

**DAS** Departamento de Automação e Sistemas  
**CTC** Centro Tecnológico  
**UFSC** Universidade Federal de Santa Catarina

# Sistema de apoio à gestão de produção baseado em estoque e ativos retornáveis para microcervejarias

*Relatório submetido à Universidade Federal de Santa Catarina  
como requisito para a aprovação da disciplina:  
DAS 5511: Projeto de Fim de Curso*

*Fabício da Silva Hoepers*

*Florianópolis, Janeiro de 2018*



# Sistema de apoio à gestão de produção baseado em estoque e ativos retornáveis para microcervejarias

*Fabício da Silva Hoepers*

Esta monografia foi julgada no contexto da disciplina

**DAS 5511: Projeto de Fim de Curso**

e aprovada na sua forma final pelo

**Curso de Engenharia de Controle e Automação**

*Prof. Ricardo José Rabelo*

---

Banca Examinadora:

Prof. Ricardo José Rabelo/UFSC  
Orientador na Empresa

Prof. Ricardo José Rabelo  
Orientador no Curso

Prof. Hector Bessa Silveira  
Responsável pela disciplina

Prof. Leandro Buss Becker, Avaliador

Renan Grzebieluckas, Debatedor

Mateus Abreu de Andrade, Debatedor

# Agradecimentos

Este trabalho é fruto do apoio e esforço de muitas pessoas importantes na minha vida.

Primeiramente gostaria de agradecer aos meus pais por todo o suporte e apoio dado durante toda minha vida acadêmica até aqui, bem como na vida fora da academia também. Em nenhum momento eles mediram esforços para que eu conseguisse atingir meus objetivos, seja de forma financeira ou na forma de conselhos, incentivos ou partilha de conhecimentos.

Agradeço também à minha namorada, Alice, por todo suporte dado a mim durante a graduação, entendimento de noites mal dormidas e programas nossos cancelados por causa de estudos ou trabalhos, e incentivos nos momentos desanimadores do curso. Sua companhia foi fundamental para que eu chegasse até onde estou agora e acredito que será fundamental no futuro a nossa frente também.

Agradeço ainda ao Departamento de Automação e Sistemas, e a UFSC, pela formação e oportunidades, concedidas a mim durante esses 7 anos de graduação, que me fizeram crescer como pessoa e profissional. Aqui cabe um agradecimento especial à alguns professores que marcaram minha passagem pela graduação, dentre eles, o professor Julio e suas provas difíceis porém sempre disposto a ajudar os alunos e repetir conceitos vistos em sala, ao professor Hector sempre disposto a auxiliar os alunos nos mais diversos assuntos, dentro e fora de sala, ao professor Rômulo também por conseguir resgatar meu interesse pela área de informática com suas aulas sempre muito elucidativas, e também ao professor Ricardo José Rabelo, com quem eu possuo uma grande afinidade com a área de atuação e que também aceitou gentilmente ser orientador deste trabalho estando sempre disposto a dar conselhos e realizar reuniões até que chegássemos à melhor solução possível do problema, mesmo que estivesse extremamente ocupado com outros assuntos.

Também vale ressaltar a importância de alguns órgãos externos ao curso que contribuíram para meu crescimento pessoal e profissional, e onde fiz grandes amigos, com os quais enfrentei dificuldades dos mais diversos tipos, como Autojun, CAECA, Organizações dos Linguíçoes da Automação da qual fiz parte e das organizações de Trotes Integrados do CTC que fiz parte.

Cabe agradecimento especial aos amigos Rafael Jung e Rafael Scheffer, que por já terem certa experiência com Ruby on Rails me ajudaram em algumas vezes para encontrar e solucionar erros. Também aos amigos Bruno Gustavo, Guilherme Perosa, Nicolas Gorgulho, Arthur Lima pelos jogos de Dota 2 que ajudaram a manter a sanidade e a espalhar diversão diversas vezes durante o curso.

Por último, mas não menos importante, gostaria de agradecer aos inúmeros amigos que fiz durante o decorrer deste curso e que pretendo levar para a minha vida adiante, e que me ajudaram de uma forma ou de outra, durante ou antes deste projeto, como Fernando Redondo, Fernando Barbosa, Paulo Curado, Renan Daquina, Gislaine Hoffman, Alline Rolim, Rodrigo Gesser, Adriano Brandão, Marina Stasiak, Rafael Sartori, Thaise Damo, Max Müller, Bruno Battistotti, Renan Grzebieluckas, Andjara Consentino, Eduardo Kneipp, Maria Alice Duz, Henrique Furtado, Eduardo Barbeta, Vincenzo Benetti, Artur Cook, Caio Romam, Fernanda Ortolan, Pedro Casali, Bruno Sechini, Kaio Siqueira, Georgiy Tanca, Lucas Testoni, Leandro Vahelle, Bruno Lima, Amanda Brinhosa, Rafael Nunes, Gustavo Gonçalves, Diandra Moraes, Thiago Bertoldi, Pedro Neves, Marco Geremias, Bruno Bay, Antonio Sandri dentre tantos outros.

# Resumo

O mercado brasileiro de cervejas é o terceiro maior do mundo, ficando atrás apenas de China e EUA e nos últimos anos vem crescendo exponencialmente com a abertura de novas microcervejarias, chegando ao impressionante número de 148 novas microcervejarias abertas em 2016 segundo o MAPA - Ministério da Agricultura, Pecuária e Abastecimento. Apesar do crescimento vertiginoso, há pouco conhecimento na área, até mesmo por parte dos gestores dessas novas microcervejarias. Isso acarreta em uma carência de serviços voltados para esse mercado, em todos os níveis, desde a logística com fornecedores até a logística interna de estoque, por exemplo. Dentre esses problemas, temos a logística interna de estoques e o controle de ativos retornáveis como problemas que são chave para a tomada de decisões da fábrica e por isso serão foco deste trabalho. Para isso propôs-se o desenvolvimento de um sistema web de apoio à gestão de produção baseado em estoque e ativos retornáveis para microcervejarias. Esse sistema foi desenvolvido através de metodologia iterativa e orientação a serviços, utilizando Ruby on Rails como linguagem de programação, em uma microcervejaria parceira da região de Florianópolis. O sistema utiliza formulários de entrada de dados e também leitores de código de barras para colher os dados dos estoque primário e final da microcervejaria, fornecendo assim dados para posterior análise de indicadores de desempenho da fábrica. Além disso, o sistema, da mesma forma, fornece dados sobre os ativos retornáveis da microcervejaria, como barris, chopeiras e cilindros de gás.

**Palavras-chave:** Microcervejaria. Sistema web, SaaS. Ruby on Rails.





# Abstract

The Brazilian beer market is the third large of the world, being behind of China and USA only and in the last few years keep growing exponentially with the opening of new microbreweries, achieving the stunning number of 148 new microbreweries opened in 2016 according to MAPA - Department of Agriculture, Livestock and Food Supply. Despite the vertiginous growth, there is few knowledge in the field, even from the managers of these new microbreweries. This results in a lack of services turned to this market, in every level, since the logistics involving suppliers to the internal logistic of the brewery itself, like inventory for example. Among these problems, we have inventory and returnable assets control as problems that are very important to the correct decisions to factory and because of that are focus of this work. For that, it was proposed the development of a web system that supports management of the production based on inventory and returnable assets to microbreweries. This system was developed with iterative methodology and service orientation, using Ruby on Rails as programming language, in a partner brewery from Florianópolis. The system uses forms and barcode code readers to take the data from the primary inventory and the final inventory from the microbrewery, thus providing data for a posterior analysis of performance indicators of the factory. Furthermore, the system also provide data about the returnable assets of the microbrewery, like barrels, choppers and gas cylinders.

**Keywords:** Microbrewery. Web system, Saas. Ruby on Rails.



# Lista de ilustrações

Figura 1 – Número de cervejarias no Brasil por ano . . . . .	15
Figura 2 – Logotipo Cervejaria Kairós . . . . .	17
Figura 3 – Exemplo de Metodologia em Cascata . . . . .	24
Figura 4 – Exemplo de Metodologia Iterativa Incremental . . . . .	25
Figura 5 – Exemplo de Banco de Dados Relacional . . . . .	26
Figura 6 – Exemplo de código de barras . . . . .	27
Figura 7 – Diagrama MVC . . . . .	28
Figura 8 – Diagrama MVC para o caso de um sistema com Rails . . . . .	28
Figura 9 – Arquitetura geral do sistema a ser desenvolvido . . . . .	37
Figura 10 – Esquema da comunicação entre leitor de código de barras, tablet e sistema . . . . .	38
Figura 11 – Diagrama de uso do serviço 1 . . . . .	39
Figura 12 – Exemplo de diagrama de sequência pra caso de uso do serviço 1 . . . . .	40
Figura 13 – Diagrama de uso do serviço 2 . . . . .	41
Figura 14 – Exemplo de diagrama de sequência pra caso de uso do serviço 2 . . . . .	43
Figura 15 – Diagrama de uso do serviço 3 . . . . .	44
Figura 16 – Exemplo de diagrama de sequência pra caso de uso do serviço 3 . . . . .	46
Figura 17 – Logotipo do PaaS Heroku . . . . .	47
Figura 18 – Logotipo do Ruby on Rails . . . . .	48
Figura 19 – Modelo Conceitual simplificado do banco de dados . . . . .	49
Figura 20 – Modelo lógico do banco de dados . . . . .	50
Figura 21 – Fluxograma demonstrando funcionamento de login e logout . . . . .	53
Figura 22 – Fluxograma demonstrando funcionamento do cadastro de matérias primas . . . . .	55
Figura 23 – Fluxograma demonstrando funcionamento do cadastro de produtos . . . . .	56
Figura 24 – Template de Dashboard Gentellela on Rails . . . . .	60
Figura 25 – Tela inicial com menus laterais sobre os serviços . . . . .	63
Figura 26 – Tela exemplo do funcionamento dos cadastros, sem menus a mostra . . . . .	63
Figura 27 – Tela exemplo do Serviço 1, de visualização dos indicadores, sem menus a mostra. . . . .	64
Figura 28 – Tela exemplo do Serviço 2, de atualização do nível do estoque final atual, utilizando leitor de código de barras como entrada de dados. . . . .	65
Figura 29 – Tela exemplo da visualização de receitas com seus respectivos ingredientes . . . . .	66
Figura 30 – Tela exemplo de cadastro e edição de receitas cadastradas pela cervejaria . . . . .	66
Figura 31 – Tela exemplo do Serviço 3, de visualização da localização e estado dos barris, de quais barris estão fora da fábrica a mais tempo e também sobre quantidade de barris em cada localização geral e por capacidade deles, sem menus a mostra. . . . .	67

Figura 32 – Tela exemplo do Serviço 1, de visualização do nível do estoque atual, juntamente, com receitas que podem ou não ser produzidas, sem menus a mostra. . . . .	68
Figura 33 – Tela exemplo do Serviço 1, de visualização das receitas que podem ou não ser produzidas, sem menus a mostra, a partir da tela de celular. . .	69

# Lista de abreviaturas e siglas

Amazon EC2 - Amazon Elastic Compute Cloud

CSS - Cascading Style Sheets

EAN-13 - European Article Number-13

ERP - Enterprise Resource Planning

GS1 Brasil - Associação Brasileira de Automação

HTML - HyperText Markup Language

IaaS - Infrastructure as a Service

MAPA - Ministério da Agricultura, Pecuária e Abastecimento

MVC - Model, Control, View

PaaS - Plataform as a Service

SaaS - Software as a Service

Sicobe - Sistema de Controle de Produção de Bebidas

SQL - Structured Query Language

UPC-A - Universal Product Code-A



# Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>15</b>
<b>1.1</b>	<b>Contextualização</b>	<b>15</b>
1.1.1	Descrição do problema	15
1.1.2	A empresa	16
<b>1.2</b>	<b>Objetivos do trabalho</b>	<b>17</b>
1.2.1	Objetivo geral	17
1.2.2	Objetivos específicos	18
<b>1.3</b>	<b>Aderência ao curso de Engenharia de Controle e Automação</b>	<b>18</b>
<b>1.4</b>	<b>Estrutura do documento</b>	<b>19</b>
<b>2</b>	<b>FUNDAMENTOS TEÓRICOS E SOFTWARES UTILIZADOS</b>	<b>21</b>
<b>2.1</b>	<b>Avaliação de desempenho</b>	<b>21</b>
2.1.1	Indicadores de desempenho	21
<b>2.2</b>	<b>Software como serviço</b>	<b>22</b>
<b>2.3</b>	<b>Metodologia de desenvolvimento de software</b>	<b>23</b>
2.3.1	Metodologia cascata	23
2.3.2	Metodologia iterativa incremental	24
2.3.3	Metodologia escolhida	25
<b>2.4</b>	<b>Modelagem de Banco de Dados</b>	<b>25</b>
<b>2.5</b>	<b>Código de barras</b>	<b>26</b>
<b>2.6</b>	<b>MVC</b>	<b>27</b>
<b>3</b>	<b>ESPECIFICAÇÃO E PROJETO</b>	<b>31</b>
<b>3.1</b>	<b>Definição dos serviços que seriam desenvolvidos</b>	<b>31</b>
<b>3.2</b>	<b>Levantamento de indicadores</b>	<b>32</b>
<b>3.3</b>	<b>Requisitos Funcionais e Não-funcionais</b>	<b>33</b>
3.3.1	Usabilidade	34
3.3.2	Confiabilidade	34
3.3.3	Desempenho	34
3.3.4	Portabilidade	35
3.3.5	Segurança	35
<b>3.4</b>	<b>Solução escolhida</b>	<b>35</b>
<b>3.5</b>	<b>Aquisição de equipamentos</b>	<b>36</b>
<b>3.6</b>	<b>Modelagem dos serviços</b>	<b>38</b>
3.6.1	Serviço 1: Controle de estoque primário	38
3.6.2	Serviço 2: Controle de estoque final	41

3.6.3	Serviço 3: Ciclo de vida de ativos retornáveis . . . . .	43
<b>3.7</b>	<b>Estudo da linguagem e das tecnologias a serem usadas . . . . .</b>	<b>46</b>
3.7.1	Heroku . . . . .	47
3.7.2	Ruby on Rails . . . . .	48
<b>3.8</b>	<b>Modelagem do banco de dados . . . . .</b>	<b>49</b>
<b>3.9</b>	<b>Desenvolvimento em Ruby on Rails . . . . .</b>	<b>51</b>
3.9.1	DRY . . . . .	51
3.9.2	Convention over Configuration . . . . .	51
3.9.3	Login e sign up . . . . .	52
3.9.4	Funções de administrador . . . . .	53
3.9.5	Serviço 1: Controle de estoque primário . . . . .	54
3.9.6	Serviço 2: Controle de estoque final . . . . .	55
3.9.7	Serviço 3: Ciclo de vida de bens retornáveis . . . . .	56
<b>4</b>	<b>IMPLEMENTAÇÃO DO SISTEMA E RESULTADOS . . . . .</b>	<b>59</b>
<b>4.1</b>	<b>Implementação do sistema . . . . .</b>	<b>59</b>
4.1.1	Escolha e Modelagem do Dashboard . . . . .	60
<b>4.2</b>	<b>Resultados . . . . .</b>	<b>60</b>
<b>5</b>	<b>CONCLUSÃO E PERSPECTIVAS . . . . .</b>	<b>71</b>
	<b>REFERÊNCIAS . . . . .</b>	<b>73</b>
	<b>APÊNDICES . . . . .</b>	<b>75</b>
	<b>APÊNDICE A – EXEMPLOS DE PSEUDO-CÓDIGOS . . . . .</b>	<b>77</b>



# 1 Introdução

O presente capítulo busca apresentar o problema abordado neste trabalho apresentando uma contextualização do problema, causas e consequências do mesmo, e os objetivos que foram buscados durante o desenvolvimentos do projeto. Este capítulo também tem a finalidade de apresentar a estrutura do trabalho e a metodologia utilizadas no desenvolvimento.

## 1.1 Contextualização

No presente item será abordado o problema alvo do projeto.

### 1.1.1 Descrição do problema

O Brasil é um dos maiores consumidores de cerveja do mundo, ficando atrás apenas de China e Estados Unidos. De acordo com o Sistema de Controle de produção de Bebidas da Receita Federal (Sicobe), entre 2005 e 2014, a produção nacional de cervejas cresceu 64%. Grande parte desse crescimento se deve ao desenvolvimento e surgimento de novas microcervejarias espalhadas principalmente nas regiões Sul e Sudeste do país. De acordo com o MAPA, Ministério da Agricultura, Pecuária e Abastecimento, em 2016 foram registradas 148 novas microcervejarias no Brasil, e esse crescimento do número de microcervejarias continua como pode ser visto em gráfico, figura 1, gerado a partir de dados do relatório do MAPA de 2017 [1].

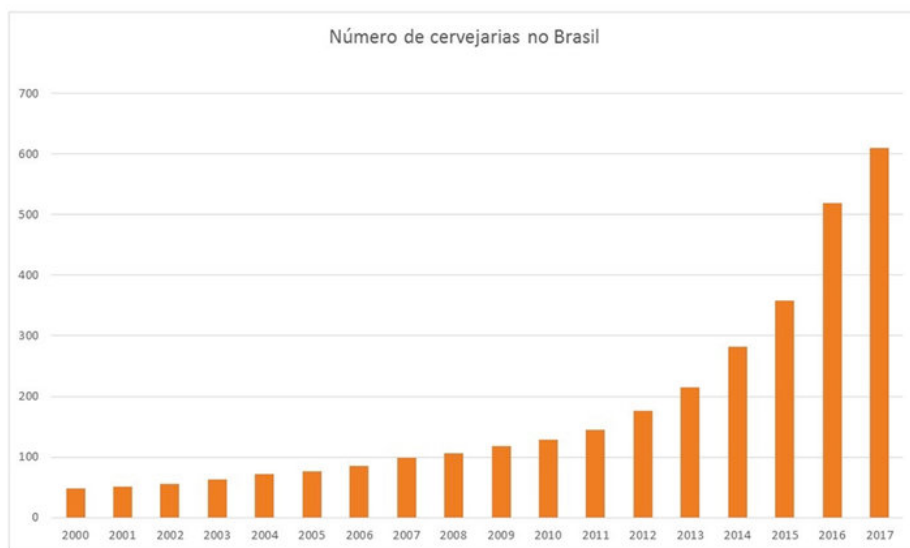


Figura 1 – Número de cervejarias no Brasil por ano

Apesar do crescimento acentuado, muito desse crescimento é realizado de maneira apressada e desordenada, tendo em vista o desconhecimento de empreendedores que buscam um investimento de alto retorno financeiro (quase 2% ao ano) e taxa de crescimento altíssima. Isso acarreta que grande parte das microcervejarias são montadas a partir de conhecimento nulo, ou quase nulo, dos processos, sejam eles, relacionados a venda, logística interna, logística externa, contato com fornecedores, ou até mesmo da produção, como é o caso da cervejaria Schörstein, de Pomerode, onde os sócios proprietários não possuíam conhecimento algum sobre a produção de cerveja. Esses dados foram obtidos durante o curso "Como Montar sua Cervejaria"<sup>1</sup>, realizado na Escola Superior de Cerveja e Malte em Blumenau.

