



**Fundação Educacional do Município de Assis
Instituto Municipal de Ensino Superior de Assis
Campus "José Santilli Sobrinho"**

GUILHERME GOMES BELANDA

**SISTEMA DE FECHAMENTO PARA LEVANTAMENTO DE INSUMOS
GASTOS NO SETOR DE BEBIDAS**

Assis/SP

2024



**Fundação Educacional do Município de Assis
Instituto Municipal de Ensino Superior de Assis
Campus "José Santilli Sobrinho"**

GUILHERME GOMES BELANDA

**SISTEMA DE FECHAMENTO PARA LEVANTAMENTO DE INSUMOS
GASTOS NO SETOR DE BEBIDAS**

Trabalho de conclusão de curso apresentado ao Curso Superior de Análise e Desenvolvimento de Sistemas do Instituto Municipal de Ensino Superior de Assis – IMESA e a Fundação Educacional do Município de Assis – FEMA, como requisito parcial à obtenção do Certificado de Conclusão.

Orientando(a): Guilherme Gomes Belanda

Orientador(a): Esp. Célio Desiró

Assis/SP

2024

Dados Internacionais de Catalogação na Publicação (CIP)

Belanda, Guilherme Gomes

B426s Sistema de fechamento para levantamento de insumos gastos no setor de bebidas / Guilherme Gomes Belanda. -- Assis, 2024.

34p.

Trabalho de Conclusão de Curso (Graduação em Análise e Desenvolvimento de Sistemas) -- Fundação Educacional do Município de Assis (FEMA), Instituto Municipal de Ensino Superior de Assis (IMESA), 2024.

Orientador: Prof. Esp. Célio Desiró.

1. Automação. 2. Sistemas integrados de gestão empresarial. I Desiró, Célio.
II Título.

CDD 003

SISTEMA DE FECHAMENTO PARA LEVANTAMENTO DE INSUMOS GASTOS NO SETOR DE BEBIDAS

GUILHERME GOMES BELANDA

Trabalho de Conclusão de Curso apresentado ao Instituto Municipal de Ensino Superior de Assis, como requisito do Curso de Graduação, avaliado pela seguinte comissão examinadora:

Orientador: Esp. Célio Desiró

Examinador: Dr. Almir Rogério Camolesi

Assis/SP

2024

AGRADECIMENTOS

Primeiramente, agradeço a Deus, cuja presença e força me sustentaram durante toda essa jornada. Sem sua orientação e sabedoria, não teria sido possível superar os desafios que encontrei pelo caminho.

Confesso que, em muitos momentos, pensei em desistir. Porém, com o apoio incondicional dos meus pais, que sempre acreditaram em mim e me incentivaram a seguir em frente, consegui vencer esta importante etapa da minha vida. Vocês são minha base e inspiração.

Aos meus professores que contribuíram com seu conhecimento ao longo da minha formação, sou imensamente grato. Cada um de vocês deixou uma marca importante na minha trajetória.

Por fim, agradeço aos meus amigos, que estiveram ao meu lado em cada etapa, oferecendo apoio, encorajamento e momentos de descontração que foram essenciais para manter o equilíbrio durante essa fase.

RESUMO

O presente trabalho aborda o desenvolvimento de um aplicativo para realizar o levantamento de insumos gastos em uma produção de bebidas, assim tendo um fechamento com menor tempo gasto e uma proximidade maior dos insumos gastos em uma determinada produção. Ao automatizar o levantamento de insumos, proporcionamos uma maneira mais eficiente de monitorar e controlar estoque, além de facilitar a análise de custos e a tomada de decisões estratégicas. Com a proximidade dos insumos gastos em cada produção, os gestores terão uma visão mais clara dos recursos necessários, permitindo um planejamento mais preciso e uma gestão mais eficaz dos recursos.

Palavras-chave: setor de bebidas; embaladora de latas; indústria de cerveja; sistema de fechamento; insumos gastos.

ABSTRACT

The present work addresses the development of an application to carry out a survey of inputs spent in a beverage production, thus having a closure with less time spent and a greater proximity to the inputs spent in a given production. By automating the collection of inputs, we provide a more efficient way to monitor and control inventory, in addition to facilitating cost analysis and strategic decision-making. With the proximity of inputs spent on each production, managers will have a clearer view of the resources needed, allowing for more precise planning and more effective resource management.

