todos os relatórios do sistema seguem o mesmo fluxo, representado a seguir pela FIGURA 53.

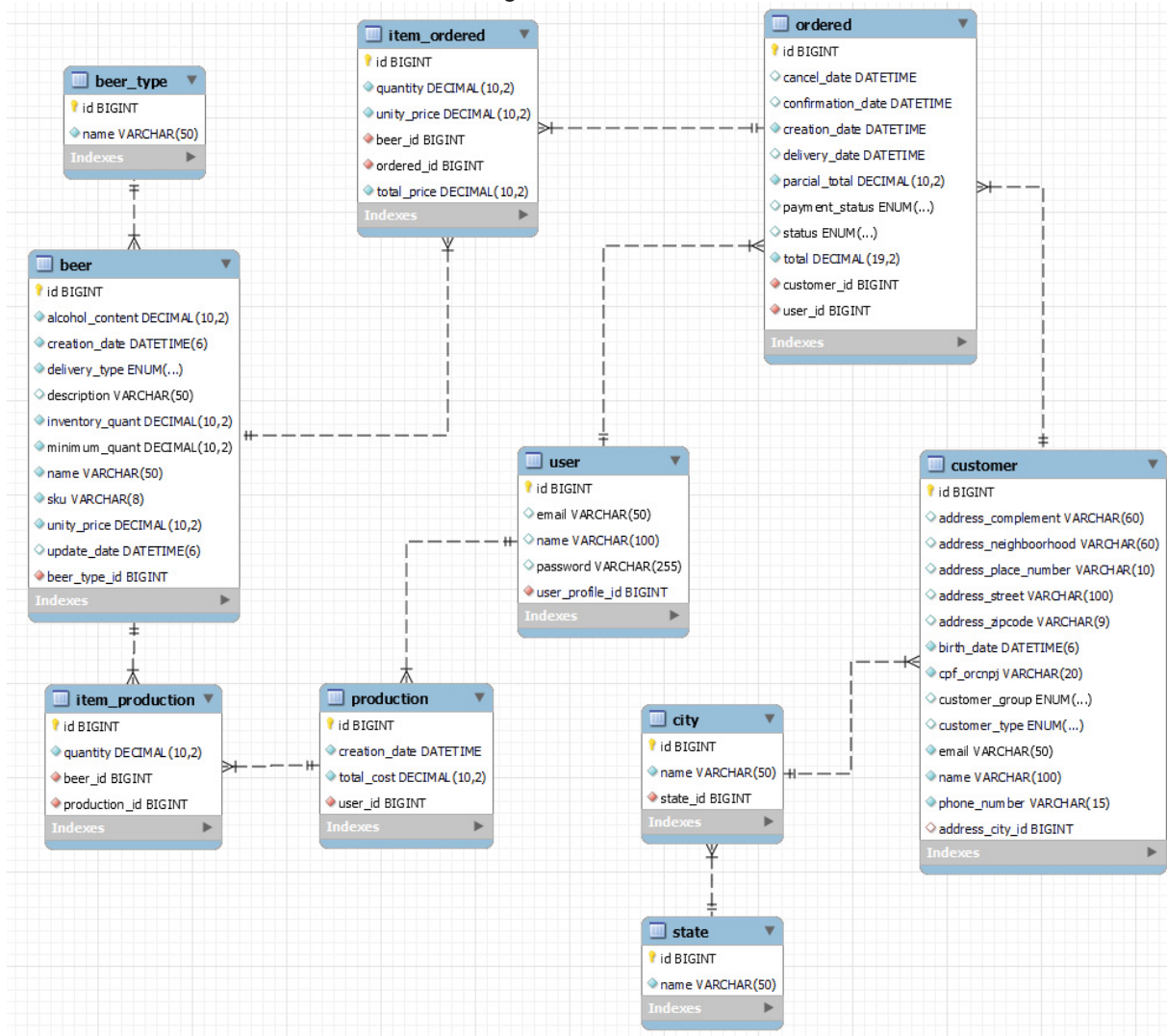
FIGURA 55 - Diagrama de Sequência Emissão de Relatórios



FONTE: O autor (2022).