Essa falta de preparo dos empreendedores desse ramo se reflete também no uso de tecnologias e técnicas básicas para garantir o sucesso do negócio. A partir do conhecimento adquirido no curso em Blumenau, foi realizado um levantamento junto as microcervejarias da região de Florianópolis, como Kairós, Liffey BrewPub, Cozalinda, Cervejaria da Ilha, Sambaqui, entre outras, para ver o que elas possuíam de maior carência. Com esse levantamento foram descobertos vários problemas, como por exemplo, que a grande maioria delas não possui nenhum sistema ou método de controle de estoque, ou apenas possui um controle realizado em anotações em folhas de papel, ou ainda que a maioria delas não possui nenhum controle sobre onde estão seus barris a partir do momento que eles saem da fábrica. O problema do não uso de tecnologias em um controle de estoque é que um controle dos itens em estoque realizado por anotações em papel, por exemplo, resulta em atrasos na atualização dos dados, seja por falta de cultura dos funcionários em anotar cada entrada ou saída de itens do estoque, ou por perda da folha de papel em que os dados estão anotados, fazendo com que os dados de itens em estoque quase nunca reflitam o estoque real atual. Outro problema resultante desse atraso na atualização de dados ou da inexistência de dados atuais de estoque está relacionado ao fato de que sem esses dados coletados corretamente não é possível realizar uma análise dos mesmos para um melhor planejamento da produção e da fábrica, ou melhoria dos processos da mesma.

Por esse motivo, optou-se por realizar um projeto com base em atender essa necessidade de coleta de dados para análise e planejamento do estoque, com o desenvolvimento de um sistema que possa ser comercializado para várias microcervejarias que possuem essa carência ao mesmo tempo.

### 1.1.2 A empresa

O projeto foi desenvolvido de maneira autônoma, por mim, porém com o auxílio e a parceria, fornecendo equipamentos, dados necessários para o desenvolvimento, informações

<sup>1</sup> Curso realizado no período de 18/01/2016 à 29/01/2016, com carga horária de 80 horas. Maiores informações sobre o curso podem ser encontradas no link: <http://www.cervejaemalte.com.br/curso/como-montar-sua-cervejaria>

sobre necessidades e capacidades financeiras das microcervejarias em geral, e ambiente para testes do projeto uma vez pronto, da Cervejaria Kairós.

A Cervejaria Kairós é uma microcervejaria, localizada na ilha de Florianópolis com fundação no final do ano de 2016. A meta inicial deles é de se estabelecer como sendo uma cerveja da região, fixando-se, portanto, no mercado consumidor de cervejas artesanais de Florianópolis e região.

Foi montada com um investimento considerado alto para os padrões de microcervejarias, mais de 6 milhões de reais, divididos entre capital dos sócios fundadores e de investidores externos. Por esse motivo é uma cervejaria que, apesar de nova, costuma investir e fomentar novas ideias que possam ser agregadas aos processos deles, ou que possam alterá-los visando maior produtividade e diminuição de gastos. Como exemplo mais recente disso, podemos citar o desenvolvimento e instalação de protótipo de uma estação de tratamento de esgoto por eletrólise, tecnologia inovadora para o Brasil.



Figura 2 – Logotipo Cervejaria Kairós

## 1.2 Objetivos do trabalho

No presente item são apresentados os objetivos, gerais e específicos, do trabalho, que norteiam o desenvolvimento do projeto.

### 1.2.1 Objetivo geral

Desenvolver um sistema web de apoio à gestão de produção com a coleta dos dados de estoque, primário e final, e ativos retornáveis.

Em termos de estratégia de desenvolvimento e em função do modelo de negócios preliminarmente definido, utilizou-se de uma metodologia de software orientado a serviços, desenvolvendo-o assim, de forma mais genérica e customizável possível, para mais de uma cervejaria, e tornado possível que a cervejaria-cliente escolha os serviços mais adequados à sua necessidade.

Resulta disso um sistema que coleta os dados sobre estoques e ativos retornáveis e os mostra em tempo real, para posterior análise de indicadores de desempenho, uma vez que o ponto de partida para uma empresa obter melhorias no desempenho de seus processos é a gestão por meio de indicadores de desempenho, pois, o que não é medido, não é gerenciado; o que não é gerenciado, não é passível de melhorias, e se não há um constante processo de melhorias, em breve deixará de existir [2].

### 1.2.2 Objetivos específicos

Os objetivos específicos do projeto são:

- Entender as necessidades do cliente;
- Especificar os serviços a serem incluídos no sistema;
- Selecionar e estudar a linguagem de programação escolhida;
- Modelar o banco de dados;
- Criar um dashboard para visualização dos dados obtidos através dos serviços e análise dos mesmos;
- Desenvolver o sistema para aquisição de dados sobre os estoques primário, final e ativos retornáveis;

## 1.3 Aderência ao curso de Engenharia de Controle e Automação

Dentro do escopo do curso de Engenharia de Controle e Automação, verifica-se relação direta do projeto desenvolvido com as matérias de Introdução a Engenharia de Controle e Automação, onde os alunos são apresentados a um problema para buscar solução do mesmo, e Metodologia de Desenvolvimento de Sistemas, visto que no sistema em questão verifica-se a necessidade do engenheiro entender o problema, definir a metodologia mais adequada para a solução do problema incluindo planejamento, execução e análise dos resultados obtidos com o projeto visando melhorias constantes no mesmo.

Ainda podemos ver a relação com a disciplina de Avaliação de Desempenho de Sistemas de Automação Discreta, pois o trabalho todo se baseia em colher dados e disponibilizá-los para que as microcervejarias possam utilizá-los para avaliar o desempenho da fábrica e de seus processos de acordo com os indicadores de desempenho definidos por cada cervejaria. Seguindo nessa mesma linha, podemos ver relação entre o projeto e a disciplina de Integração de Sistemas Corporativos, uma vez que o entendimento sobre ERP,

Enterprise Resource Planning, principalmente, auxilia no desenvolvimento do sistema para coletar dados de diferentes formas e também para juntar diferentes tecnologias. Além disso, ainda temos conceitos sobre o que são softwares orientados a serviços e o que são serviços que vem da disciplina de Engenharia de Software Orientada a Serviços

Por fim, podemos ver também uma forte ligação do projeto com as matérias que fazem parte do escopo de informática do curso, como Fundamentos de Sistemas de Bancos de Dados, uma vez que é necessário o entendimento, por parte do engenheiro, em como realizar a correta modelagem do banco de dados que será utilizado no sistema, Introdução à informática para Controle e Automação, Fundamentos da Estrutura da Informação, entre outras, uma vez que foi necessário o entendimento sobre programação e o uso de boas práticas de programação para o desenvolvimento do projeto.

## 1.4 Estrutura do documento

Este documento é dividido em 5 capítulos. O capítulo 2 apresenta a fundamentação teórica e os softwares utilizados, juntamente com a descrição conceitual do projeto. A especificação é apresentada no capítulo 3, juntamente, com a solução encontrada. No capítulo 4 temos o desenvolvimento e implantação do projeto, bem como os resultados do projeto e, finalmente, no capítulo 5 temos as conclusões e perspectivas futuras.



## 2 Fundamentos Teóricos e Softwares Utilizados

Este capítulo tem por objetivo fornecer embasamento teórico para um melhor entendimento do que foi realizado nesse projeto. Serão explicados aqui conceitos utilizados sobre Avaliação de Desenvolvimento, Software como Serviço, Metodologia de Desenvolvimento, Modelagem de Banco de Dados, Código de barras, Heroku e Ruby on Rails.

### 2.1 Avaliação de desempenho

O crescimento exponencial nos últimos anos de microcervejarias faz com que, cada vez mais, seja necessário que essas microcervejarias tenham diferencial em seus processos e em suas fábricas buscando o melhor desempenho possível para não ficarem atrás de outras microcervejarias, devido a concorrência entre as mesmas, e até mesmo com as grandes cervejarias já estabelecidas no mercado brasileiro. Sendo assim, a necessidade de uma avaliação de desempenho através de um acesso rápido as informações da fábrica para posterior melhorias da mesma torna-se indispensável.

Adaptado de informação dada durante a disciplina de Avaliação de Desempenho de Sistemas de Automação Discreta, pelo professor Ricardo José Rabelo, Avaliação de Desempenho é uma atividade que se aplica a um sistema qualquer para se analisar o seu estado, atual ou futuro, frente a um conjunto de objetivos definidos pelos avaliadores através de dados desse sistema para que se possa diagnosticar atividades de melhoria do sistema.

Para que se possa realizar essa avaliação de seus processos, as microcervejarias necessitam coletar os dados de forma mais dinâmica possível, isto é, coletar esses dados em tempo real para não perder tempo e conseguir adaptar seus processos de forma a melhorá-los. A partir da posse desses dados, os gestores podem analisá-los de acordo com os indicadores de desempenho definidos pela cervejaria, e, sendo assim, traçar ações para evitar desperdícios e tornar seus processos mais otimizados.

#### 2.1.1 Indicadores de desempenho

A definição dos indicadores de desempenho é considerada uma das partes, ou a parte, mais importante de todo o processo de avaliação de desempenho, pois, afinal, se a empresa possuir indicadores de desempenho mal formulados, a sua avaliação será feita de forma errada e com isso os dados coletados podem se tornar inúteis e as ações traçadas pós

avaliação visando melhorias podem ser tomadas de maneira errônea, inclusive, as vezes, prejudicando mais do que auxiliando na melhoria dos processos. Segundo o SEBRAE, grande parte das empresas que morrem durante os primeiros dois anos de vida, sofre do mal da escolha incorreta de indicadores.

Segundo os artigos "5 indicadores de desempenho para medir seu sucesso" [3], para definir bons indicadores de desempenho a empresa necessita, primeiramente, entender o planejamento estratégico da empresa e ter objetivos claros na hora da definição das metas que se deseja alcançar. A partir disso, a definição dos indicadores deve ser feita para monitorar os resultados que se quer atingir. Para isso é necessária uma correta base de dados coletados, como podemos ver no artigo "base de dados é a base de tudo" [4].

## 2.2 Software como serviço

Em virtude da necessidade de ter um sistema para auxiliar com a aquisição de dados, advindos dos estoques e dos bens retornáveis, e análise dos mesmos, ser comum entre as cervejarias da região de Florianópolis, fato este descoberto através de pesquisa com os donos de microcervejarias realizadas por mim, faz-se necessário que o desenvolvimento do sistema seja feito através do uso de técnicas voltadas a software orientado a serviços.

Segundo informação repassada durante a disciplina de Integração de Sistemas Corporativos, pelo professor Ricardo José Rabelo, SaaS, Software as a Service, é a disponibilização de serviços, com uma ou mais funcionalidades, através de internet (baseado em computação na nuvem), em que cada serviço é acessado de maneira independente pelo cliente de acordo com as suas necessidades. Essa disponibilização de serviços, na maioria dos casos é realizada sem nenhuma instalação de software em computadores do cliente, sendo apenas necessária uma conexão com internet e um browser para fazer uso do serviço.

O acesso ao serviço sem a necessidade de instalações em computadores do cliente dão margem para outras duas práticas comuns nesse tipo de software:

- O uso do serviço por meio de cobrança de assinatura, em que o cliente paga uma taxa, mensal, anual ou de acordo com a quantidade de usos, e utiliza o serviço pelo qual pagou;
- O uso de arquitetura multi-tenant, ou multi-inquilino, o qual se refere a um software que esteja disponível na nuvem e que seja acessado por vários usuários ao mesmo tempo, com diferentes graus de acesso a informações do sistema e do serviço utilizado.

A necessidade de que o sistema seja desenvolvido dessa forma advém de algumas características a serem explicitadas a seguir:



- Escalabilidade, uma vez que sistemas desenvolvidos utilizando essa metodologia possuem alto grau de expansão, tanto em questão de um único serviço que pode ganhar novas funcionalidades sem que essas atrapalhem o funcionamento das já existentes, quanto em questão de uma plataforma que pode ganhar novos serviços a serem oferecidos aos clientes;
- Preço de custo relativamente baixo, uma vez que o sistema todo a ser desenvolvido é pensado para ser comercializado entre microcervejarias, ou seja, pequenas empresas, com pouca disponibilidade financeira para gastos com softwares proprietários caros;
- Independência dos serviços, uma vez que sistemas desenvolvidos dessa maneira, apesar de, as vezes, possuírem uma única plataforma que disponibilize os serviços contratados ao cliente, possuem um grau de independência elevado entre seus serviços fazendo com que diferentes serviços possam funcionar e não obrigando o cliente a contratar serviços que ele julga desnecessários para ele;
- Reutilização do mesmo código fonte para vários clientes, ou seja, apenas um serviço desenvolvido pode ser usado por vários clientes;

## 2.3 Metodologia de desenvolvimento de software

Segundo o artigo "Metodologia de Software" [5], Metodologia de desenvolvimento de software é a maneira de se utilizar um conjunto coerente e coordenado de métodos para atingir um produto de software. Em suma, a metodologia tem o papel de definir quem faz o que, quando, como e onde, para todos os envolvidos no desenvolvimento do software em questão.

### 2.3.1 Metodologia cascata

O modelo em cascata é um modelo de desenvolvimento de software sequencial no qual o projeto tem um constante fluir para frente, como uma cascata. O software desenvolvido através dessa metodologia passa pelas fases de análise de requisitos, projeto, implementação, testes, integração e manutenção de software, como visto na figura 3.

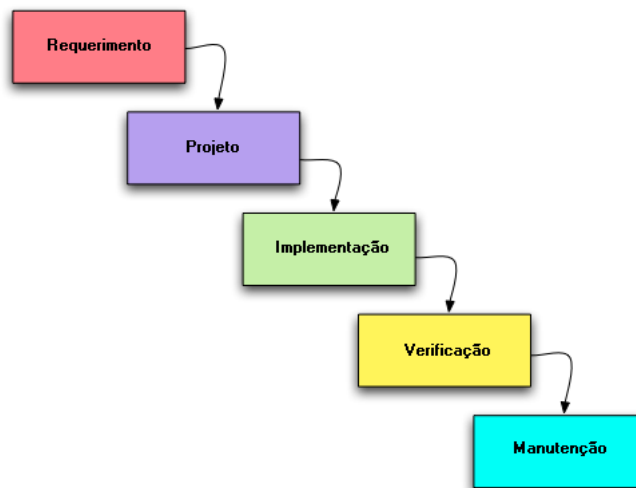


Figura 3 – Exemplo de Metodologia em Cascata

Apesar do modelo em cascata possuir uma estrutura para o desenvolvimento de software bem moldada, ele possui uma grande desvantagem, que é o atraso na entrega do software desenvolvido, uma vez que, quando usada essa metodologia, o desenvolvedor só passará para a fase seguinte quando terminar a fase em que está, ou seja, caso ele esteja na fase de implementação, ele só passará para a fase de testes após terminar de implementar todas as funcionalidades, o que acarretará no atraso já mencionado caso o software que esteja sendo desenvolvido seja muito grande.

### 2.3.2 Metodologia iterativa incremental

De acordo com o artigo "Iterative and incremental developments. A brief history" [6], o modelo iterativo incremental divide o desenvolvimento do projeto em etapas, de modo que podemos ver as etapas existentes no modelo em cascata, análise de requisitos, projeto, implementação, testes, integração e manutenção, em todas essas etapas também.

Cada uma dessas etapas possui seus próprios requisitos, uma vez que são consideradas apenas uma parte do sistema todo a ser desenvolvido, resultando ao final da etapa uma versão do sistema com algum incremento, como pode ser visto na figura 4.

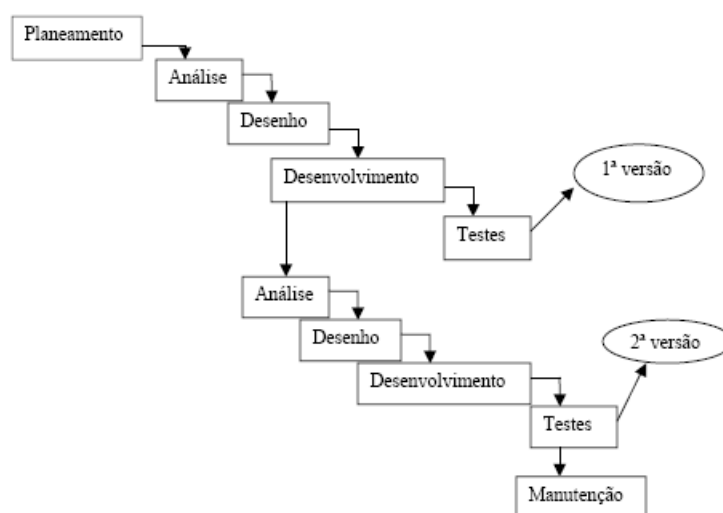


Figura 4 – Exemplo de Metodologia Iterativa Incremental

Em resumo, simplificadamente, podemos dizer que a metodologia iterativa incremental é composta por uma divisão do projeto em partes menores, etapas, e que cada uma dessas etapas possui uma metodologia em cascata internamente em seu desenvolvimento.

### 2.3.3 Metodologia escolhida

Este projeto foi desenvolvido utilizando a metodologia de desenvolvimento de software iterativa e incremental, justamente, pela mesma fornecer uma agilidade maior na entrega das partes desenvolvidas (serviços e funcionalidades), permitindo assim uma iteração maior com o cliente e um feedback dele em relação a como o sistema estava, contribuindo para alterações para tornar o sistema melhor para o cliente.

## 2.4 Modelagem de Banco de Dados

A modelagem do banco de dados tem como objetivo organizar os dados de forma simples e facilitar o uso dos mesmos pelo programa. Nesse projeto foi utilizado a modelagem de um banco de dados relacional, figura 5, pois a mesma possui uma abordagem mais intuitiva da organização dos dados em forma de relações entre esses dados, enxergadas pelo usuário no formato de tabelas. As tabelas são usadas para reter informações sobre os objetos a serem representados no banco de dados. [7]

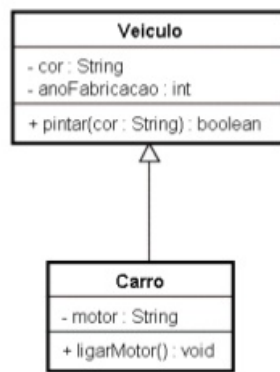


Figura 5 – Exemplo de Banco de Dados Relacional

A arquitetura de um banco de dados relacional tem as seguintes características:

- Tabelas: todos os dados são armazenados em tabelas que possuem linhas e colunas. Essas tabelas associam-se entre si por meio de relacionamentos, que consistem em associar um ou mais atributos da tabela com outro um ou vários atributos de outras tabelas;
- Registros: Cada linha com a sua lista de colunas consiste um registro;
- Atributos: Cada coluna de uma tabela de dados é chamada de atributo;
- Chave: As associações entre as tabelas são realizadas através de chaves, sejam elas estrangeiras ou primárias. As chaves primárias nunca se repetem pois são o identificador de cada registro da tabela. As chaves estrangeiras, por sua vez, são formadas através do relacionamento entre um registros de tabelas diferentes, portanto, é a relação formada entre uma chave primária de um registro com a chave estrangeira de outro registro.

A opção por um banco de dados do tipo relacional foi feita pela possibilidade de serem usados diversos métodos para o tratamento das informações contidas nas tabelas, facilitando assim o uso durante o projeto. Para usar um banco de dados desse tipo, é necessário, também, conhecimento de SQL, que nada mais é que a linguagem estruturada utilizada para realizar as pesquisas nos bancos de dados relacionais.