Keywords: beverage sector; can packaging machine; beer industry; closing system; spent inputs.

LISTA DE ILUSTRAÇÕES

Figura 1: Mapa mental.....	21
Figura 2: Casos de uso.....	22
Figura 3: Diagrama de classe.....	27
Figura 4: Diagrama entidade relacionamento.....	28
Figura 5: Método Salvar.....	29
Figura 6: Método Atualizar.....	30
Figura 7: Método deletar.....	31
Figura 8: Operador service.....	32

LISTA DE TABELAS

Tabela 1: Autenticação usuário.....	23
Tabela 2: Cálculo automático.....	24
Tabela 3: Registro de produção.....	25
Tabela 4: Geração de relatórios.....	26

SUMÁRIO

1. INTRODUÇÃO.....	11
1.1 OBJETIVO GERAL.....	12
1.2 OBJETIVO ESPECÍFICO.....	13
1.3 JUSTIFICATIVA.....	13
1.4 MOTIVAÇÃO.....	13
1.5 PERSPECTIVAS DE CONTRIBUIÇÃO.....	14
1.6 METODOLOGIA.....	14
1.7 ESTRUTURA DO TRABALHO.....	14
2. TECNOLOGIAS UTILIZADAS.....	16
2.1 SPRING BOOT.....	16
2.2 ANGULAR.....	16
2.3 POSTGRESQL.....	17
2.4 POSTMAN.....	17
2.5 INTELLIJ IDEA.....	17
2.6 JAVA.....	17
2.7 VISUAL STUDIO CODE.....	18
2.8 LOMBOK.....	18
2.9 MAVEN.....	18
3. MODELAGEM DO SOFTWARE.....	20
3.1 LISTA DE REQUISITOS.....	20
3.2 MAPA MENTAL.....	20
3.3 DIAGRAMA DE CASO DE USO.....	21
3.4 NARRATIVAS DOS UC.....	22
3.4.1 AUTENTICAÇÃO DE USUÁRIO.....	23
3.4.2 CÁLCULO AUTOMÁTICO.....	24
3.4.3 REGISTRO DE PRODUÇÃO.....	25
3.4.4 GERAÇÃO DE RELATÓRIOS.....	26
3.5 DIAGRAMA DE CLASSES.....	27
3.6 DIAGRAMA ENTIDADE RELACIONAMENTO.....	28
4. DESENVOLVIMENTO DO SISTEMA.....	29
4.1 MÉTODO SALVAR.....	29
4.2 MÉTODO ATUALIZAR.....	30
4.3 MÉTODO DELETAR.....	31
4.4 OPERADOR SERVICE.....	32
5. CONSIDERAÇÕES FINAIS.....	33
5.1 TRABALHOS FUTUROS.....	33

1. INTRODUÇÃO

Na indústria um controle de gastos é fundamental para saber onde a empresa está alocando seus gastos e onde possui o maior gasto concentrado, desde a compra do material até o processo final de produção. Com isso, é importante dispor de um sistema para contabilizar os insumos gastos para uma determinada produção de bebidas, desde o processo inicial da produção até o processo final, para que a empresa tenha um levantamento preciso de gastos e quebras de materiais e, assim ter um controle melhor e mais preciso no momento do fechamento.

O presente trabalho documenta o desenvolvimento de um aplicativo para o setor de alimentos e bebidas, com o intuito de realizar o levantamento de insumos gastos em sua produção, independente de qual seja seu segmento. Na maioria dos casos o sistema para contabilizar é feito de forma manual e, o foco deste trabalho é aperfeiçoar o acesso e a entrega das informações de uma forma mais rápida e com um valor mais preciso do gasto real.

Na maior parte das empresas isto é feito de forma escrita e uma pessoa é responsável pela coleta destas informações e reuni-las de forma simplificada apontando os gastos de insumos e quebras durante toda a operação. Além disso, é possível analisar as possíveis soluções com as informações fornecidas com esse aplicativo, em redução de tempo gasto para elaborar relatórios, uma forma melhor de buscar esses dados e a precisão em cima dos gastos.