## APÊNDICE 5 - DIAGRAMA FÍSICO DE BANCO DE DADOS

FIGURA 56 - Diagrama físico de banco de dados



FONTE: O autor (2022).

## APÊNDICE 6 – DOCUMENTAÇÃO API REST (BACKEND)

FIGURA 57 - Recursos disponíveis entidade 'Cerveja'

beer-controller Beer Controller	
GET	/beers listAll
POST	/beers toAdd
GET	/beers/{beerId} findById
PUT	/beers/{beerId} toUpdate
DELETE	/beers/{beerId} toRemove

FONTE: O autor (2022).

FIGURA 58 - Recursos disponíveis entidade 'Tipos de Cerveja'

beer-type-controller Beer Type Controller	
GET	/beertypes listAll
POST	/beertypes toAdd
GET	/beertypes/{beertypeId} findById
PUT	/beertypes/{beertypeId} toUpdate
DELETE	/beertypes/{beertypeId} toDelete
GET	/beertypes/find/{name} listAll

FONTE: O autor (2022).

FIGURA 59 - Recursos disponíveis entidade 'Cliente'

customer-controller Customer Controller	
GET	/customers listAll
POST	/customers save
GET	/customers/{customerId} find
PUT	/customers/{customerId} toUpdate
DELETE	/customers/{customerId} delete
GET	/customers/filter/subs filterAssinantes

FONTE: O autor (2022).

FIGURA 60 - Recursos disponíveis entidade 'Usuário'

user-controller User Controller	
GET	/users listAll
POST	/users toAdd
GET	/users/{userId} findById
PUT	/users/{userId} toUpdate
PUT	/users/{userId}/password changePassword

FONTE: O autor (2022).

FIGURA 61 - Recursos disponíveis entidade 'Estado'

state-controller State Controller	
GET	/states listAll
POST	/states toAdd
GET	/states/{stateId} findById
PUT	/states/{stateId} toUpdate
DELETE	/states/{stateId} toDelete

FONTE: O autor (2022).

FIGURA 62 - Recursos disponíveis entidade 'Cidade'

city-controller City Controller	
GET	/cities listAll
POST	/cities toAdd
GET	/cities/{cityId} findById
PUT	/cities/{cityId} toUpdate
DELETE	/cities/{cityId} toRemove

FONTE: O autor (2022).

FIGURA 63 - Recursos disponíveis entidade 'Pedido'

ordered-controller Ordered Controller	
GET	/ordereds listAll
POST	/ordereds toAdd
GET	/ordereds/{orderId} findById
GET	/ordereds/filter filter

FONTE: O autor (2022).

FIGURA 64 - Recursos disponíveis fluxo de status de pedido

ordered-flow-controller Ordered Flow Controller	
PUT	/ordereds/{orderId}/cancel toCancel
PUT	/ordereds/{orderId}/confirmation toConfirm
PUT	/ordereds/{orderId}/delivery toDelivery
PUT	/ordereds/{orderId}/paid toPaid

FONTE: O autor (2022).

FIGURA 65 - Recursos disponíveis entidade 'Produção'

production-controller Production Controller	
GET	/productions listAll
POST	/productions toAdd
GET	/productions/{productionId} findById
GET	/productions/filter filter
GET	/productions/teste/{id} items

FONTE: O autor (2022).

FIGURA 66 - Recursos disponíveis 'Estoque'

inventory-controller Inventory Controller	
GET	/inventory/1 hasInventory

FONTE: O autor (2022).

FIGURA 67 - Recursos disponíveis *dashboard/home page*

dashboard-controller Dashboard Controller	
GET	/dashboards/customers countCustomers
GET	/dashboards/inventoryTotal inventoryQuantity
GET	/dashboards/monthSales monthSales
GET	/dashboards/openOrdereds openOrdered
GET	/dashboards/salesBeer salesPerBeer
GET	/dashboards/salesBeerType salesPerBeerType
GET	/dashboards/yearSales yearSales

FONTE: O autor (2022).

FIGURA 68 - Recursos disponíveis relatórios

reports-controller Reports Controller	
GET	/reports/beer-sales beerSalesPdf
GET	/reports/customer-sales customerSalesPdf
GET	/reports/daily-sales consultDailySalesPdf

FONTE: O autor (2022).