## 2.5 Código de barras

Segundo a GS1 Brasil, Associação Brasileira de Automação, o código de barras é usado para representar uma numeração (identificação) atribuída a produtos, unidades logísticas, localizações, ativos fixos e retornáveis, documentos, contêineres, cargas e serviços

facilitando a captura de dados através de leitores (scanners) e coletores de código de barras, propiciando a automação de processos trazendo eficiência, maior controle e confiabilidade para a empresa. [8]

O princípio de funcionamento do código de barras é o código binário, e as listras representam zeros e uns. Porém, é o padrão atribuído à sequência que faz com que os leitores consigam descobrir qual produto estão lendo. O código é dividido em 15 seções, como podemos ver na figura, sendo que a primeira sempre é um número ímpar e a última sempre é um número par, para que o leitor consiga identificar se o código de barras está sendo lido de cabeça para baixo ou não. A seção do meio serve para dividir em dois o restante do código (12 seções restantes), metade para cada lado, pois são essas duas partes que contêm as informações sobre os produtos.

A quantidade de números presentes e o que cada um representa dependem do padrão usado. No Brasil os padrões mais usados e que são aceitos, normalmente, por qualquer sistema leitor de código de barras são os padrões UPC-A e EAN-13. Um exemplo de código de barras EAN-13 é dado na figura 6. Além disso os dados sobre cada produto registrado sobre esses padrões ficam armazenados no CNP, Cadastro Nacional de Produtos, para serem consultados pelos sistemas leitores de código de barras.

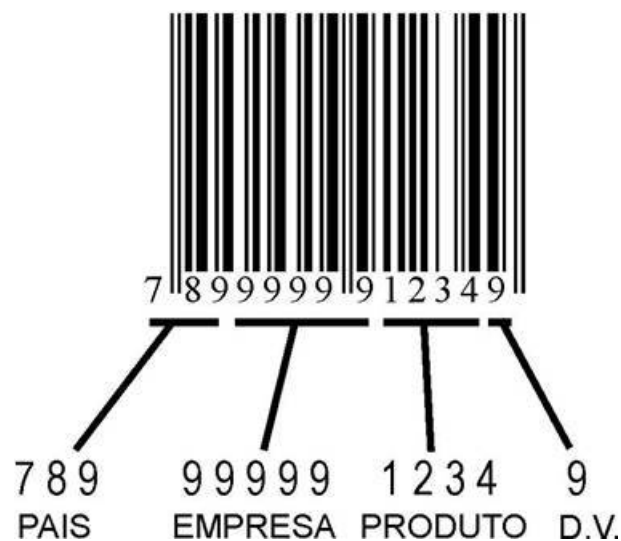


Figura 6 – Exemplo de código de barras

## 2.6 MVC

MVC, ou "Model, View, Controller", ou ainda, em português Modelo, Visão, Controlador, é um padrão de arquitetura de software que separa a representação da informação, da interação do usuário com a mesma, como podemos ver pela figura 7.

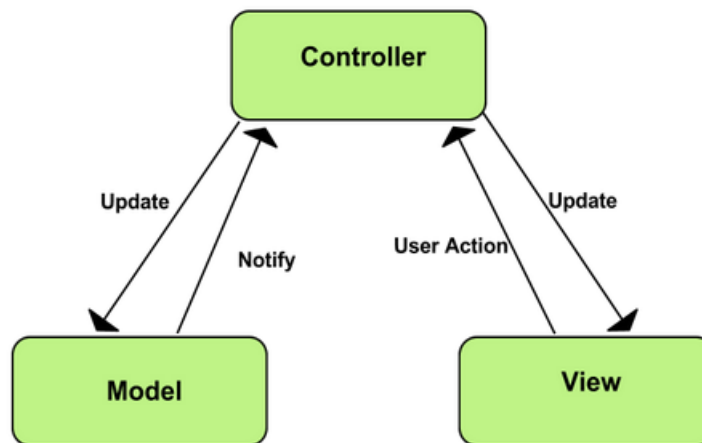


Figura 7 – Diagrama MVC

O Modelo (Model) consiste nos dados da aplicação, fazendo ligação direta com o Banco de Dados. Já uma Visão (View) pode ser qualquer representação dos dados, como por exemplo, uma tabela ou um gráfico. O Controlador (Controller) por sua vez faz o papel de intermediário entre as entradas, convertendo-as em comandos para o Modelo, e as saídas enviando-as para a Visão.

O funcionamento do MVC, para o caso específico do framework Ruby on rails, como podemos ver pela figura 8, o usuário tem contato apenas com a Visão, as entradas do usuário no sistema são enviadas ao Controlador que as traduz em comandos para o Modelo. O Modelo, por sua vez, se conecta com o banco de dados realizando a consulta ou alterações necessárias no banco de dados e devolve a informação a ser mostrada ao usuário final ao Controlador, que na sequência repassa para a Visão.

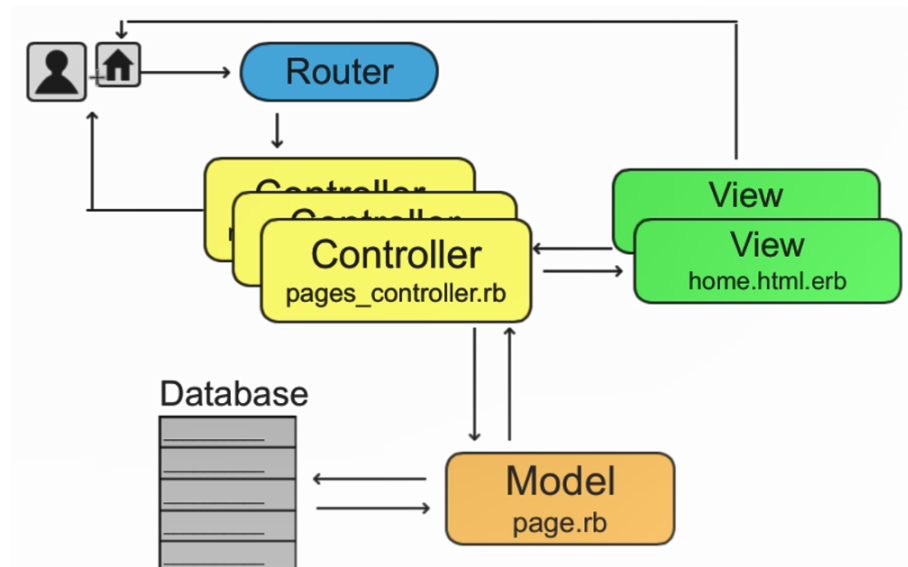


Figura 8 – Diagrama MVC para o caso de um sistema com Rails

No caso específico de um sistema em Rails ainda possuímos mais um elemento na figura 8, o Router. Podemos dizer que ele é responsável por informar ao Controlador quais as informações ou página deve ser mostrada ao usuário na Visão no momento.





## 3 Especificação e Projeto

Este capítulo visa explicitar como o projeto foi desenvolvido e a solução escolhida. Para o desenvolvimento serão abordados na prática os conceitos explicados, anteriormente, no capítulo 3. Então temos como atividades desenvolvidas, o levantamento de indicadores para microcervejarias, a definição de requisitos do sistema e a aquisição de equipamentos, a modelagem de cada serviço do sistema, o estudo da linguagem e das tecnologias usadas no desenvolvimento, a modelagem do banco de dados, a escolha e modelagem de um Dashboard, e por fim, o desenvolvimento do código programado.

É importante lembrar também, que pela metodologia de desenvolvimento usada no projeto ser iterativa incremental, temos um grande feedback por parte do cliente resultando em melhorias constantes e em acréscimos ao sistema, e, sendo assim, algumas etapas descritas aqui ocorrem simultaneamente com outras.

### 3.1 Definição dos serviços que seriam desenvolvidos

A definição de quais serviços seriam desenvolvidos foi realizada através de pesquisa, já mencionada no Capítulo 1, na qual soube-se que para as microcervejarias da região de Florianópolis, três aspectos são mais importantes, atualmente, perante os problemas enfrentados por elas, também mencionados no Capítulo 1, e tamanho delas, maioria estabelecida como micro empresa. São eles:

- Alguma forma de controlar o que há no estoque de matérias primas, também chamado de estoque primário, e atualizá-lo em tempo real, podendo visualizar essas informações a qualquer hora, até mesmo de fora da fábrica;
- Alguma forma de controlar o que há no estoque final, ou de produtos finalizados, e atualizá-lo em tempo real, podendo visualizar as informações dele a qualquer hora, inclusive de fora da fábrica, tal qual as informações de estoque primário;
- Alguma forma de visualizar o estado dos barris, quantos estão cheios e quanto vazios, por exemplo, e sua localização, dentro e fora da fábrica. Do mesmo modo que os itens anteriores, também com visualização a qualquer hora, inclusive fora da fábrica;

Em virtude dessas necessidades deles, e por descobrir que uma cervejaria maior gostaria de possuir esses três itens, enquanto outra menor gostaria apenas um ou dois deles por conta de dificuldades financeiras ou outros motivos, optou-se por desenvolver o sistema de forma que cada um dos itens expostos acima fosse englobado por um serviço separado e independente, resultando assim em três serviços:

- Serviço 1: Controle de estoque primário
- Serviço 2: Controle de estoque final
- Serviço 3: Ciclo de vida de bens retornáveis

A definição das funcionalidades de cada serviços serão expostas mais a frente neste documento.

## 3.2 Levantamento de indicadores

A primeira atitude a ser tomada para que o projeto pudesse ser desenvolvido de maneira a conter informações relevantes para as microcervejarias, foi o levantamento de indicadores, que, de maneira resumida, são as informações que vão aparecer para o cliente final do sistema na tela do seu browser. Como dito, anteriormente, nesse mesmo documento, foi realizada uma pesquisa sobre as necessidades das microcervejarias da região de Florianópolis, e dessa pesquisa, resultou a parceria com apoio para o desenvolvimento do projeto, em questão de equipamentos necessários, ambiente para testes e feedbacks sobre o sistema, com a cervejaria Kairós.

Para o levantamento dos indicadores, tendo em vista os problemas que precisariam ser solucionados pelo sistema e uma ideia do software que eles precisavam, sabidos desde a pesquisa anterior, algo que coletasse os dados da maneira mais rápida possível, os armazenasse em tempo real em ambiente web, para que os dados pudessem ser acessados de qualquer lugar via browser, e que também fizesse o controle do ciclo de vida de ativos retornáveis indicando onde eles estão atualmente, foi usada técnica semelhante, uma pesquisa do que interessava aos clientes enxergarem no sistema quando estivessem utilizando ele. Dessa forma, chegou-se aos seguintes indicadores que deveriam ser mostrados no sistema:

- Quantidade de cada item presente no estoque primário, para saber a quantidade de estoque atual;
- Quantidade de cada item presente no estoque final, para saber a quantidade de estoque atual;
- As receitas que podem ser produzidas, e que não podem ser produzidas, a partir dos níveis dos itens no estoque primário atual;
- A utilização de fornecedores, isto é, a quantidade de usos de cada fornecedor no total, desde a implantação do sistema, para saber em quais fornecedores pode-se tentar chegar a algum tipo de acordo para realização de compras maiores ao invés de várias compras pequenas;

- A variação do estoque primário, por mês, para cada item presente nele, com quantidades total ao final do mês, quantidade de entradas no estoque do item e quantidades de saída do item, ambas por mês também. Isto para se ter conhecimento do quanto se está usando aquele item em questão e de quantas vezes ele é comprado por mês;
- Variação da quantidade de cada produto, por lote, isto é, a quanto tempo cada produto de cada lote está parado no estoque final, junto com o máximo de tempo que aquele produto já permaneceu no estoque para efeitos comparativos com o valor atual;
- Variação por mês de cada produto no estoque final, independente de lote, com quantidade total do produto presente no estoque final ao final do mês e quantidades de entrada e saída do mesmo por mês, para que se tenha conhecimento do quanto se vende e do quanto se produz aquele produto;
- Quantidade produzida de cada produto, independente de lote, por mês;
- Localização dos bens retornáveis, inclusive caso estejam fora da fábrica. Aqui é bom deixar claro que em relação a cilindros de gás e chopeiras, basta saber se estão na fábrica ou fora dela, enquanto que os barris é necessário saber também, caso estejam na fábrica, em que local eles estão, na sala de limpeza, na câmara fria ou no estoque final;
- Estado dos bens retornáveis. Para esse caso, também há diferença entre cilindros de gás e chopeiras em relação aos barris. Enquanto os primeiros necessita-se, apenas, saber se estão sujos ou limpos e prontos para uso, os últimos necessita-se que seja conhecido se estão sujos e vazios, limpos e vazios ou ainda se estão cheios e de que tipo de cerveja estão cheios;
- Quais barris, que estão fora da fábrica, precisam ser recolhidos, uma vez que a validade desse produto é de até 7 dias e ao final desses 7 eles devem retornar à fábrica, mesmo se ainda não tiverem terminado;
- Quantidade de barris, de cada capacidade, estão em que local, seja na fábrica ou fora dela;
- Quantidade total, desde a implantação do sistema, de uso de cada barril, para saber quais barris estão no seu limite de uso de acordo com o fabricante;

### 3.3 Requisitos Funcionais e Não-funcionais

Era necessário ainda realizar o levantamento dos requisitos funcionais e não-funcionais antes de definir, efetivamente, como seria o sistema a ser desenvolvido.

Para os requisitos funcionais, foram usados como base os indicadores de desempenho levantados junto à cervejaria, apresentados na seção 3.2, passando estes a serem requisitos funcionais também. Já os requisitos não-funcionais do sistema baseiam-se mais nos objetivos das cervejarias com um sistema desse. Dessa maneira podemos classificá-los em 5 áreas:

### 3.3.1 Usabilidade

Requisitos de usabilidade estão presentes em qualquer projeto que possua interação com o usuário, pois são os responsáveis por especificar a satisfação do usuário no uso do sistema. Nessa categoria de requisitos podemos enunciar os seguintes:

- O sistema deve ser intuitivo de maneira a facilitar o aprendizado do usuário;
- O sistema deve ser fácil de usar de maneira a diminuir erros do sistema na interação do usuário com o mesmo;

### 3.3.2 Confiabilidade

Confiabilidade de software representa a probabilidade de o sistema causar falhas. Dessa forma possuímos os seguintes requisitos:

- O software deve estar disponível sempre que o cliente desejar utilizá-lo, independente de hora e localização desde que o cliente tenha acesso à internet;
- O sistema deve ter uma taxa de falhas igual ou inferior à 2/1000, ou seja, 0,2%;

### 3.3.3 Desempenho

Desempenho representa como o sistema deve responder às solicitações do cliente. Nesta categoria temos definidos os seguintes requisitos:

- O tempo de resposta do sistema deve ser o mais rápido possível, respeitando as condições da internet local, para que atualizações de estoque, por exemplo, possam ser realizadas no momento que forem necessárias pelo usuário;
- O sistema deve ser capaz de conseguir atender múltiplos usuários de diferentes cervejarias ao mesmo tempo;
- O sistema não deve ocupar memória computacional do usuário;

### 3.3.4 Portabilidade

Portabilidade nos diz onde que o sistema deve funcionar. Para esse caso, temos apenas um item nesse quesito, que é:

- O sistema deve funcionar em qualquer aparelho, adaptando-se automaticamente à tablets, celulares e desktops de diferentes tamanhos;

### 3.3.5 Segurança

Segurança define os acessos ao sistema apenas à usuários cadastrados. Também define o acesso a diferentes serviços ou a funcionalidades de acordo com as permissões previamente estabelecidas aos usuários.

- O sistema deve ser acessível apenas a usuários cadastrados e logados;
- Os serviços devem ser acessíveis apenas por usuários cadastrados para uso daquele serviço. O mesmo para as funcionalidades de cada serviço;
- Apenas usuários com privilégios de administrador devem ser capazes de alterar informações sensíveis para a cervejaria, como a receita de cervejas, os serviços utilizados pela mesma ou informações sobre usuários cadastrados pela cervejaria;
- O sistema deve ser capaz de isolar as informações referentes a diferentes cervejarias clientes do sistema;
- informações referentes a senhas devem ser criptografadas para evitar roubo de senhas.

## 3.4 Solução escolhida

Tendo em vista os conceitos abordados no Capítulo 2, os requisitos, funcionais e não-funcionais, apresentados neste capítulo para o sistema, e os objetivos apresentados ainda no Capítulo 1, podemos ver algumas escolhas, listadas abaixo, realizadas em relação às tecnologias utilizadas para o desenvolvimento de um sistema com o intuito de auxiliar os gestores das microcervejarias com a coleta e análise de desempenho da fábrica através de dados de seus estoques e bens retornáveis.

- O sistema deve ser baseado em serviços, SaaS, uma vez que sistemas desse tipo fornecem a possibilidade de escalabilidade com a implementação de novos serviços ao longo do tempo e, também, independência no uso desses serviços por diferentes clientes, respeitando assim tanto requisitos funcionais quanto não-funcionais;

- O sistema deve ser implantado na nuvem, para que assim requisitos como, a utilização do sistema por diversos usuários, e visualização de dados em qualquer lugar e a qualquer hora, sejam respeitados. Para isso é necessário escolher uma linguagem de programação capaz de realizar isso. Essa escolha será justificada mais a frente neste mesmo documento;
- Os serviços devem fazer uso da melhor tecnologia disponível para garantir requisitos não-funcionais de desempenho e também sua automação eliminando ao máximo interação do usuário com o sistema para alteração de dados de estoque por exemplo. Para isso a solução ideal seria fazer uso de tecnologia RFID, que se utilizaria de transmissores e receptores de ondas de rádio para leitura de dados e alteração dos mesmos no sistema, bem como também permite uma maior automação desses processos, porém essa tecnologia ainda é cara demais no país, o que resulta na utilização de leitores de código de barras para realizar esses processos. Por sua vez os leitores de código de barras não permitem que o sistema haja de maneira autônoma, sem a interferência de usuários para atualização de dados;

Dessa forma o sistema a ser desenvolvido deveria ter a arquitetura geral mostrada na figura 9. Nela podemos ver a presença de um servidor que se comunica com um banco de dados, também na nuvem, sempre que necessário realizar uma consulta ou alteração de algum dado. Esse mesmo servidor também comunica-se com uma interface gráfica para visualização do usuário dos dados presentes no sistema e também para que o usuário possa realizar a alteração de dados no sistema. Ainda podemos ver nessa figura que cada serviço tem sua própria interface com o usuário funcionando de maneira independente e, também, de acordo com os serviços escolhidos pelo cliente/usuário.

### 3.5 Aquisição de equipamentos

Um vez conhecido o que deve ser mostrado ao cliente final do sistema, precisou-se definir como seriam as entradas de dados no sistema e então definir os equipamentos que deveriam ser adquiridos para realizar o projeto da maneira que as cervejarias desejam, de acordo com suas limitações.