Conforme destacado por MaqBool e Rafiq (2014), a gestão da cadeia de suprimentos é uma atividade essencial em todos os setores da economia, desde o início, atravessando a indústria até chegar ao comércio. À medida que a indústria fornece uma gama mais ampla e diversificada de produtos, a complexidade dessa tarefa aumenta, tornando-se um diferencial significativo nos dias de hoje.

Segundo apontou Tubino (2007), todos esses desafios estão diretamente relacionados e fazem parte de uma ampla área de estudo conhecida como planejamento e controle de produção.

Muller (2003) destaca que a ampliação e criação de unidades produtivas geralmente estão relacionadas a previsões de longo prazo, enquanto a decisão de ressuprimento de estoque de um insumo está associada à previsão de curto prazo.

Corrêa (2018) destaca que, no âmbito do controle de produção, o estoque figura como um dos conceitos fundamentais. Ele representa a acumulação de materiais em diversos níveis, podendo ser tanto um fator positivo quanto negativo para a empresa. A integração eficiente dessas informações em um sistema pode exercer uma influência positiva significativa no fechamento mensal/anual da empresa, proporcionando uma gestão mais eficaz dos recursos e melhorando o desempenho financeiro.

De acordo com Rosa (2006), a composição da produção engloba o fornecimento de insumos produtivos, como latas e garrafas. Esses materiais desempenham um papel crucial na estratégia de precificação de bebidas, uma vez que possuem um valor agregado significativo ao produto. Especialmente quando se busca a prática de preços mais reduzidos, a consideração cuidadosa desses insumos torna-se fundamental para a eficácia da estratégia de mercado.

O trabalho com ideias mais próximas aqui foi apresentado por Anjos (2021). A autora conclui que ao executar melhorias, a tendência é que o fluxo de informações e parametrizações da empresa, bem como a gestão dos estoques, traga ganhos em sua cadeia de suprimentos. Ela destaca que a padronização de informações de estoque é necessária para o bom desenvolvimento.

1.1 OBJETIVO GERAL

O presente trabalho documenta o desenvolvimento de um aplicativo destinado a uma indústria do setor de bebidas, com foco no fechamento operacional de suas produções. O objetivo é realizar um levantamento preciso de insumos utilizados no processo de envase de cerveja, realizando assim relatórios que reflitam de maneira mais próxima do valor real desses insumos. O aplicativo visa, assim, proporcionar uma ferramenta eficaz para aperfeiçoar o controle e transparência dos insumos envolvidos na produção de bebidas, contribuindo para um controle mais preciso e eficiente.

1.2 OBJETIVO ESPECÍFICO

O objetivo deste aplicativo é permitir que um usuário, especificamente um operador de máquina, faça login e registre a quantidade de produção durante o seu turno. As informações a serem registradas incluem a quebra de filme shrink, a quantidade de pacotes retirados para análise e a quebra de latas. Ao finalizar a produção, o operador responsável realizará a soma total desses itens, efetuando assim um fechamento abrangente de toda a produção. O aplicativo visa simplificar e agilizar o processo de registro, proporcionando uma ferramenta eficaz para o controle detalhado dessas informações.

1.3 JUSTIFICATIVA

Este trabalho se justifica em virtude da grande escassez de aplicativos voltados para o gerenciamento desse tipo de controle, abrangendo empresas de pequeno e grande porte. Adicionalmente, destaca-se a expressiva demanda por ferramentas capazes de efetuar a contabilização detalhada desses insumos, desde a matéria-prima até sua transformação em produto final. Esta carência de soluções eficientes resulta em uma grande quantidade de tempo para ser elaborada, tornando evidente a necessidade premente de uma abordagem inovadora para otimizar e simplificar esse processo.

1.4 MOTIVAÇÃO

A motivação para a realização deste trabalho emerge dos desafios enfrentados diariamente. Assim sendo, o objetivo principal deste estudo consiste em desenvolver um aplicativo que, de maneira fundamental, alcance os objetivos propostos, visando otimizar o tempo e atender às demandas de maneira eficaz e ágil. Este trabalho visa não apenas superar os obstáculos cotidianos, mas também proporcionar uma solução que se destaque pela eficiência e rapidez na resposta às necessidades apresentadas.

1.5 PERSPECTIVAS DE CONTRIBUIÇÃO

No presente trabalho, almeja-se que ele opere de maneira eficiente e cumpra integralmente todas as funcionalidades propostas. Além disso, espera-se que atenda às necessidades da empresa e possa servir como um modelo para outras empresas seja qual for o ramo dela, possibilitando sua adoção e utilização ampla.

1.6 METODOLOGIA

A metodologia de pesquisa deste trabalho consiste na vivência do dia a dia dos processos de envase de bebidas e como funciona a forma atual desse sistema de fechamento. A revisão bibliográfica tem como objetivo apresentar os conceitos teóricos que fundamentam o tema abordado, bem como os trabalhos anteriores que tratam de assuntos relacionados.

Para a realização deste trabalho, será utilizada a linguagem Java, Spring Boot e aplicado o método de Orientação a Objetos. Para a escrita do código, serão adotados alguns princípios, tais como código limpo e arquitetura hexagonal. Por fim, para finalizar e integrar todo o conteúdo, será utilizado o Angular.

1.7 ESTRUTURA DO TRABALHO

O presente trabalho está organizado no seguinte formato. Na seção 1 está a introdução. A seção 1.1 apresenta o objetivo geral deste trabalho. A seção 1.2 apresenta o objetivo específico. A seção 1.3 está a justificativa. A seção 1.4 apresenta a motivação. A seção 1.5 estão as perspectivas deste trabalho. A seção 1.6 apresenta a metodologia. A seção 1.7 apresenta a organização deste trabalho. Na seção 2 estão as tecnologias que serão utilizadas para o desenvolvimento deste trabalho. Na seção 2.1 apresentamos o Spring Boot. Na seção 2.2 apresentamos o Angular. Na seção 2.3 está o PostgreSQL. Na seção 2.4 apresentamos o Postman. Na seção 2.5 está o IntelliJ IDEA. Na seção 2.6 está o Java. Na seção 2.7 apresentamos o Visual Studio Code. Na seção 2.8 o Lombok. E por fim na seção 2.9 o Maven.

Na seção 3 está a modelagem do software, apresentando diversos diagramas de casos de uso. Na seção 3.1 está a lista de requisitos. Na seção 3.2 está o mapa mental. Na seção 3.3 está o Diagrama de Caso de Uso (UC). A seção 3.4 apresenta as narrativas de Caso de Uso. Na seção 3.4.1 apresentamos a Narrativa do Caso de Uso da Autenticação do Usuário. Na seção 3.4.2, a narrativa do Cálculo Automático dos insumos. Na seção 3.4.3, o Registro de Produção. E na seção 3.4.4, a Geração de Relatórios no processo de envase da cerveja. A seção 3.5 é o Diagrama de Classe E na seção 3.6 o Diagrama Entidade Relacionamento. Na seção 4 o desenvolvimento do sistema. A seção 4.1 apresentamos o método Salvar. Na seção 4.2 o método de Atualizar. Na seção 4.3 o método Deletar. E na seção 4.4 o Operador Service.

E na seção 5 as considerações finais deste trabalho. E por fim na seção 6 as conclusões finais, seguido pela seção 6.1 com os trabalhos futuros.

2. TECNOLOGIAS UTILIZADAS

O trabalho atenderá aos objetivos propostos e incluirá uma pesquisa teórica para fundamentá-lo. Será utilizado o método UML para apresentar o projeto, desenvolvendo diagramas de caso de uso, classe e sequência. Na fase de implementação, serão utilizadas tecnologias como Spring Boot, Angular, Postman e IntelliJ IDEA. Para o banco de dados, foi escolhido o PostgreSQL devido à sua robustez e confiabilidade.