Foi optado pelo uso de leitores de código de barras para realizar a coleta dos dados sobre os estoques, primário e final, e também sobre os ativos retornáveis, uma vez que leitores de códigos de barras são uma solução muito mais barata, e que se encaixa melhor no orçamento de micro e pequenas empresas, frente a outras tecnologias mais avançadas, como RFID, Identificação por Rádio-Frequência. Porém, como explicitado no Capítulo 3, existem diversos padrões de códigos de barras que são utilizados no nosso país e no mundo, e por esse motivo, acabou-se optando por fazer uso de leitores de código de barras

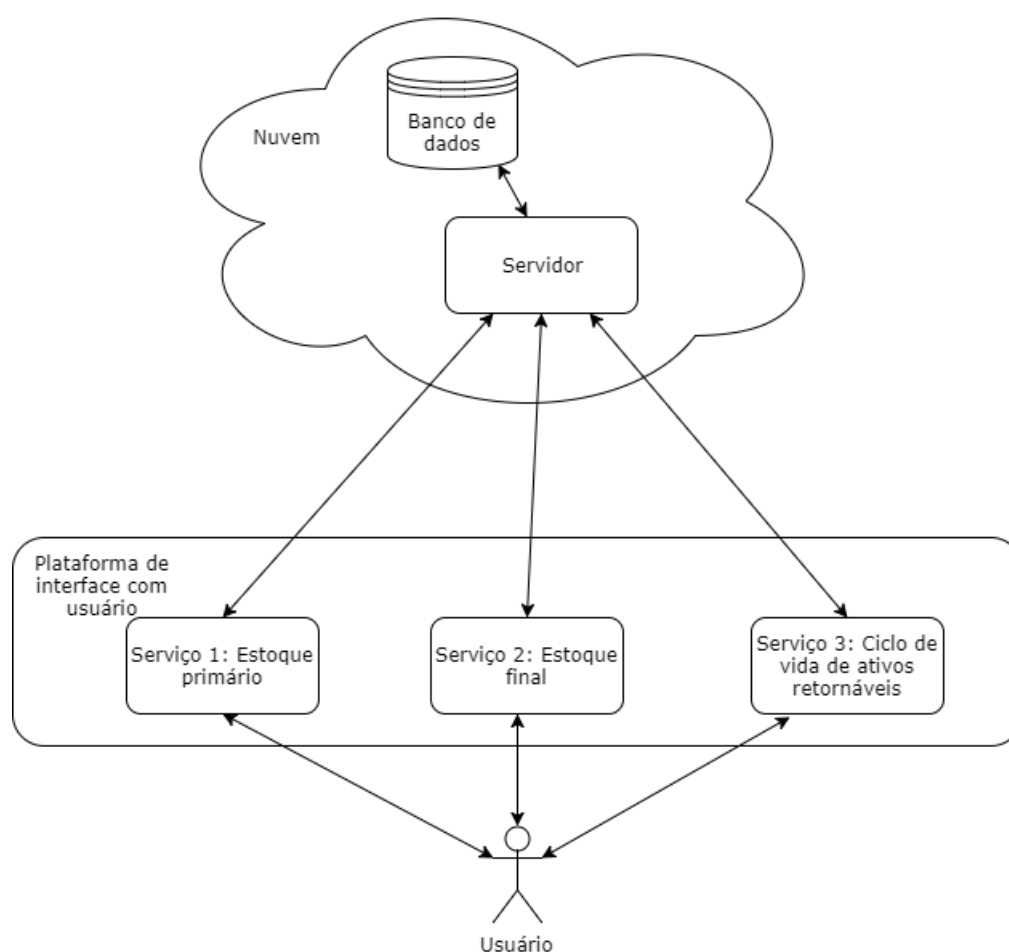


Figura 9 – Arquitetura geral do sistema a ser desenvolvido

apenas no estoque final e no ciclo de vida dos ativos retornáveis, como barris, cilindros de gás e chopeiras, já que no estoque primário há diferentes tipos de padrões uma vez que cada matéria prima vem de um fornecedor diferente com seu padrão podendo ser diferente do que o sistema utiliza.

Após pesquisa entre os diferentes tipos de leitores, acabou-se optando por um leitor com conexão Bluetooth, para permitir mobilidade do leitor, sem base. Esse tipo de leitor sem base, necessita de um computador, de um tablet ou celular para realizar a correta conversão das barras lidas em um código numérico.

Por esse motivo, além do leitor de código de barras, cada leitor comprado pela fábrica necessita também de um tablet. Como tablets não industriais, possuem conexão wifi, que será usada para acessar o sistema e não pode ser ocupada com outra comunicação ao mesmo tempo, e não possuem nenhuma entrada de cabo que seja compatível com os leitores de código de barras existentes, esse também é um motivo pelo qual o leitor de código de barras deve ter a comunicação Bluetooth, para que a conexão entre leitor e tablet seja estabelecida. Dessa forma, a arquitetura de comunicação dessa parte envolvendo

leitores de códigos de barras pode ser representada como na figura 10.

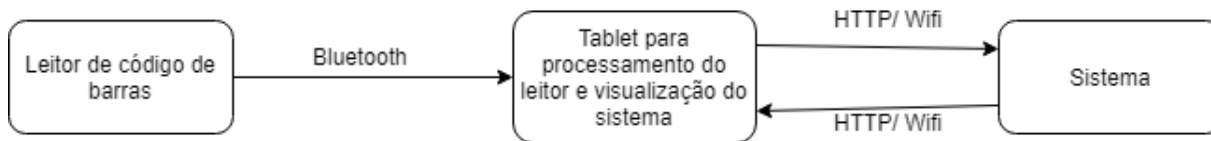


Figura 10 – Esquema da comunicação entre leitor de código de barras, tablet e sistema

Outro fator que leva a escolha de tablets como "computador" para processamento dos dados do leitor de código de barras, é o fato de que com eles há possibilidade do usuário visualizar outras informações do sistema também.

A quantidade de cada equipamento, tablet e leitor de código de barras, para compra que foi definida para cada implementação de serviço que faça uso deles (serviços 2 e 3), foi de 1 leitor e 1 tablet.

Para a entrada de dados no sistema referente ao Serviço 1: Controle de Estoque primário, temos o uso de um formulário diretamente no sistema informando os dados necessários sobre a matéria prima que está entrando ou saindo do estoque. Isto pode ser feito através de celular, computador, ou mesmo, através de um dos tablets adquiridos.

## 3.6 Modelagem dos serviços

A modelagem dos serviços que será explicitada aqui refere-se à divisão das funcionalidades entre os diferentes serviços desenvolvidos, de acordo com os indicadores apresentados no item 4.2. A explicação da modelagem adotada para os serviços será feita aqui de maneira dividida entre cada um dos serviços, apresentando suas funcionalidades e a interação entre usuário e sistema para eles.

### 3.6.1 Serviço 1: Controle de estoque primário

Para o serviço de controle de estoque primário temos as seguintes funcionalidades:

- Visualização dos dados sobre o estoque primário, sobre o uso de fornecedores e, também, sobre cada matéria prima registrada no estoque, conforme informado no item 4.2 deste relatório;
- Cálculo de quais receitas da cervejaria podem ser produzidas com o montante de estoque primário atual e quais não podem ser produzidas, através do registro das receitas no sistema;



- Atualização dos itens em estoque atualmente, tanto para entradas, quanto para saídas do estoque primário.

Podemos ver na figura 11, o diagrama de casos de uso para o serviço 1, desde que o usuário esteja logado no sistema. Através dele pode-se ver as ações que podem ser tomadas pelo usuário, inclusive com diferenciação entre usuários com privilégio comum e usuários com privilégio de administrador. Os usuários com privilégio comum podem realizar as ações de visualizar o estoque atual com os itens que estão armazenados no momento e suas quantidades, visualizar os indicadores relacionados ao serviço 1 previamente citados no item 4.2 deste relatório, atualizar entradas ou saídas do estoque primário, registrar novas matérias primas, e editar matérias primas já cadastradas no sistema. O usuário com privilégios de administrador, por sua vez, possui todas as ações do usuário comum à sua disposição, somadas às ações de visualizar receitas cadastradas, editar receitas cadastradas, deletar receitas cadastradas, cadastrar nova receita e deletar matérias primas cadastradas.

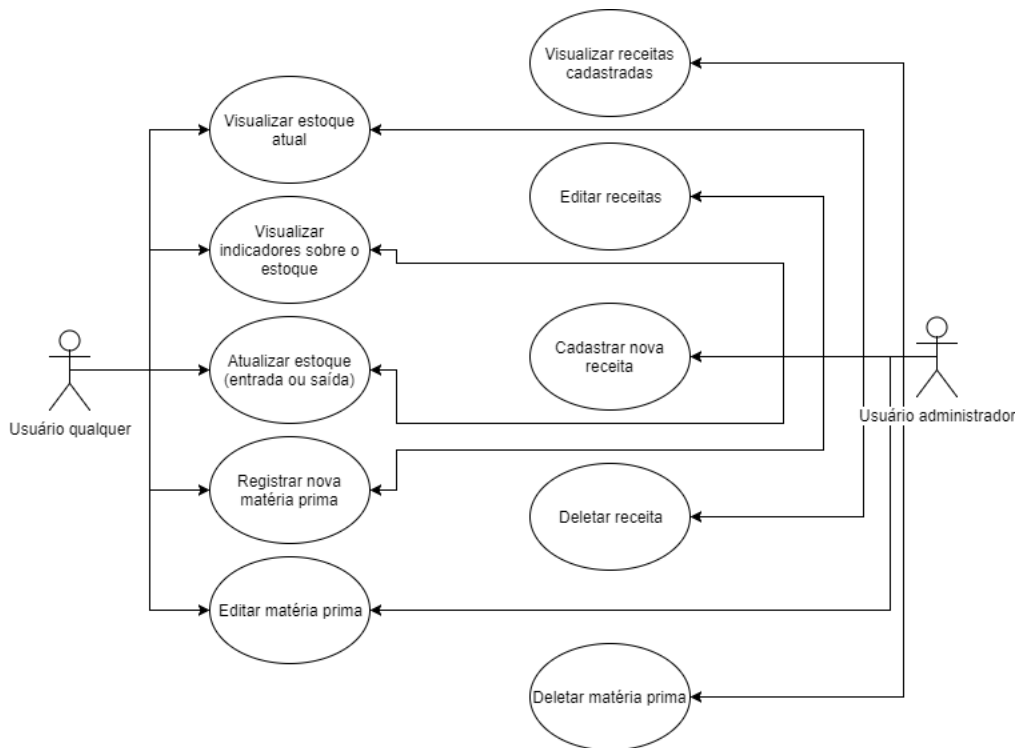


Figura 11 – Diagrama de uso do serviço 1

Através da figura 12, podemos ver um exemplo de uso do serviço 1, para o caso do usuário estar logado, através do diagrama de sequência simplificado das atividades envolvidas no caso de uso do usuário querer visualizar os dados das matérias primas no estoque atualmente. Nesse caso, a sequência de eventos é o usuário faz uma requisição ao sistema pela página de visualização dos dados do estoque primário atual, após isso o sistema realiza a pesquisa dos dados do estoque atual no banco de dados, o banco de dados retorna os dados consultados, e, posteriormente, o sistema retorna a página com

os dados do estoque atual ao usuário. Isso para o caso de existirem dados no banco de dados, ou seja, já houve uma atualização com entrada ou saída de matéria prima do estoque primário, porém, caso não haja dados salvos no banco de dados ainda, podemos verificar a existência de um tratador de erros.

No caso de não haver dados salvos no banco de dados ainda, representando que não houve atualização do estado do estoque primário ainda, o banco de dados retorna nulo para a pesquisa realizada pelo sistema, e o mesmo por sua vez, retorna mensagem ao usuário informando que não há itens no estoque ainda. Após isso, o usuário realiza uma requisição do formulário de atualização do estado do estoque primário ao sistema, que por sua vez, retorna ao usuário a página com o formulário. Após preenchido o formulário, caso não haja nenhum erro no preenchimento dele, o usuário envia a requisição de alteração no banco de dados ao sistema, e o mesmo salva os novos dados no banco de dados, retornando ao usuário que os dados foram salvos com sucesso e a situação do estoque foi atualizado com sucesso.

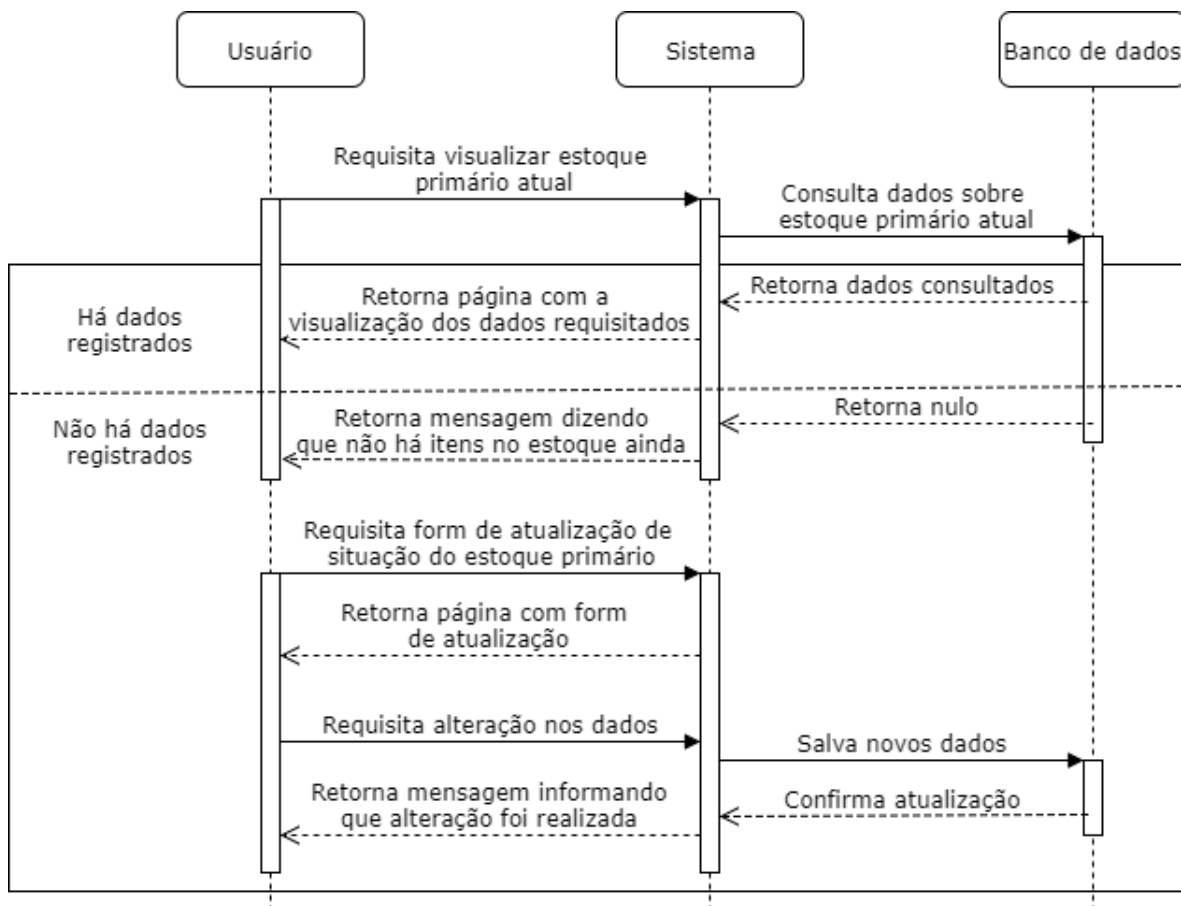


Figura 12 – Exemplo de diagrama de seqüência pra caso de uso do serviço 1

Mais a frente neste mesmo documento, na seção 4.8, poderemos ver mais a fundo o funcionamento dos casos de uso, e do sistema em si, através de pseudo código, bem como as seqüências corretas de uso do sistema, para que o mesmo funcione corretamente

trazendo ao usuário todas as informações desejadas.

### 3.6.2 Serviço 2: Controle de estoque final

Para o serviço de controle de estoque final temos as seguintes funcionalidades:

- Visualização dos dados sobre o estoque final, sobre as quantidades produzidas de cada produto, também, sobre cada produto, por lote, registrado no estoque, conforme informado no item 4.2 deste relatório;
- Atualização dos itens em estoque atualmente, tanto para entradas, quanto para saídas do estoque final.

Podemos ver na figura 13, o diagrama de casos de uso para o serviço 2, desde que o usuário esteja logado no sistema. Através dele pode-se ver as ações que podem ser tomadas pelo usuário, inclusive com diferenciação entre usuários com privilégio comum e usuários com privilégio de administrador. Os usuários com privilégio comum podem realizar as ações de visualizar o estoque atual com os itens que estão armazenados no momento e suas quantidades, visualizar os indicadores relacionados ao serviço de estoque final, previamente citados no item 4.2 deste relatório, atualizar entradas ou saídas do estoque final, registrar novos produtos, e editar produtos já cadastrados no sistema. O usuário com privilégios de administrador, por sua vez, possui todas as ações do usuário comum à sua disposição, juntamente com a ação de deletar produtos cadastrados.

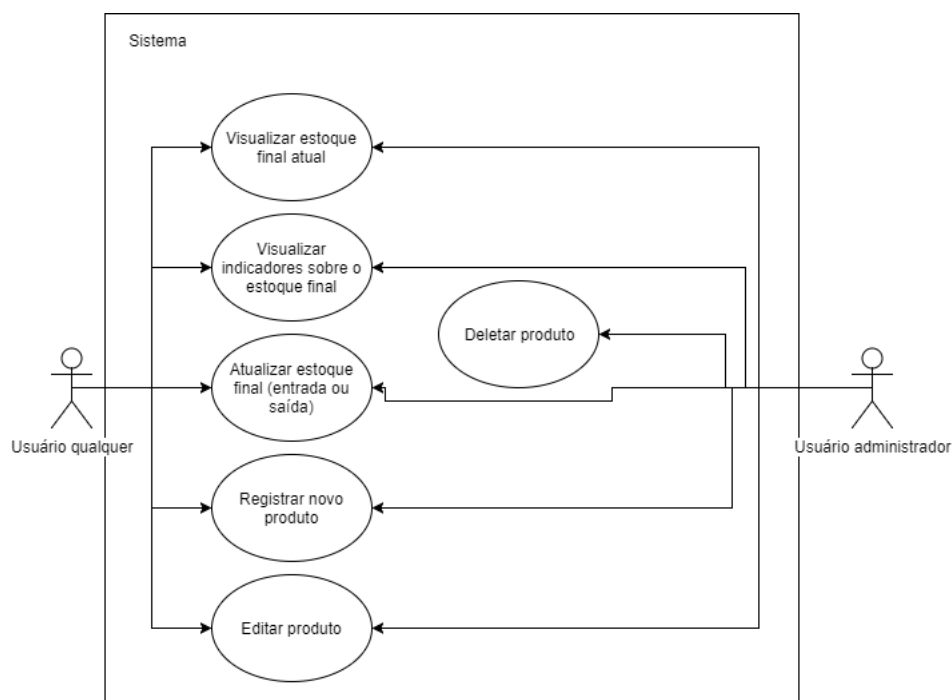


Figura 13 – Diagrama de uso do serviço 2

Podemos ver um exemplo de uso do serviço 2, para o caso do usuário estar logado, através do diagrama de sequência simplificado das atividades envolvidas no caso de uso do usuário querer visualizar os dados dos produtos no estoque atualmente, figura 14. Nesse caso, a sequência de eventos é o usuário faz uma requisição ao sistema pela página de visualização dos dados do estoque final atual, após isso o sistema realiza a pesquisa dos dados do estoque atual no banco de dados, o banco de dados retorna os dados consultados, e, posteriormente, o sistema retorna a página com os dados do estoque atual ao usuário, caso os dados existam no banco de dados.