2.1 SPRING BOOT

Segundo Santos (2021), Spring boot é um framework desenvolvido na linguagem de programação Java, baseado no padrão de injeção de dependência e bastante leve devido à utilização de POJO (Plain Old Java Objects). Além disso, ele oferece um novo paradigma para o desenvolvimento de aplicativos Spring, com o mínimo de conflitos possíveis e com mais agilidade. Ele é um framework baseado em Java que facilita a criação de aplicações web robustas e escaláveis. Ele simplifica a configuração e o desenvolvimento, oferecendo uma abordagem para configurar o Spring Framework com componentes prontos para uso. Com ele, você pode construir microsserviços e APIs REST rapidamente, com suporte integrado a diversas bibliotecas e bancos de dados.

2.2 ANGULAR

De acordo com Bagliotti e Gibertoni (2020), angular é uma estrutura feita em JavaScript de código aberto, com o propósito de promover produtividade no desenvolvimento, permitindo a utilização da sintaxe HTML para manipular elementos de forma eficiente. Ele foi desenvolvido pela Google, usado para criar interfaces de usuário dinâmicas e responsivas em aplicações web, permitindo a criação de aplicações de página única, utilizando TypeScript para melhorar a organização e a manutenção do código. Angular oferece recursos como two-way data binding, injeção de dependências e roteamento para criar experiências de usuário mais fluidas.

2.3 POSTGRESQL

Segundo Carneiro, Moreira e Freitas (2009), PostgreSQL é um sistema gerenciador de banco de dados. Devido à sua licença aberta, pode ser utilizado, modificado e distribuído por qualquer pessoa para qualquer finalidade, seja privada, comercial ou acadêmica. Ele é altamente escalável e oferece suporte a uma vasta gama de tipos de dados, índices avançados e transações complexas, sendo ideal para aplicações de grande escala que exigem integridade de dados e performance.

2.4 POSTMAN

Carvalho destaca que, Postman pode executar testes automatizados de forma fácil e tranquila, utilizando a ferramenta de execução de coleções, mas ele não possui integração contínua, o que faz toda a diferença quando o assunto é teste automatizado. Ele facilita o desenvolvimento de serviços web ao permitir testes, automação de fluxos de trabalho e a colaboração entre equipes através da criação de coleções de requisições.

2.5 INTELLIJ IDEA

Segundo Cambruzzi (2020), IntelliJ IDEA é uma IDE que oferece uma excelente assistência no ambiente de codificação e características de aumento de produtividade para Java, Spring e outras estruturas, juntamente com ferramentas de implantação para a maioria dos servidores de aplicativos. Ele oferece recursos avançados de refatoração, sugestões de código e ferramentas de depuração, o que agiliza o processo de desenvolvimento e melhora a produtividade do programador.

2.6 JAVA

Mendes (2009) destaca que a linguagem de programação Java representa uma linguagem simples, orientada a objetos, multithread, compilada e interpretada, neutra de arquitetura portátil, robusta, segura e que oferece alto desempenho. Sua

tecnologia é composta por uma linguagem de programa e uma plataforma (API e a máquina virtual Java). Por ser uma linguagem compilada para bytecode, as aplicações em Java podem ser executadas em qualquer plataforma que tenha a JVM, o que torna ideal para o desenvolvimento de aplicações corporativas, móveis e web.

2.7 VISUAL STUDIO CODE

O Visual Studio Code, é um editor de código-fonte desenvolvido pela Microsoft, que ganhou popularidade devido à sua leveza, extensibilidade e recursos avançados. Ele é uma plataforma de desenvolvimento que suporta diversas linguagens de programação, como Java, Python, JavaScript, entre outras. Além de ser multiplataforma, funcionando em Windows, macOS e Linux, o VS Code oferece uma integração rica com Git e outras ferramentas de controle de versão, facilitando o acompanhamento de mudanças no código.

2.8 LOMBOK

Lombok é uma biblioteca Java que tem como principal objetivo reduzir a quantidade de código necessário para escrever classes Java. Ele utiliza anotações que, durante o processo de compilação, geram automaticamente código como getter, setters, construtores e muito mais. Isso torna o código mais conciso, legível e de fácil manutenção. Neste trabalho, a biblioteca Lombok foi utilizada para otimizar a escrita de classes no desenvolvimento da aplicação, tornando o código mais simples e focado na lógica de negócio, ao mesmo tempo em que manteve a estrutura clara e eficiente.