Caso não haja dados salvos no banco de dados ainda, representando que não houve atualização do estado do estoque primário ainda, podemos verificar a existência de um tratador de erros, em que o banco de dados retorna nulo para a pesquisa realizada pelo sistema, e o mesmo por sua vez, retorna mensagem ao usuário informando que não há itens no estoque ainda. Após isso, o usuário realiza uma requisição do formulário de atualização do estado do estoque final ao sistema, que por sua vez, retorna ao usuário a página com o formulário. O usuário utiliza o leitor de código de barras para informar o código de barras do produto que irá ser atualizado no estoque (entrada ou saída), então, o sistema pesquisa no banco de dados pelo código de barras e retorna as informações sobre o produto para a página de formulário. Após isso o usuário requisita uma alteração daquele produto no estoque, o sistema salva as novas informações sobre o produto no banco de dados e retorna mensagem ao usuário informando que a alteração do estoque foi realizada com sucesso.

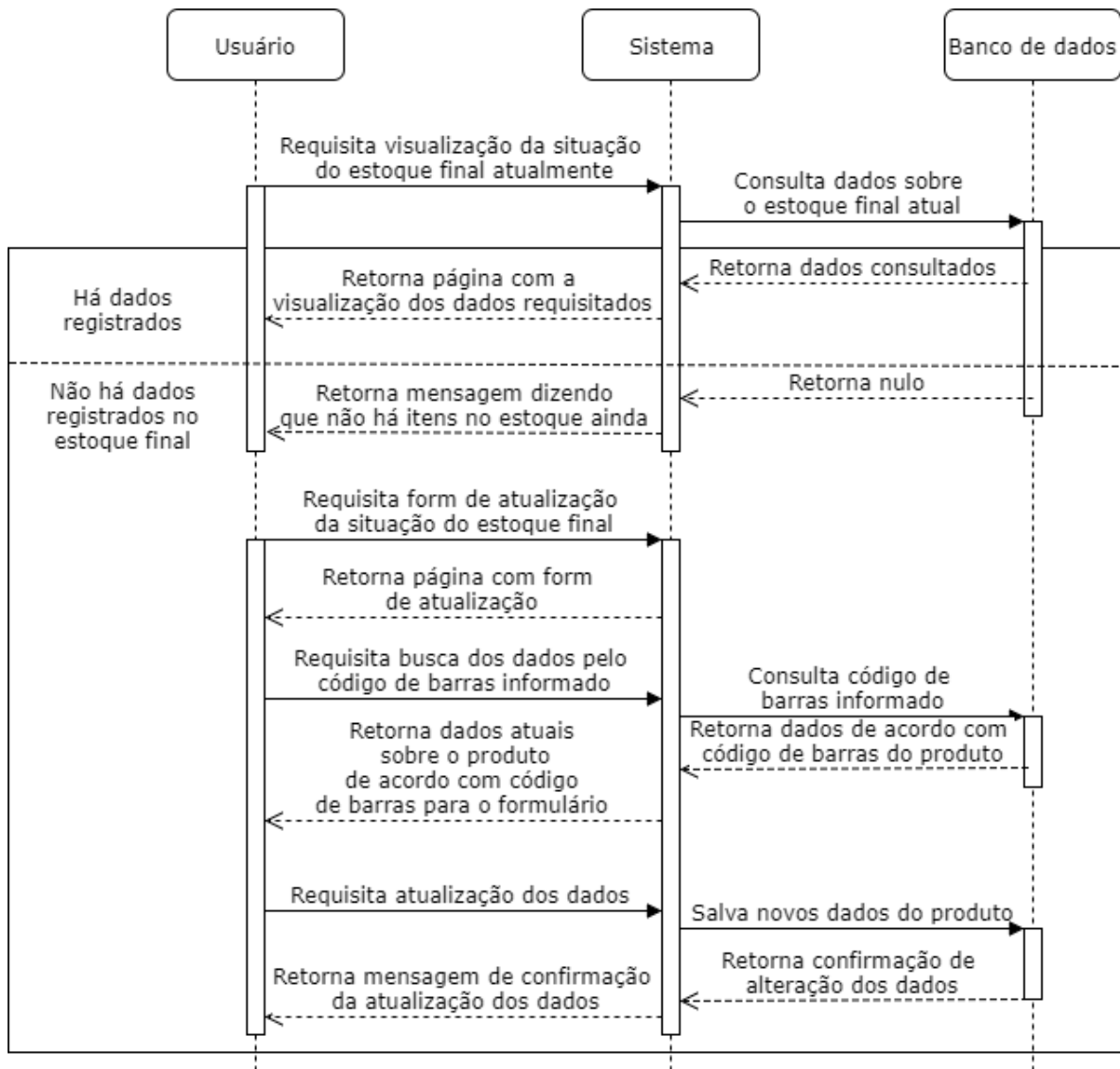


Figura 14 – Exemplo de diagrama de sequência pra caso de uso do serviço 2

Na seção 4.8 poderemos ver mais sobre o funcionamento de cada caso de uso do serviço 2, através de pseudo-código. Também será mostrada a correta sequência de funcionamento do serviço de estoque final.

### 3.6.3 Serviço 3: Ciclo de vida de ativos retornáveis

Para o serviço de ciclo de vida de bens retornáveis temos as seguintes funcionalidades:

- Visualização dos dados sobre os bens retornáveis registrados no estoque, incluindo localização deles, seja na fábrica ou não, e estado deles (sujos e vazios, limpos e vazios, limpos e cheios), conforme informado no item 4.2 deste relatório;
- Atualização do estado e localização de cada um desses ativos retornáveis.

Podemos ver na figura 15, o diagrama de casos de uso para o serviço 3, desde que o usuário esteja logado no sistema. Através dele pode-se ver as ações que podem ser tomadas pelo usuário, inclusive com diferenciação entre usuários com privilégio comum e usuários com privilégio de administrador. Os usuários com privilégio comum podem realizar as ações de visualizar a localização e estado de cada um dos ativos retornáveis, visualizar os indicadores relacionados ao serviço de ciclo de vida de ativos retornáveis, previamente citados no item 4.2 deste relatório, atualizar estado e localização dos mesmos, registrar novos bens, e editar bens já cadastrados no sistema. O usuário com privilégios de administrador, por sua vez, possui todas as ações do usuário comum à sua disposição, juntamente com a ação de deletar bens cadastrados.

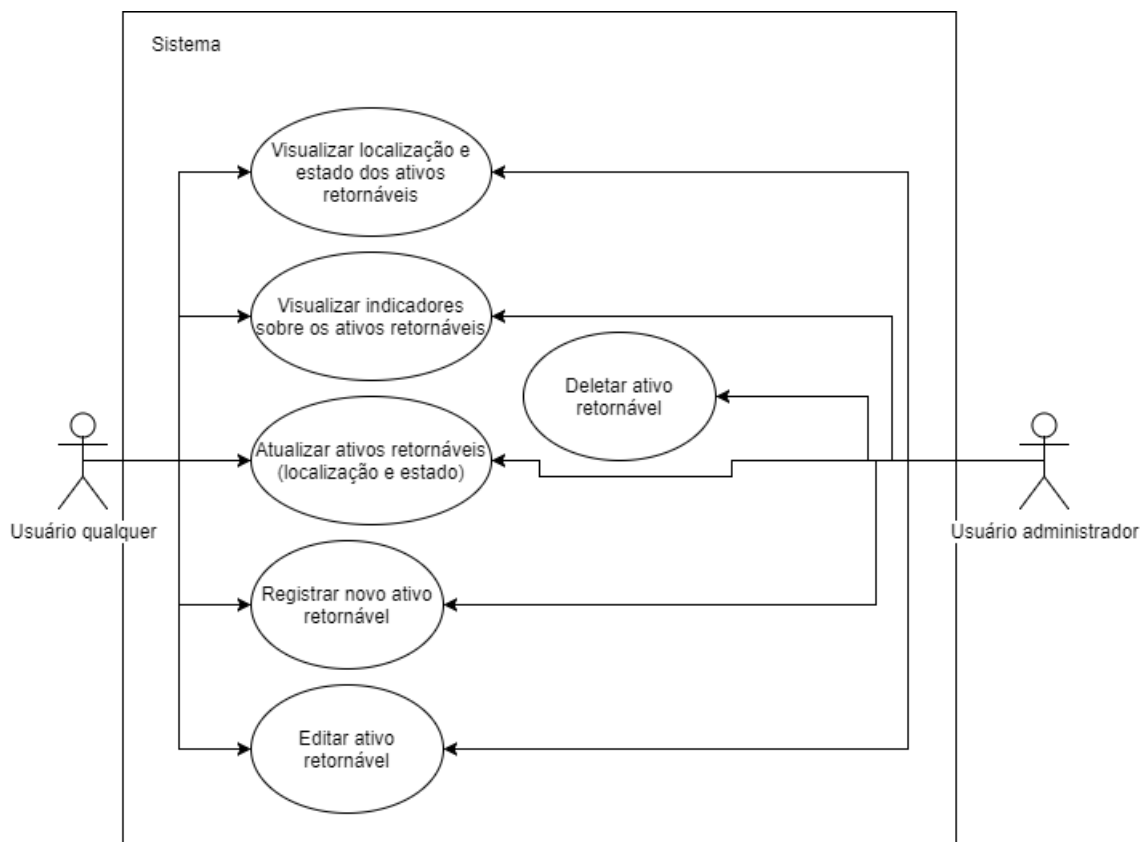


Figura 15 – Diagrama de uso do serviço 3

Podemos ver um exemplo de uso do serviço 3, caso o usuário esteja logado, através do diagrama de sequência simplificado das atividades envolvidas no caso de uso do usuário querer visualizar os dados dos ativos retornáveis, localização e estado atuais, figura 16. Nesse caso, a sequência de eventos é o usuário faz uma requisição ao sistema pela página de visualização da localização e estados dos bens retornáveis atual, após isso o sistema realiza a pesquisa dos dados dos bens retornáveis no banco de dados, o banco de dados retorna os dados consultados, e, posteriormente, o sistema retorna a página com os dados atuais de localização e estado dos bens retornáveis ao usuário, caso os dados existam no

banco de dados.

Caso não haja dados salvos no banco de dados ainda, representando que não houve atualização do estado e localização dos bens ainda, podemos verificar a existência de um tratador de erros, em que o banco de dados retorna nulo para a pesquisa realizada pelo sistema, e o mesmo por sua vez, retorna mensagem ao usuário informando que não há itens atualizados ainda. Após isso, o usuário realiza uma requisição do formulário de atualização das informações dos bens retornáveis ao sistema, que por sua vez, retorna ao usuário a página com o formulário. O usuário utiliza o leitor de código de barras para informar o código de barras do produto que irá ser atualizado no sistema, então, o sistema pesquisa no banco de dados pelo código de barras e retorna as informações sobre o bem retornável para a página de formulário. Após isso, o usuário requisita uma alteração das informações relacionadas àquele bem retornável em questão, o sistema salva as novas informações sobre ele no banco de dados e retorna mensagem ao usuário informando que a alteração dos dados foi realizada com sucesso.

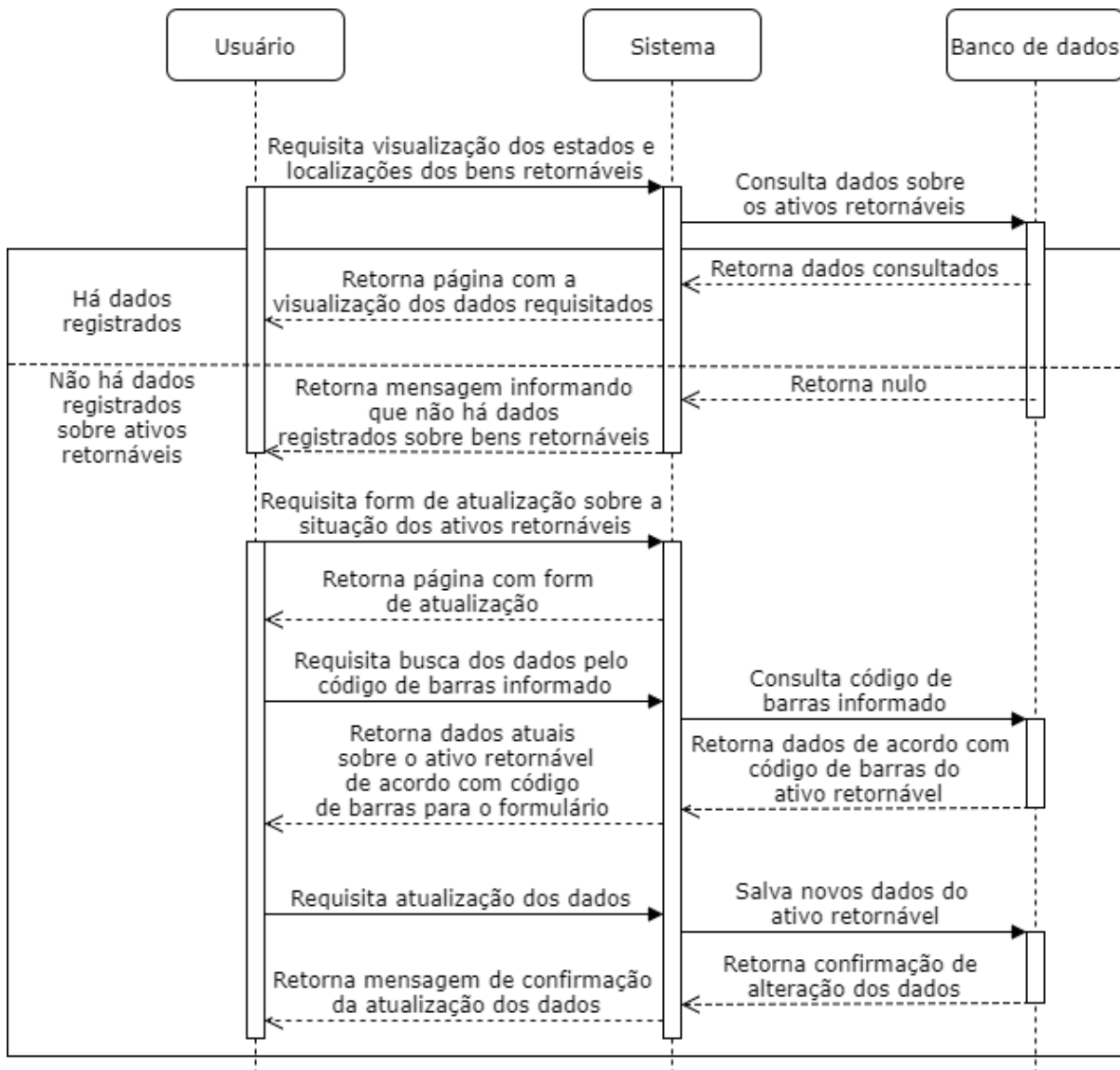


Figura 16 – Exemplo de diagrama de sequência pra caso de uso do serviço 3

Na seção 4.8 poderemos ver mais sobre o funcionamento de cada caso de uso do serviço 3, através de pseudo-código. Também será mostrada a correta sequência a ser seguida para correto funcionamento do serviço de ciclo de vida de bens retornáveis.

### 3.7 Estudo da linguagem e das tecnologias a serem usadas

Este projeto envolve uma grande parte de pesquisa e aprendizado por parte do desenvolvedor, uma vez que não se tinha conhecimento sobre, basicamente, nenhuma das ferramentas utilizadas durante a implementação ou sobre a estrutura de desenvolvimento de um SaaS em nuvem, e nem mesmo sobre a linguagem de programação utilizada.

Tendo em vista isso, podemos perceber que para o correto desenvolvimento desse projeto, foram realizadas extensas pesquisas sobre as tecnologias e ferramentas utilizadas,



como código de barras e programação de SaaS em nuvem, como usá-los e também para entender seu funcionamento. Porém isso ainda foi pouco se comparado com o nível de pesquisa que precisou ser realizada na linguagem de Ruby on Rails, e não só nisso, bem como também em programação web como um todo, pois não se tinha base de programação voltada para esse fim até o início desse projeto. Foi necessário aprender conceitos sobre programação web, pesquisar sobre linguagens utilizadas nesse tipo de projeto, como o próprio Rails, e, também, html, javascript e CSS.

Além disso algumas tecnologias desenvolvidas visando facilitar programação de projetos web e em nuvem também foram pesquisadas, como, por exemplo, Bootstrap, que é um framework de pacotes javascript e CSS com módulos amplamente utilizados nesses projetos.

Para essas pesquisas todas podemos citar diversas fontes, dentre elas, cursos, como "The Complete Ruby on Rails Developer Course" [9], "The Complete Web Developer Course" [10], "Cloud computing com AWS" [11], "Introduction to Cloud Computing with Amazon Web Services" [12], e manuais das próprias ferramentas, como, por exemplo Ruby on Rails [13], Bootstrap [14] e Heroku [15].

### 3.7.1 Heroku

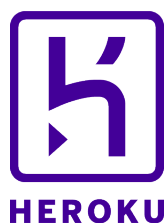


Figura 17 – Logotipo do PaaS Heroku

Heroku é uma plataforma de serviço em nuvem (PaaS), que possui suporte a várias linguagens de programação. Ele roda utilizando um Amazon EC2, que é um IaaS, Infraestrutura como Serviço, segundo informações encontradas em [15]. PaaS e IaaS são semelhantes a SaaS, porém enquanto SaaS é um software oferecido como serviços independentes, o IaaS é a infraestrutura de nuvem comercializada como serviço para que esse SaaS consiga funcionar, e o PaaS por sua vez é uma plataforma oferecida como serviço que seleciona a infraestrutura de nuvem necessária para rodar determinada linguagem e realiza a configuração da mesma de forma automatizada.

Portanto, configurar um novo ambiente para hospedar uma aplicação desenvolvida,

ou um sistema SaaS que é o caso deste projeto, torna-se simples. O Heroku utiliza uma unidade de máquina virtual, o "Dyno", que equivale a uma máquina virtual pequena, de pouca memória RAM e processamento baixo, para realizar a cobrança dos seus serviços. Quanto mais "Dynos" o aplicativo hospedado usar, maior será a conta a ser paga.

Apesar disso, para realizar testes de campo, com baixo uso ainda, ou para aplicações sem muitos usuários ele acaba sendo livre de cobranças, uma vez que utiliza o banco de dados compartilhado PostgreSQL que é gratuito. Este foi um dos motivos para a escolha desse PaaS para este projeto, ao invés de utilizar estruturas em nuvem de empresas bem consolidadas ou com maior renome, como Amazon, Google, entre outras.

Outro motivo que fez o Heroku ser escolhido para este projeto foi a facilidade em realizar a configuração de um novo serviço devido a sua documentação extensa, bem detalhada e atualizada, inclusive em relação à linguagem de programação escolhida para realizar o desenvolvimento do sistema. Sua integração com a IDE, ambiente de programação, online Cloud9 também foi fator positivo a favor da sua escolha para o desenvolvimento deste projeto, uma vez que dessa forma o deployment de novas versões do código era realizado com apenas uma linha de comando no próprio Cloud9.

### 3.7.2 Ruby on Rails



Figura 18 – Logotipo do Ruby on Rails

Ruby on Rails é um framework livre, ou seja, uma união de diversos códigos comuns a projetos de software, que visa agilizar a programação de sites orientados a banco de dados. É programado utilizando a linguagem de programação de código aberto, Ruby e as aplicações criadas através de Ruby on Rails são desenvolvidas seguindo o padrão de arquitetura MVC (Model, View, Controller). Rails ainda segue dois conceitos que visam aumentar a produtividade do desenvolvedor: DRY e Convention over Configuration, que serão explicadas mais a fundo a seguir, segundo informações que podem ser encontradas em [13].

## 3.8 Modelagem do banco de dados

Com as definições de funcionalidades dos serviços, e seus diagramas de uso planejados, e tendo em vista os objetivos, geral e específico, do projeto, chega a hora de realizar a modelagem do banco de dados que será utilizado pelo sistema. A modelagem do banco de dados de maneira correta é essencial para o funcionamento do projeto inteiro, uma vez que é nele que serão armazenadas todas as informações dos 3 serviços e de usuários. Como dito em capítulos anteriores, o banco de dados usado pelo sistema é o relacional PostgreSQL, por ser gratuito a sua utilização.

Nas figuras a seguir, 19 e 20, podemos ver, respectivamente, os modelos conceitual simplificado, e lógico do banco de dados relacional modelado. No primeiro, temos as relações entre as tabelas criadas com suas cardinalidades, por exemplo, a relação entre a tabela Clientes e a Users ocorre de 1 para N, isto é, um Cliente deve possuir 1 User no mínimo e pode ter N User no máximo (1,n). De maneira inversa, a relação entre User e Cliente ocorre de 1 para 1, ou seja, um User deve ter no mínimo 1 Cliente e no máximo 1 Cliente.

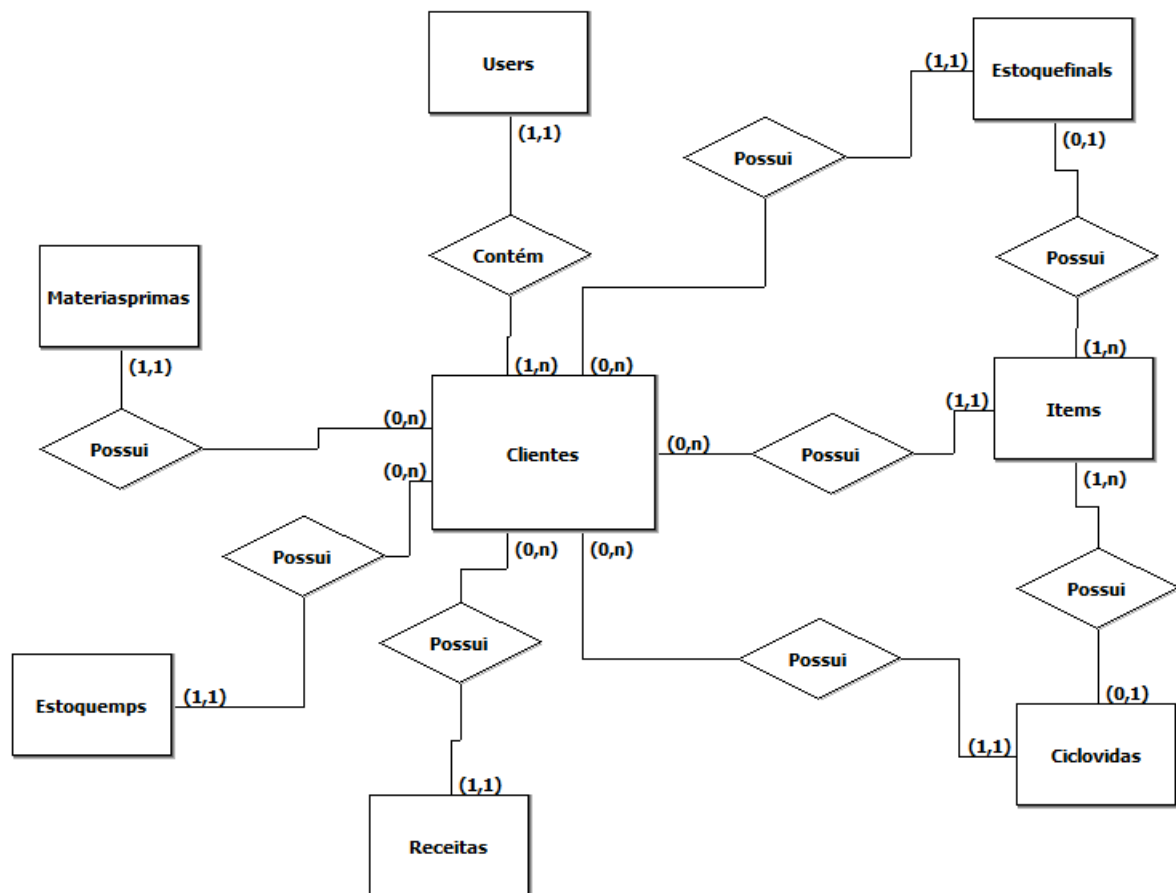


Figura 19 – Modelo Conceitual simplificado do banco de dados

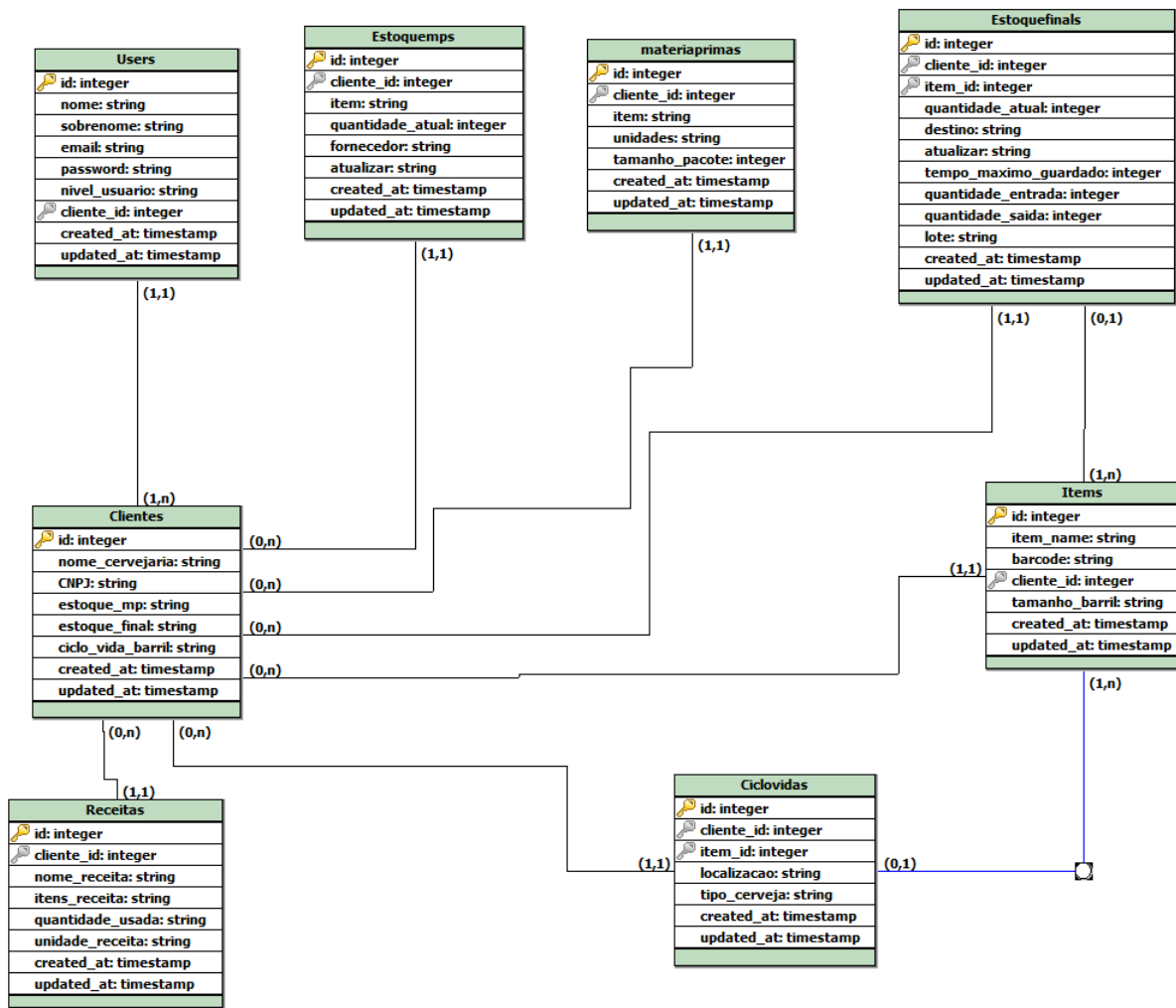


Figura 20 – Modelo lógico do banco de dados

No modelo lógico, figura 20, podemos ver, além das relações entre as tabelas e suas cardinalidades, os campos que fazem parte de cada tabela, juntamente, com os tipos de dados que compõe cada um, que são:

- Tabela Clientes: id (chave primária), nome\_cervejaria, CNPJ, estoque\_mp, estoque\_final, ciclo\_vida\_barril, "created\_at" e "updated\_at";
- Tabela Users: id (chave primária), nome, sobrenome, email, password, nivel\_usuario, cliente\_id (chave estrangeira para Clientes), "created\_at", "updated\_at";
- Tabela Receitas: id (chave primária), cliente\_id (chave estrangeira para Clientes), nome\_receita, itens\_receita, quantidade\_usada, unidade\_receita, "created\_at", "updated\_at";

- Tabela Materiaprimsas: id (chave primária), cliente\_id (chave estrangeira para Clientes), item, unidades, tamanho\_pacote, "created\_at", "updated\_at";
- Tabela Estoquemps: id (chave primária), cliente\_id (chave estrangeira para Clientes), item, quantidade\_atual, fornecedor, atualizar, "created\_at", "updated\_at";
- Tabela Estoquefinals: id (chave primária), cliente\_id (chave estrangeira para Clientes), item\_id (chave estrangeira para Items), quantidade\_atual, destino, atualizar, tempo\_maximo\_guardado, quantidade\_entrada, quantidade\_saida, lote, "created\_at", "updated\_at";
- Tabela Ciclovidas: id (chave primária), cliente\_id (chave estrangeira para Clientes), item\_id (chave estrangeira para Items), localizacao, tipo\_cerveja, "created\_at", "updated\_at";
- Tabela Items: id (chave primária), cliente\_id (chave estrangeira para Clientes), item\_name, barcode, tamanho\_barril, "created\_at", "updated\_at";

## 3.9 Desenvolvimento em Ruby on Rails

Esta seção destina-se a explicar o funcionamento do sistema e sua sequência de funcionamento, com o auxílio de figuras, e do pseudo-código exemplo contido nos apêndices deste documento. Esta seção ainda é responsável por apresentar dois conceitos nos quais o trabalho foi norteado e que também fazem parte da base do Ruby on Rails.

### 3.9.1 DRY

DRY, ou simplesmente, "Don't Repeat Yourself", que significa "não se repita", é o conceito que resulta na técnica de definir nomes, propriedades e códigos apenas em um lugar, e assim, reaproveitar essas informações em outros locais do código.

Um exemplo disso seria que ao invés de possuir uma tabela Pessoas no meu banco de dados e uma classe Pessoa definida também, gerando assim, um "getter"(leitor), um "setter"(modificador), para cada campo da tabela dentro do código programado para o sistema/site, tem-se apenas isso dentro do banco de dados, e através de funcionalidades da linguagem Ruby esses métodos são usados dentro do código escrito para o projeto.

### 3.9.2 Convention over Configuration

"Convention over Configuration"significa "convenção acima de configuração", e o conceito é, basicamente, isso mesmo, respeitar convenções feitas visando facilitar entendimento e manutenção por parte de outros desenvolvedores ao invés de gastar muito tempo realizando configurações. Para citar um exemplo específico de Ruby on Rails, podemos

pegar uma classe Usuário que poderia ser armazenada em uma tabela Consumidores por exemplo, mas pelo conceito de "Convention over Configuration", essa classe será salva na tabela Usuários, poupando tempo de desenvolvimento assim e facilitando a compreensão do que está acontecendo no código.

### 3.9.3 Login e sign up

Como o sistema possui usuários ele precisa de de um sistema de sign up e de login, que são, respectivamente, o cadastro do usuário na primeira vez que utiliza o sistema, e o ato de realizar login no sistema para utilizá-lo, isto é, entrar com seus dados para ser autenticado pelo sistema.

Dessa forma, foi necessário desenvolver um sistema de registro de usuários e de autenticação dos mesmos. Para isso foi utilizada uma ferramenta que o próprio Rails possui internamente, as sessions em conjunto com o banco de dados e com criptografia da senha (password) do usuário para que nem mesmo o desenvolvedor tenha acesso a esse dado e não possa entrar na conta do cliente/usuário.

O funcionamento do cadastro do usuário é realizado da seguinte maneira:

- Usuário acessa a página inicial;
- Usuário clica em cadastrar novo usuário;
- Usuário entra com os dados da cervejaria e escolhe os serviços que deseja usufruir, desde que seja o primeiro usuário dessa cervejaria. Se a cervejaria já tiver sido cadastrada, o sistema não permite que o novo usuário seja criado, já que a criação de novos usuários a partir do segundo, é privilégio de administrador;
- Usuário entra com os dados pessoais dele e escolhe uma senha. Após isso o usuário está criado juntamente com a nova cervejaria cliente do sistema.

As "sessions", no Rails, funcionam, fazendo uma analogia, como se fossem um banco de dados a parte e interno do próprio Rails, sendo criado e destruído a medida que usuário interage com o sistema realizando login e logout. Através da figura 21 e do pseudo-código que pode ser visto nos apêndices deste documento, é demonstrado o funcionamento do login e do logout através das "sessions" a partir da entrada de email e senha por parte do usuário.

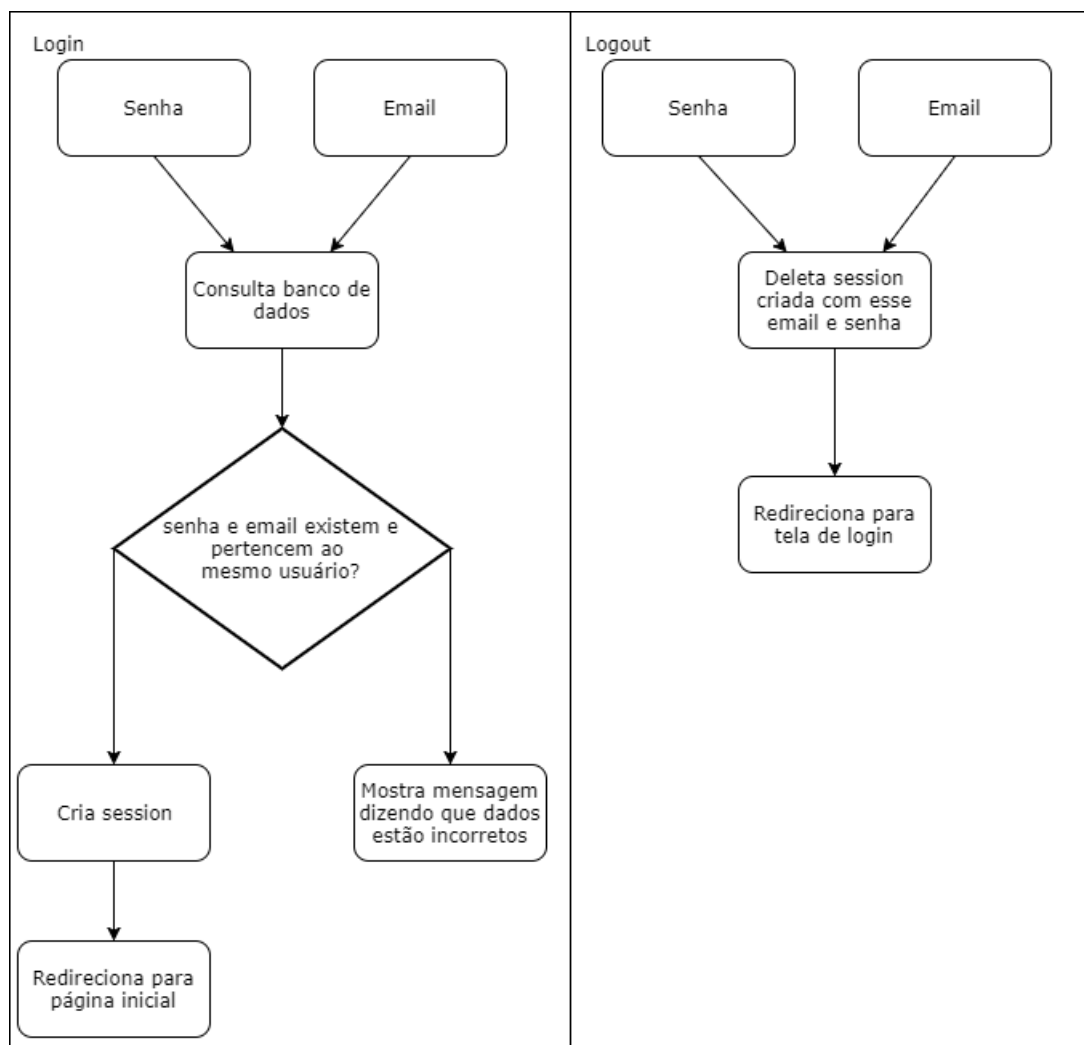


Figura 21 – Fluxograma demonstrando funcionamento de login e logout

### 3.9.4 Funções de administrador

O sistema possui dois diferentes níveis de acesso para os usuários, e por esse motivo é necessário explicar o que um administrador pode realizar que um usuário com nível comum não consegue. Como demonstrado acima, o primeiro usuário que se cadastrar no sistema pela cervejaria será o usuário com privilégios de acesso de administrador. Este usuário terá, além de realizar as mesmas ações de um usuário comum (visualizar informações sobre os serviços, atualizar informações dos serviços, cadastrar novos itens/produtos/ativos retornáveis e visualizar informações atuais a respeito dos estoques ou do ciclo de vida de ativos retornáveis), a capacidade de:

- Escolher os serviços que a cervejaria deseja usufruir;
- Adicionar novas receitas, caso a cervejaria cliente faça uso do serviço 1;

- Visualizar, editar e deletar receitas da cervejaria cadastradas, caso a cervejaria cliente faça uso do serviço 1;
- Acionar novos usuários para a cervejaria;
- Visualizar, editar e deletar usuários a partir da hora que novos usuários estiverem cadastrados;
- Todas as funções de deletar algum item de alguma tabela do banco de dados, sejam elas dados de um serviço ou dados de receita, por exemplo, estão permitidas apenas para os usuários com privilégio de administrador, para manter uma segurança maior nos dados da cervejaria.

### 3.9.5 Serviço 1: Controle de estoque primário

Para o correto funcionamento do serviço 1, com a coleta correta das informações necessárias, uma sequência específica de eventos deve ser seguida.

- Usuário cadastra matérias primas que a cervejaria faz uso e que ficam estocadas no seus estoque primário;
- Usuário atualiza as entradas no estoque primário de matérias primas;
- Administrador cadastra receitas usando os itens que já foram previamente cadastrados como matéria prima;
- A partir disso, todas as outras funções de visualização estão liberadas, e as funções de atualização do estoque e de cadastro, de receita ou de matéria prima, estão liberadas também.

O código do serviço 1 segue a lógica demonstrada a seguir, na figura 22 e no apêndice com pseudo-código neste documento.