2.9 MAVEN

O Apache Maven é uma ferramenta de gerenciamento e automação de projetos voltada principalmente para projetos Java. Ele simplifica o processo de construção, documentação e dependências de um projeto, utilizando o conceito de Projeto Object Model (POM), onde todas as configurações e dependências são definidas

em um arquivo XML. A partir desse arquivo, o Maven baixa automaticamente as bibliotecas necessárias, garantindo que o projeto tenha todas as dependências corretamente resolvidas e configuradas. No contexto deste trabalho, o Maven foi utilizado para gerenciar as dependências do projeto, como Lombok, Spring Boot e outras bibliotecas, garantindo que o ambiente de desenvolvimento estivesse configurado de maneira eficiente e padronizada.

3. MODELAGEM DO SOFTWARE

Neste capítulo, será apresentada a análise e documentação do software. Será incluída a Documentação de Requisitos de Software, destacando os atores e uma lista de requisitos apresentada por meio de um Mapa Mental. Também serão abordados diversos Diagramas UML e o Modelo Entidade-Relacionamento.

3.1 LISTA DE REQUISITOS

ATORES:

- Operador

MANTER:

- Quebra de filme shrink
- Pacotes p/ análise
- Quebra de latas

MOVIMENTAÇÃO:

- Valores reais dos insumos

3.2 MAPA MENTAL

De acordo com o artigo Alcantara (2020) destaca que os mapas mentais partem de uma ideia central, a partir da qual se articulam as ideias conectadas, numa estrutura em árvore, rede ou em forma de um neurônio.