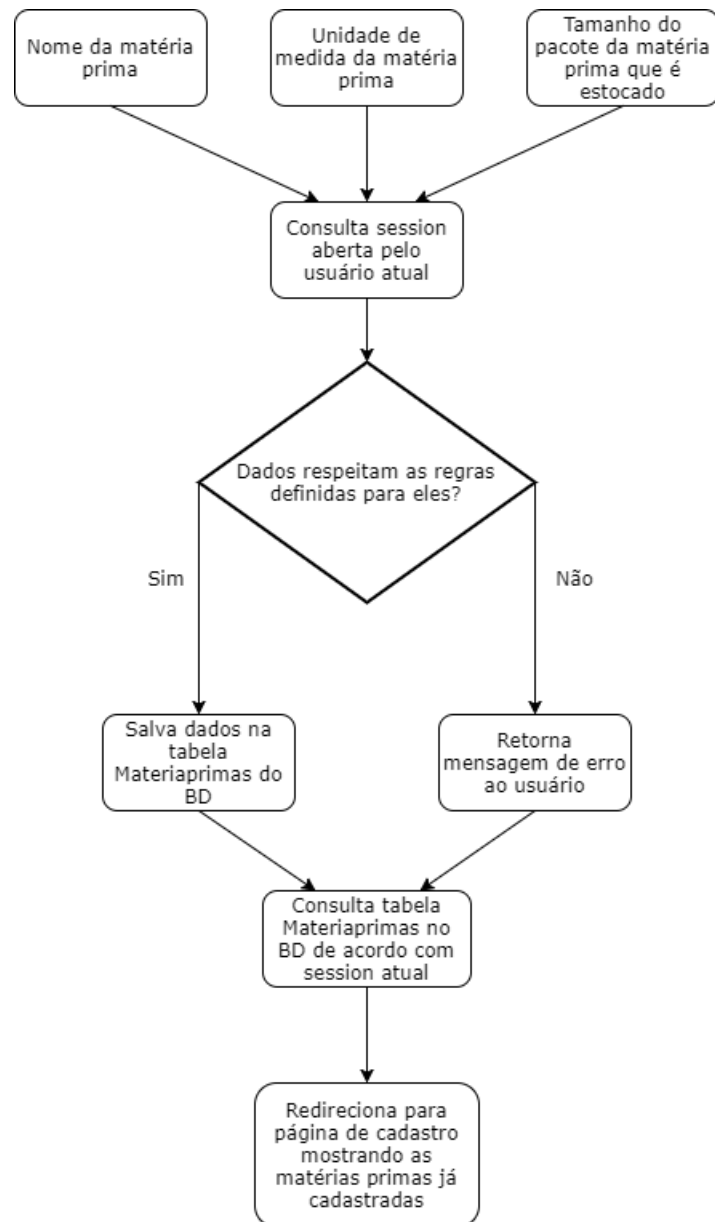


Figura 22 – Fluxograma demonstrando funcionamento do cadastro de matérias primas

### 3.9.6 Serviço 2: Controle de estoque final

Para o correto funcionamento do serviço 2, com a correta coleta de informações necessárias, uma sequência de eventos específica deve ser seguida, como demonstrada abaixo.

- Usuário cadastra produtos que ficam estocados no estoque final da cervejaria;
- Usuário atualiza as entradas de produtos no estoque final da fábrica;
- A partir disso, todas as outras funções de visualização ficam disponíveis, bem como as funções de cadastro de produto e de atualização de estoque final para continuar a

usar o sistema.

O código do serviço 2 segue a lógica demonstrada na figura 23 a seguir, e no pseudo-código presente no apêndice deste documento.

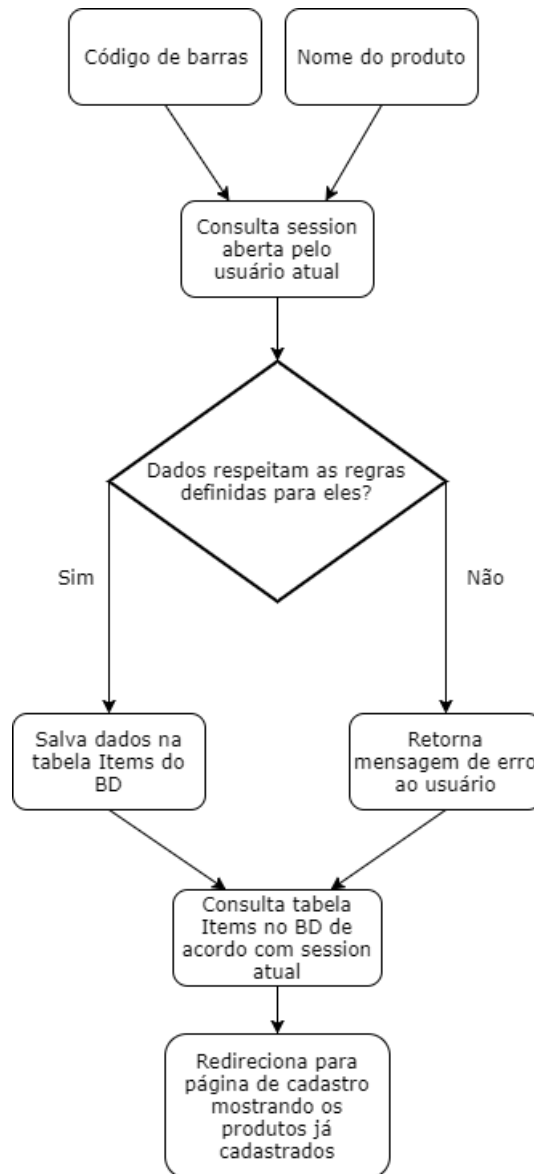


Figura 23 – Fluxograma demonstrando funcionamento do cadastro de produtos

### 3.9.7 Serviço 3: Ciclo de vida de bens retornáveis

Para o correto funcionamento do serviço 3, ciclo de vida de ativos retornáveis, é necessário seguir uma sequência de eventos específica, como demonstrada abaixo.

- Usuário cadastra ativo retornável, seja ele, barril, cilindro de gás ou chopeira;
- Usuário atualiza estado e localização dos ativos retornáveis da cervejaria;

- A partir desse momento, as funções de visualização de ativos retornáveis ficam disponíveis, juntamente, com as funções já utilizadas de cadastro de novos ativos e de atualização do estado/localização desses ativos.

Aqui vale ressaltar que a atualização de estado/localização dos itens é realizada de maneira automática pelo sistema, sem que o usuário necessite digitar essas informações. Tudo que o sistema precisa para realizar a atualização é o código de barras do item em questão. De posse dessa informação, o sistema calcula onde o ativo deve estar e em que situação ele deve estar, uma vez que a localização deles e seu estado possui um fluxo sequencial sempre igual na fábrica.

Tomando como exemplo o ciclo de vida de um barril, podemos ver que inicialmente, a partir da primeira atualização desse barril no serviço 3, o sistema interpreta que ele está no estoque final, com seu estado sendo sujo e vazio. Quando houver uma próxima atualização desse barril no sistema, o sistema interpretará que ele está no estoque, porém agora com seu estado sendo limpo e vazio. Na terceira iteração da atualização desse barril, o sistema interpretará que ele se encontra na câmara fria, cheio com algum tipo de cerveja, que deverá ser informada juntamente com o código de barras no momento da atualização. Na quarta iteração dessa atualização desse barril, o sistema interpretará que ele está indo para fora da fábrica. O destino deverá ser informado pelo usuário no momento dessa atualização. A partir da hora que o sistema enxergar o barril como estando fora da fábrica, será disparado um contador de dias que será salva no banco de dados dessa atualização desse barril para que o sistema possa informar ao usuário quando um barril deve ser recolhido, a partir de 6 dias que saiu da fábrica, e a localização dele para recolha.



## 4 Implementação do sistema e Resultados

Este capítulo tem por finalidade mostrar a implementação do sistema juntamente com as escolhas realizadas em questão de tecnologias disponíveis, e suas justificativas para tal. Posteriormente, este capítulo também mostrará os resultados obtidos com a implementação do sistema.

### 4.1 Implementação do sistema

Como já explicado, anteriormente, Ruby on Rails é um framework da linguagem Ruby que possui a finalidade de facilitar a programação de sistemas web baseados em banco de dados e executado através de MVC, portanto, a escolha dele como linguagem de programação se mostra óbvia frente a outras linguagens.

Com a finalidade de facilitar a implementação do sistema como um todo, o PaaS Heroku foi escolhido para realizar a implementação do sistema. As razões que levam à essa facilidade frente a outras alternativas, como Amazon EC2 ou outras clouds de outras empresas, estão explicitadas no Capítulo 3, porém podemos citar dentre elas a integração do Heroku com a IDE utilizada para programação do sistema, Cloud9, permitindo deploy e execução do projeto com apenas um comando, bem como a ausência da necessidade de realizar extensas configurações de servidor e seus preços acessíveis, contendo inclusive uma gama de serviços gratuitos.

Pela escolha do Heroku como plataforma para realizar o deploy e execução do projeto, o banco de dados ficou limitado a PostgreSQL, uma vez que ele é gratuito para uso e também é suportado pelo próprio Heroku.

Além disso, para método de entrada de dados no sistema, nos serviços 2 e 3, foi escolhido o código de barras com seus respectivos leitores. As razões para tal foram explicitadas no Capítulo 3, mas dentre elas podemos citar o baixo custo, apesar da necessidade de intervenção humana, dessa tecnologia em comparação com outras tecnologias presentes no mercado como RFID, por exemplo.

Como último ponto a ser escolhido está o método utilizado para visualização dos dados e análise dos mesmos de acordo com os indicadores propostos, ou seja, a dashboard e sua modelagem escolhida, que será mostrado a seguir.

### 4.1.1 Escolha e Modelagem do Dashboard

Para mostrar os indicadores descritos na seção 4.2, faz-se necessário o uso de um Dashboard. Um Dashboard é nada mais que um painel de controle, onde são apresentadas visualmente as informações mais importantes de acordo com os indicadores.

Como Ruby on Rails é um framework otimizado para programação web, e por ser open source e ter uma comunidade ativa no desenvolvimento, ele contém um grande leque de opções de templates de Dashboard e outros espalhados pela internet, uma vez que esses templates, em sua maioria contém arquivos html, css e javascript e essas linguagens são aceitas em Rails também por serem linguagens de desenvolvimento web. Por esse motivo e para poupar tempo do desenvolvimento do sistema foi pesquisado e escolhido um Dashboard que se encaixava nas necessidades do projeto. Foi escolhido o template de Dashboard "Gentelella on Rails", por possuir integração e otimização para o uso com Ruby on Rails e também por ser gratuito e de uso livre. A figura 24 demonstra uma das telas desse template.

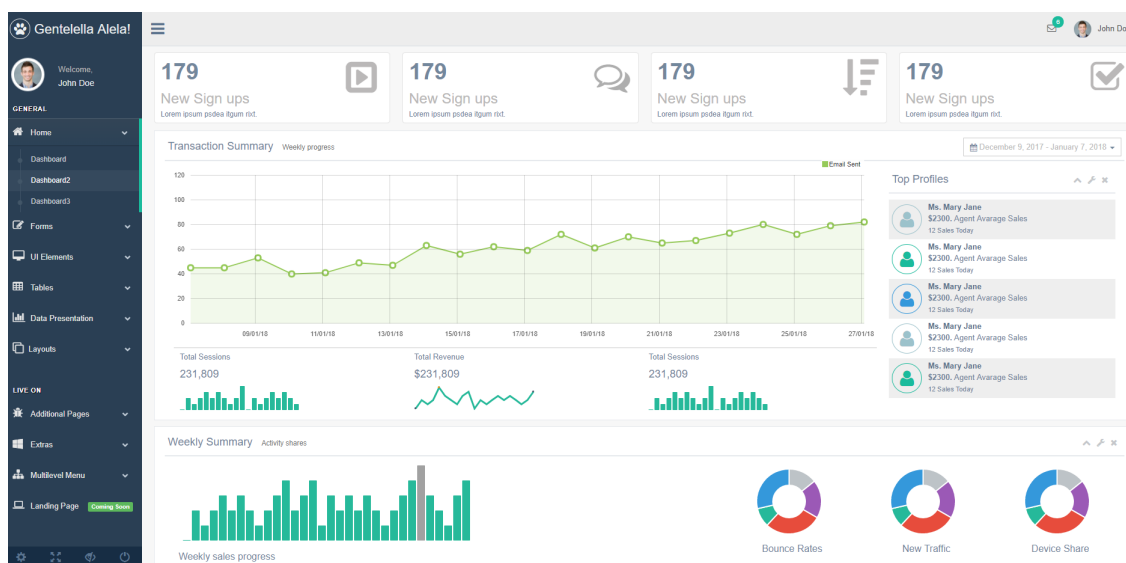


Figura 24 – Template de Dashboard Gentelella on Rails

## 4.2 Resultados

Como resultado final desse trabalho obteve-se a implantação de um sistema contendo 3 serviços com o intuito de atender às especificações feitas e também de representar os indicadores definidos no início do trabalho.

O projeto inclui o mapeamento de necessidades e indicadores desejados pelas microcervejarias da região de Florianópolis, a modelagem do banco de dados relacional utilizado para os 3 serviços, implantação de um Dashboard para visualização mais amigável dos indicadores propostos pelos usuários, a modelagem dos serviços que seriam necessários

para atender aos requisitos, incluindo o grau de granularidade deles e suas funcionalidades. Posteriormente, começando a construção de um sistema em Ruby on Rails para interação com o usuário na coleta de dados em tempo real, utilizando formulários e/ou leitores de código de barras, e visualização de indicadores através de consultas ao banco de dados e cálculos realizados no lado do servidor do código, e sua posterior publicação para uso utilizando o PaaS Heroku.

O início do projeto deu-se com a o mapeamento de indicadores e necessidades das cervejarias. Uma vez tendo isso mapeado, partiu-se para a modelagem do banco de dados, seguido pela implantação de um Dashboard e da programação em Ruby on Rails do sistema.

Lembrando que a metodologia de desenvolvimento utilizada foi a iterativa incremental, houve bastante interação com a cervejaria Kairós, que foi parceira nesse projeto cedendo tempo espaço para realização de testes, gerando inúmeros feedbacks de melhorias ao sistema e, conseqüentemente, reportes de bugs no programa.

O desenvolvimento do código em si, acabou por ser dividido entre os serviços em si, e suas funcionalidades, e algumas funcionalidades necessárias no geral, como login de usuários e cadastro dos mesmos. A divisão do desenvolvimento do código, pós escolha e implantação do template de Dashboard, ficou caracterizada da seguinte forma:

1. Cadastro de usuários e clientes, inclusive nível de hierarquia nos usuários, usuário administrador e usuário comum, e também separação de dados por cliente;
2. Login de usuários cadastrados, os "sessions" explicados no Capítulo 4;
3. Serviço 1, funcionalidade de cadastro de novas matérias primas, contando com o desenvolvimento do formulário usado para isso, a parte de acesso a banco de dados para salvar os dados informados pelo usuário, e também a parte de regras de entrada de dados, como unicidade dos dados informados por exemplo;
4. Serviço 1, funcionalidade de atualização da situação do estoque primário;
5. Serviço 1, funcionalidade de visualização das informações do estoque atual, como quantidade de cada item presente no estoque;
6. Serviço 1, funcionalidade de visualização de indicadores, como informação de uso de fornecedores ou ainda, informações sobre a variação, entradas e saídas por mês, do estoque por item presente nele;
7. Serviço 1, capacidade de editar e deletar matérias primas cadastradas, somente pelo administrador;

8. Serviço 1, funcionalidade de cadastro de receitas por parte de usuários administradores, e verificação de quais receitas podem e não podem ser produzidas a partir do estoque atual;
9. Serviço 1, capacidade de visualizar, editar e deletar receitas cadastradas por parte de usuários administradores;
10. Serviço 2, funcionalidade de cadastro de novos produtos fazendo uso do leitor de código de barras;
11. Serviço 2, capacidade de editar e deletar produtos cadastrados no sistema por parte do usuário administrador;
12. Serviço 2, funcionalidade de atualização das informações sobre o estoque final utilizando leitor de códigos de barras;
13. Serviço 2, funcionalidade de visualização da quantidade de cada item presente no estoque final;
14. Serviço 2, funcionalidade de visualização de indicadores relacionados ao serviço 2, como por exemplo, quantidade de cada produto produzido e a quanto tempo cada um deles está parado no estoque;
15. Serviço 3, funcionalidade de cadastro de ativos retornáveis da fábrica;
16. Serviço 3, capacidade de editar e deletar ativos retornáveis cadastrados;
17. Serviço 3, funcionalidade de atualização de informações sobre o estado e a localização dos ativos retornáveis;
18. Serviço 3, funcionalidade de visualização de informações sobre localização e estado dos ativos retornáveis, bem como de indicadores relacionados a eles;
19. Funcionalidades relacionadas a usuários com privilégios de administrador, como criação, edição e destruição de outros usuários, da mesma cervejaria, com privilégio comum.

Cada uma dessas partes descritas acima foi cumprida ao final do projeto e de sua posterior disponibilização para uso por parte da primeira cervejaria cliente do sistema, a Kairós. A seguir podemos ver exemplos de algumas telas, de visualização e de interação com os usuários, desenvolvidas no decorrer do projeto.





Figura 25 – Tela inicial com menus laterais sobre os serviços

Na figura 25 podemos perceber que dados sobre os serviços que a cervejaria contratou são mostrados, contendo se a cervejaria está em dia com o pagamento dos serviços, ainda a ser implementado futuramente o pagamento por meio do sistema. Ainda nessa figura, podemos ver uma tabela sobre os barris que estão a mais de 6 dias fora da fábrica e precisam ser recolhidos contendo a localização deles também, exemplificando assim, a implementação do item 18 mostrado acima.

### Cadastro de matéria prima

Cliente: teste

Item: Nome do item

Unidade de medida: quilograma

Tamanho pacote: #

**Cadastrar matéria prima**

---

### Matéria prima cadastrada

Show: 5 entries

Id	Matéria prima	Tamanho do pacote	Unidade de medida	Editar	Deletar
2	malte 1	20	Kg	Editar	Delete
3	tampinha	400	unidade	Editar	Delete

Showing 1 to 2 of 2 entries

Previous 1 Next

Figura 26 – Tela exemplo do funcionamento dos cadastros, sem menus a mostra

Na figura 26, conseguimos visualizar um exemplo de como são as telas de cadastro de itens, sejam esses itens matérias primas, ativos retornáveis, usuários ou produtos, uma vez que as páginas desses cadastros são análogas contando com apenas a diferença entre elas se todas as informações do cadastro são informadas através de formulário digitado ou se são informadas através de código de barra lido pelo leitor. Além do formulário para entrada de dados, podemos ver também nessa imagem um exemplo de tabela usada em todas as páginas de cadastro de itens para visualizar quais itens já foram cadastrados. Essa figura é responsável por exemplificar os itens 3, 10 e 15, bem como do item 1, que fazem referência, respectivamente, aos Serviços, 1, 2 e 3 e à função de administrador referente ao cadastro de usuários. Na tabela que consta nessa figura ainda podemos ver exemplo dos itens 7, 11 e 16.