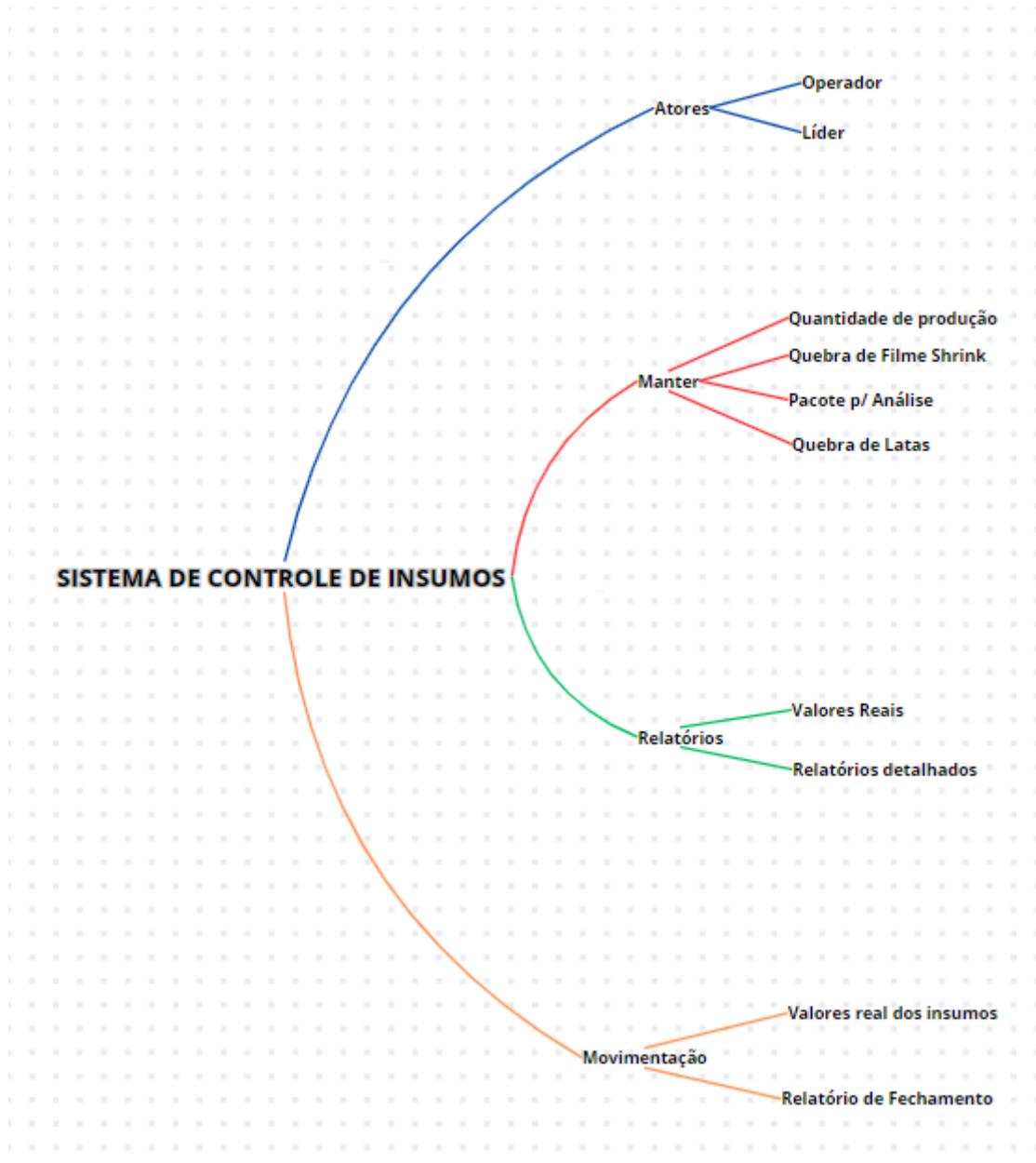


Figura 1: Mapa mental

Fonte: Autoria própria

3.3 DIAGRAMA DE CASO DE USO

De acordo com Barros (2006), o objetivo dos diagramas de caso de uso é identificar as funcionalidades requeridas para o sistema. Os casos de uso fazem parte da fase de elicitação de requisitos, na qual procuramos identificar da melhor forma possível o que nosso sistema deve fazer.

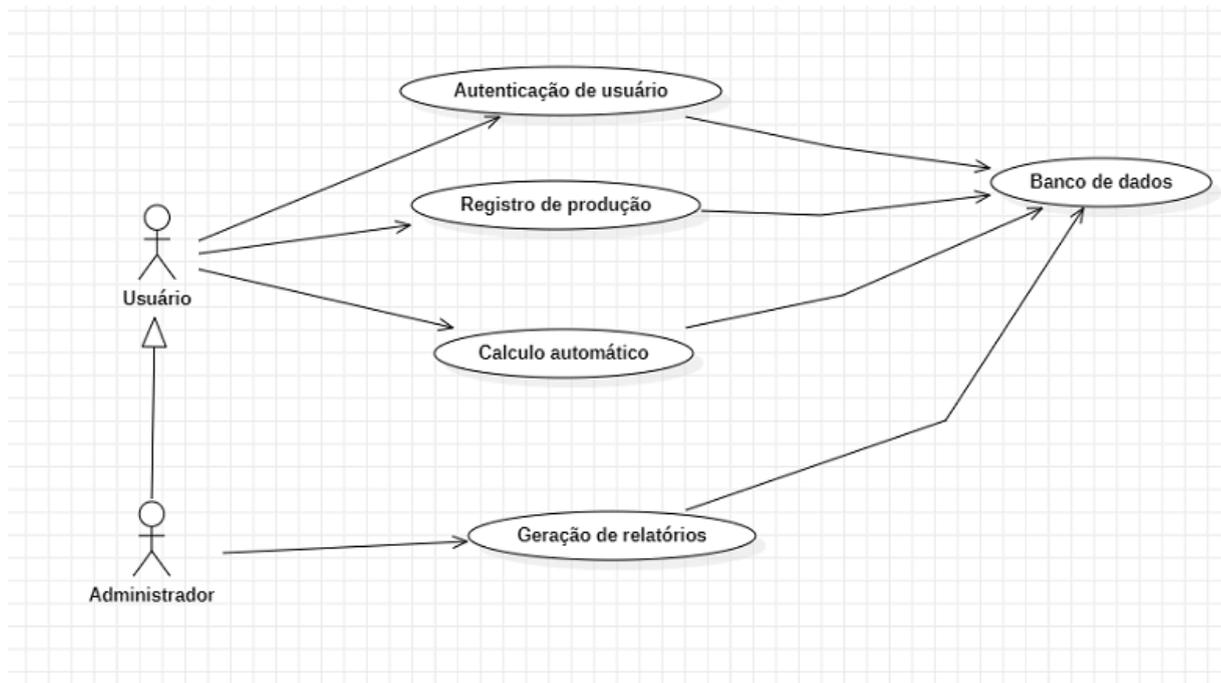


Figura 2: Casos de uso
Fonte: Autoria Própria

3.4 NARRATIVAS DOS UC

As narrativas de UC são conjuntos de passos que um usuário deve seguir para que o sistema alcance um objetivo específico. Isso envolve detalhes sobre entradas, saídas e qualquer comportamento necessário para completar a tarefa desejada.