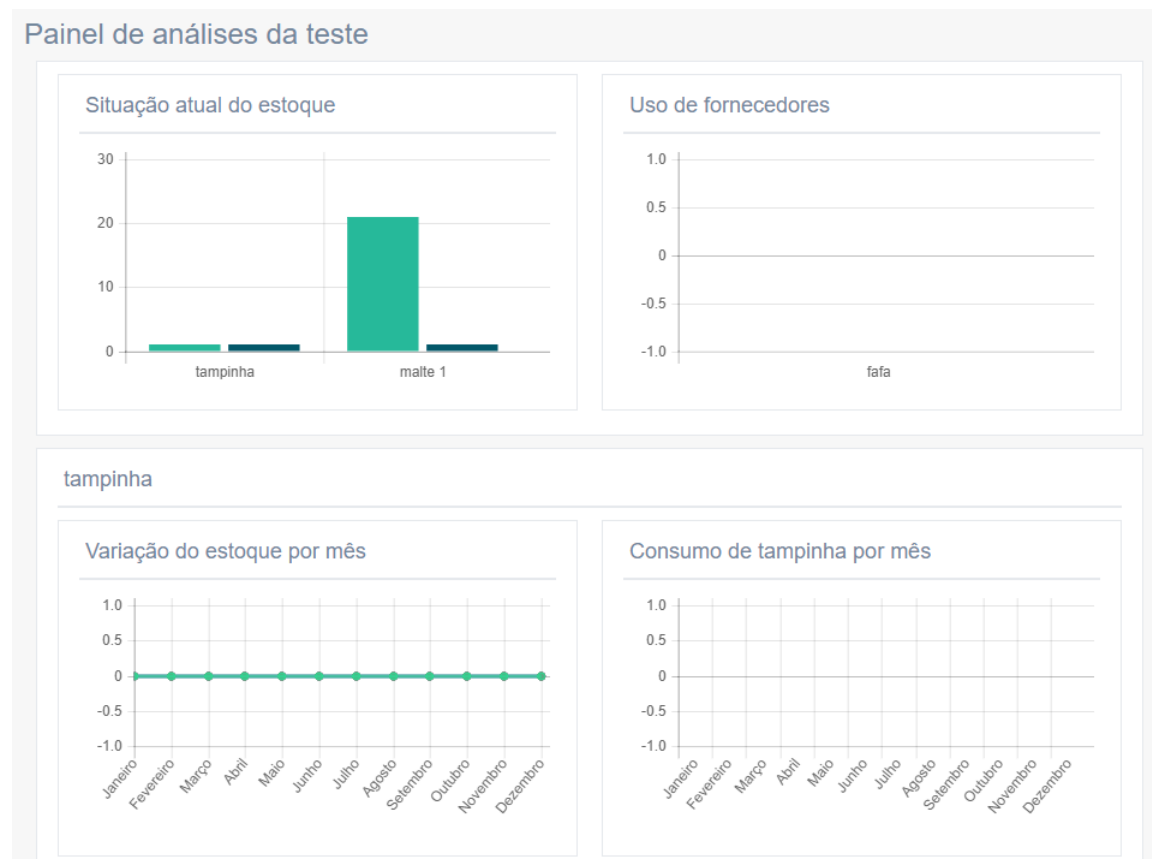


Figura 27 – Tela exemplo do Serviço 1, de visualização dos indicadores, sem menus a mostra.

Na figura 27 temos a mostra um exemplo de como é a visualização de alguns indicadores, entre eles a situação atual do estoque com as quantidades anterior (azul) e atual (verde) dos diversos itens mostrados em um gráfico de barras intitulado "Situação atual do estoque". Ao lado desse gráfico, temos outro gráfico de barras, intitulado "Uso de fornecedores", o qual nos fornece a quantidade de usos de cada fornecedor. Ainda nessa imagem podemos ver as informações sobre um item, o item "tampinha", sua variação no

estoque contando com informações de quantidade de entradas e saídas no mês e também a quantidade presente no estoque ao final do mês através de um gráfico de linha. Ao lado desse gráfico, ainda na parte das informações sobre o item "tampinha", temos um gráfico de barras nos mostrando o consumo desse item por mês. Aqui nesse exemplo podemos ver exemplificado o item 6 do serviço 1, e também os itens 14, do serviço 2, e 17, do serviço 3.

**Atualização de produtos no estoque**

Nome cervejaria  
teste

Codigo de barras  
Código de barras

Item a ser atualizado

Quantidade alterada  
Quantidade de entrada/saída

Destino  
Destino do produto se estiver saindo da fábrica

Lote  
Lote do produto produzido

Atualizar  
Entrada ▾

**Atualizar estoque final**

Figura 28 – Tela exemplo do Serviço 2, de atualização do nível do estoque final atual, utilizando leitor de código de barras como entrada de dados.

Na figura 28, temos o exemplo de uma das telas de atualização de dados de estoque ou de ativos retornáveis dos serviços, em específico do serviço 2 nesta figura. Ainda nela podemos notar o uso do leitor de código de barras como entrada de dados para atualização do estoque final. No exemplo do serviço 2, a atualização é feita por meio de um formulário onde o usuário insere o código de barras utilizando o leitor de código de barras e o sistema busca os dados restantes do produto a ter valor atualizado no estoque final. Após isso, o usuário define também se é entrada ou saída o tipo de atualização a ser feita e informações de quantidade e de lote referente à atualização. Nesta figura podemos ver exemplificado os itens 4, 12 e 17, apresentados anteriormente.

Receitas cadastradas

Ingredientes e quantidades para fabricar stout

Show  entries Search:

Id	Nome da receita	Item usado na receita	Quantidade usada na receita	Unidade de medida do item	Editar	Deletar
1	stout	tampinha	600	unidade	Editar	Delete
2	stout	malte 1	500	Kg	Editar	Delete

Showing 1 to 2 of 2 entries Previous 1 Next

---

Ingredientes e quantidades para fabricar pilsen

Show  entries Search:

Id	Nome da receita	Item usado na receita	Quantidade usada na receita	Unidade de medida do item	Editar	Deletar
3	pilsen	tampinha	200	grama	Editar	Delete

Showing 1 to 1 of 1 entries Previous 1 Next

Figura 29 – Tela exemplo da visualização de receitas com seus respectivos ingredientes

Cadastro de receita

Ciente

Nome receita

Itens usados na receita  Quantidade usada  Unidade receita

---

Receitas cadastradas

Show  entries Search:

Id	Nome da receita	Item usado na receita	Quantidade usada na receita	Unidade de medida do item	Editar	Deletar
1	stout	tampinha	600	unidade	Editar	Delete
2	stout	malte 1	500	Kg	Editar	Delete
3	pilsen	tampinha	200	grama	Editar	Delete
5	red lager	tampinha	30	unidade	Editar	Delete

Showing 1 to 4 of 4 entries Previous 1 Next

Figura 30 – Tela exemplo de cadastro e edição de receitas cadastradas pela cervejaria

Nas figuras 29 e 30, podemos ver, respectivamente, como é a tela de visualização das receitas, e a tela de edição das receitas com seus respectivos links de edição e exclusão. Na primeira, temos a visualização de cada receita e seus respectivos ingredientes em tabelas separadas por receita. Já na segunda, temos a página de cadastro das receitas

que funciona da seguinte forma, o usuário cadastra as informações de cada ingrediente separadamente identificando a qual receita ele pertence. Abaixo disso, ainda na segunda figura temos podemos ver uma tabela com os ingredientes cadastrados e suas respectivas receitas. Juntas essas duas figuras exemplificam os itens 8 e 9 mostrados anteriormente.

### Estoque da teste

#### Estoque de barris

Show  entries Search:

Item	Capacidade do barril	Localização do produto	Tempo no cliente, caso fora da fábrica	Tipo de cerveja, caso barril esteja na câmara fria	Data da última atualização
barril 2	5L	Estoque de barris. Limpo e vazio			2017-12-28
barril1	15L	Estoque final. Sujo e vazio			2017-12-28

Showing 1 to 2 of 2 entries Previous 1 Next

#### Barris fora da fábrica a mais tempo

Show  entries Search:

Item	Capacidade do barril	Localização do produto	Tempo no cliente, caso fora da fábrica	Tipo de cerveja, caso barril esteja na câmara fria	Data da última atualização
No data available in table					

Showing 0 to 0 of 0 entries Previous Next

#### Informações dos barris

##### Quantidade de barris em cada localização

Show  entries Search:

Barris sujos	Barris limpos	Barris na câmara fria	Fora da fábrica
1	1	0	0

Showing 1 to 1 of 1 entries Previous 1 Next

##### Quantidade de barris em cada localização por capacidade

Show  entries Search:

Capacidade do barril	Barris sujos	Barris limpos	Barris na câmara fria	Fora da fábrica
15L	1	0	0	0
5L	0	1	0	0

Showing 1 to 2 of 2 entries Previous 1 Next

Figura 31 – Tela exemplo do Serviço 3, de visualização da localização e estado dos barris, de quais barris estão fora da fábrica a mais tempo e também sobre quantidade de barris em cada localização geral e por capacidade deles, sem menus a mostra.

Na figura 31, podemos ver as informações sobre localização dos ativos retornáveis,

aqui exemplificados pelos barris, bem como quais barris estão há mais tempo fora da fábrica e necessitam de recolha, e também, as quantidades de barris em cada localização de maneira geral e de maneira específica por capacidade do barril. Isso exemplifica o item 18 mostrado anteriormente.

**Estoque da teste**

**Estoque de matéria prima**

Show  entries Search:

Id do item	Item	Quantidade atual em estoque	Fornecedor	Última atualização	Data da última atualização
3	malte 1	21	fafa	Entrada	2017-11-30
5	tampinha	1	fafa	Entrada	2017-12-27

Showing 1 to 2 of 2 entries Previous 1 Next

---

**Possível produção com nível de estoque atual**

Show  entries Search:

Receita	Disponibilidade com estoque atual
pilsen	Disponível para produção
red lager	Não disponível para produção
stout	Não disponível para produção

Showing 1 to 3 of 3 entries Previous 1 Next

Figura 32 – Tela exemplo do Serviço 1, de visualização do nível do estoque atual, juntamente, com receitas que podem ou não ser produzidas, sem menus a mostra.

Por último, mas não menos importante, temos o exemplo da figura 32, que nos mostra a tela de visualização da quantidade de cada matéria prima presente em estoque na primeira tabela, já na segunda tabela temos a informação sobre as receitas que podem ser ou não produzidas de acordo com as quantidades de matéria prima cadastradas junto à receita e com as quantidade de cada matéria prima presente em estoque. Nesta figura também pode-se notar exemplo da implementação dos itens 5, referente ao serviço 1, e analogamente, dos itens 13, do serviço 2, e 18, do serviço 3.

Testes de uso foram realizados no sistema, durante o desenvolvimento e por usuários reais da cervejaria Kairós inclusive, e os eventuais problemas encontrados foram consertados. Melhorias sugeridas também já foram realizadas de acordo com as imagens acima.

Além disso, vale ressaltar que como o sistema foi feito para ser acessado a partir de qualquer dispositivo ele é responsivo, o que significa que o sistema adequa o layout das telas de acordo com o tamanho da tela do dispositivo que o está acessando, seja ele um celular, um tablet, ou um computador, como podemos ver na figura 33.



Figura 33 – Tela exemplo do Serviço 1, de visualização das receitas que podem ou não ser produzidas, sem menus a mostra, a partir da tela de celular.

Também é digno de nota que o sistema implementado atende as características de multi usuários, onde podemos ter múltiplos usuários de diferentes cervejarias acessando ao mesmo tempo, cada um visualizando seus próprios dados, pois, desde o estado atual do sistema, não há necessidade alguma de realizar alterações no sistema para que ele seja implementado em mais de uma cervejaria ao mesmo tempo.

Pode-se dizer que o objetivo geral de desenvolver um sistema web de apoio à gestão de produção com a coleta dos dados de estoque, primário e final, e ativos retornáveis utilizando metodologia orientado a serviços, para que possa ser disponibilizado para mais de uma cervejaria ao mesmo tempo, e que a cervejaria cliente possa escolher os serviços

mais adequados à sua necessidade foi cumprido. A partir das figuras mostradas nesta seção podemos perceber também o cumprimento dos requisitos funcionais relacionados ao sistema, apresentados no Capítulo 3.

Sendo assim, utilização do sistema pela cervejaria tende a diminuir o atraso na atualização das informações referentes aos estoques primário, final e aos ativos retornáveis da fábrica, e com isso a análise dos indicadores e dados poderá ser realizada de forma mais rápida também, contribuindo para uma gerência da fábrica mais ágil e capaz de mudar de acordo com as tendências de mercado.

Atualmente, já podemos ver melhoras, com o uso do sistema na cervejaria Kairós, como, por exemplo, a agilidade na atualização dos dados de estoque mantendo eles sempre atuais e condizentes com a realidade, bem como a implantação de uma política de rotação nos ativos retornáveis para esticar a vida útil dos mesmos, uma vez que agora a cervejaria possui controle sobre a quantidade de usos dos mesmos.



## 5 Conclusão e Perspectivas

O desenvolvimento do projeto ocorreu em parceria com a microcervejaria de Florianópolis, Kairós, e atingiu, ao seu final, os resultados esperados por eles, tanto é, que eles já são os primeiros clientes dos três serviços desenvolvidos. Graças a isso, a Kairós já quer que o trabalho continue para que eles possam usufruir de outros serviços, os quais eles desejam.

O sistema em si está em fase de adaptação na Kairós, fato este que deve ocorrer até o final do mês de fevereiro, para que a funcionalidade de pagamento de assinatura pelos serviços usados seja adicionada e testada no sistema. Esse período de adaptação também servirá para que os custos operacionais do sistema, com estrutura, de acordo com o uso dele possam ser definidos em função de uma média.

Podemos perceber, através do explicitado neste documento, que os objetivos específicos listados no Capítulo 1 foram atendidos, as principais necessidades dos clientes foram levantadas e com isso foram especificados os serviços que fazem parte do sistema atualmente. Para a realização do projeto houve a necessidade de um extenso estudo das tecnologias relacionadas a ele, inclusive da linguagem de programação, para posterior modelagem do banco de dados usado e do dashboard utilizado. A partir disso, foi criado um sistema de aquisição dos dados utilizando-se em alguns momentos de código de barras para isso, bem como toda a visualização de dados definidos pelos indicadores.

Por ser um projeto desenvolvido, quase que inteiramente, utilizando tecnologias e ferramentas que não eram conhecidas, demandou um esforço gigantesco para aprendizado e implementação das mesmas.

Algumas limitações estão presentes no sistema atualmente, por exemplo, a funcionalidade de cobrança de mensalidade, pelo uso do sistema, no próprio sistema ainda não está implementada. Também há uma grande limitação em termos de que hoje para que os serviços 2 e 3 possam funcionar corretamente, a cervejaria cliente necessita ter ao menos 1 leitor de código de barras para poder realizar a leitura dos mesmos presentes nos ativos retornáveis ou produtos que estão no estoque.

Futuramente, a ideia é expandir o sistema, tanto em quantidade de serviços e funcionalidades, como em clientes, quem sabe até incluindo serviços voltados para os clientes final da cerveja, fornecedores e distribuidores. Outra ideia que surgiu durante o desenvolvimento do sistema, foi a possível inclusão e integração dos serviços aqui desenvolvidos como um módulo dentro de outro sistema já existente, como um dos ERPs presentes no mercado hoje que são voltados para microcervejarias.



# Referências

- 1 MAPA. Estudo do mapa sobre o mercado cervejeiro no brasil. <https://www.cervesia.com.br/artigos-tecnicos/dados-estatisticos/5073-estudo-do-mapa-sobre-o-mercado-cervejeiro-no-brasil.html>, 2017. Citado na página 15.
- 2 NEVES, M. A. O. Tudo sobre indicadores de desempenho em logística. *Revista Mundo Logística*, 2009. Citado na página 18.
- 3 BRASIL, E. 5 indicadores de desempenho para medir o seu sucesso. <https://endeavor.org.br/indicadores-de-desempenho/>, 2015. Citado na página 22.
- 4 BEZERRA, E. Base de dados é a base de tudo. <https://endeavor.org.br/base-de-dados-e-a-base-de-tudo/>, 2012. Citado na página 22.
- 5 LEITE, A. Metodologia de desenvolvimento de software. <https://www.devmedia.com.br/metodologia-de-desenvolvimento-de-software/1903>, Desconhecido. Citado na página 23.
- 6 LARMAN, V. R. B. C. Iterative and incremental developments. a brief history. *Computer*, 2003. Citado na página 24.
- 7 DESCONHECIDO. Relational database. <https://aws.amazon.com/pt/relational-database/>, Desconhecido. Citado na página 25.
- 8 <https://www.gs1br.org>. Citado na página 27.
- 9 PERCIVAL, R. The complete ruby on rails developer course. <https://www.udemy.com/>. Citado na página 47.
- 10 PERCIVAL, R. The complete web developer course. <https://www.udemy.com/>. Citado na página 47.
- 11 FERREIRA, F. Cloud computing com aws. <https://www.udemy.com/>. Citado na página 47.
- 12 SKILLS, I. Introduction to cloud computing with amazon web services. <https://www.udemy.com/>. Citado na página 47.
- 13 <http://rubyonrails.org/>. Citado 2 vezes nas páginas 47 e 48.
- 14 <https://getbootstrap.com/>. Citado na página 47.
- 15 <https://www.heroku.com>. Citado na página 47.



# Apêndices



# APÊNDICE A – Exemplos de pseudo-códigos

## **Pseudo-código para Login:**

pega senha

pega email

consulta banco de dados verificando se existem e se estão ok

se consulta ok

cria session com os dados do usuário e do cliente (cervejaria)

envia para página inicial do sistema

se consulta não ok

reproduz mensagem dizendo que os dados de login estão incorretos para o usuário

## **Pseudo-código para Logout:**

pega senha

pega email

deleta session criada com esse email e senha

## **Pseudo-código para cadastro de nova matéria prima**

Recebe nome da matéria prima, unidade de medida da matéria prima, tamanho do pacote da matéria prima que é estocado na fábrica a partir dos dados digitados pelo usuário realizando cadastro

Recebe dados do cliente (cervejaria) diretamente do banco de dados de acordo com a session que está sendo executada, ou seja, de acordo com o usuário que está realizando cadastro da matéria prima

Se os dados informados respeitam as regras definidas para eles

Salva nome, unidade\_medida, cliente\_id, tamanho\_pacote no banco de dados na tabela Materiaprimsas nas suas devidas colunas

Se os dados informados não respeitam as regras definidas para eles

Retorna mensagem de erro ao usuário informando os erros

Consulta tabela Materiaprimsas no banco de dados para pegar as informações sobre as matérias primas cadastradas, de acordo com cliente\_id, ou seja, de acordo com a

cervejaria da qual o usuário faz parte

Retorna para a pagina de cadastro de matéria prima mostrando as matérias primas cadastradas até o momento

### **Pseudo-código para cadastro de novos produtos**

Recebe código de barras a partir do código de barras que o usuário coloca utilizando o leitor

Recebe nome do produto a partir do que for digitado pelo usuário realizando o cadastro

Recebe dados do cliente (cervejaria) diretamente do banco de dados de acordo com a session que está sendo executada, ou seja, de acordo com o usuário que está realizando cadastro do produto

Se os dados informados respeitam as regras definidas para eles

Salva nome e código de barras no banco de dados na tabela Items nas suas devidas colunas

Se os dados informados não respeitam as regras definidas para eles

Retorna mensagem de erro ao usuário informando os erros

Consulta tabela Items no banco de dados para pegar as informações sobre os produtos cadastrados, de acordo com cliente\_id, ou seja, de acordo com a cervejaria da qual o usuário faz parte

Retorna para a página de cadastro de produtos mostrando os produtos cadastrados até o momento