3.4.1 AUTENTICAÇÃO DE USUÁRIO

Caso de uso	Autenticação de Usuário
Sumário	Permitir que o usuário acesse o sistema fornecendo credenciais válidas.
Ator primário	Colaborador
Ator secundário	
Pré-condições	O sistema está disponível; O usuário possui credenciais válidas.
Fluxo principal	
	<ul style="list-style-type: none"> • O usuário acessa a página de login do sistema. • O usuário insere seu nome de usuário e senha. • O sistema valida as credenciais do usuário. • Se as credenciais forem válidas, o sistema permite o acesso às funcionalidades do sistema. • Caso contrário, o sistema exibe uma mensagem de erro.
Fluxo alternativo	
	<ul style="list-style-type: none"> • Se houver algum erro durante o registro, o operador pode corrigi-lo antes de finalizar o registro.

Tabela 1: Autenticação usuário

Fonte: Autoria própria

3.4.2 CÁLCULO AUTOMÁTICO

Caso de uso	Cálculo Automático
Sumário	Calcular automaticamente a soma total dos itens registrados pelo operador de máquina.
Ator primário	Sistema
Ator secundário	
Pré-condições	O operador de máquina registrou a quantidade de produção e outros itens relacionados.
Fluxo principal	
<ul style="list-style-type: none"> • O sistema recebe as informações registradas pelo operador de máquina. • O sistema realiza o cálculo automático da soma total dos itens registrados. • O sistema exibe o resultado do cálculo para o usuário. 	

Tabela 2: Cálculo automático

Fonte: Autoria própria

3.4.3 REGISTRO DE PRODUÇÃO

Caso de uso	Registro de Produção
Sumário	Permitir que o operador de máquina registre a quantidade de produção durante seu turno.
Ator primário	Operador de Máquina
Ator secundário	
Pré-condições	O operador de máquina está autenticado no sistema.
Fluxo principal	
<ul style="list-style-type: none"> • O operador de máquina acessa a funcionalidade de registro de produção. • O operador registra a quantidade de produção durante seu turno. • O operador registra a quebra de filme shrink, a quantidade de pacotes retirados para análise e a quebra de latas. • O sistema calcula automaticamente a soma total desses itens. 	
Fluxo alternativo	
<ul style="list-style-type: none"> • Se houver algum erro durante o registro, o operador pode corrigi-lo antes de finalizar o registro. 	

Tabela 3: Registro de produção

Fonte: Autoria própria

3.4.4 GERAÇÃO DE RELATÓRIOS

Caso de uso	Geração de Relatórios
Sumário	Gerar relatórios detalhados sobre os insumos utilizados no processo de envase de cerveja.
Ator primário	Administrador do Sistema
Ator secundário	
Pré-condições	O administrador do sistema está autenticado no sistema.
Fluxo principal	
<ul style="list-style-type: none"> • O administrador do sistema acessa a funcionalidade de geração de relatórios. • O administrador seleciona os parâmetros para o relatório, como período de tempo e tipo de insumos. • O sistema gera o relatório com base nos parâmetros selecionados. • O administrador pode visualizar e/ou exportar o relatório gerado. 	
Fluxo alternativo	
<ul style="list-style-type: none"> • Se necessário, o administrador pode ajustar os parâmetros do relatório e regerá-lo conforme necessário. 	

Tabela 4: Geração de relatórios

Fonte: Autoria própria

3.5 DIAGRAMA DE CLASSES

Segundo Almeida (2006), os diagramas de classe são importantes na modelagem, pois mostram a estrutura de um sistema. Eles são considerados os diagramas mais empregados e mais bem compreendidos dentro da UML.

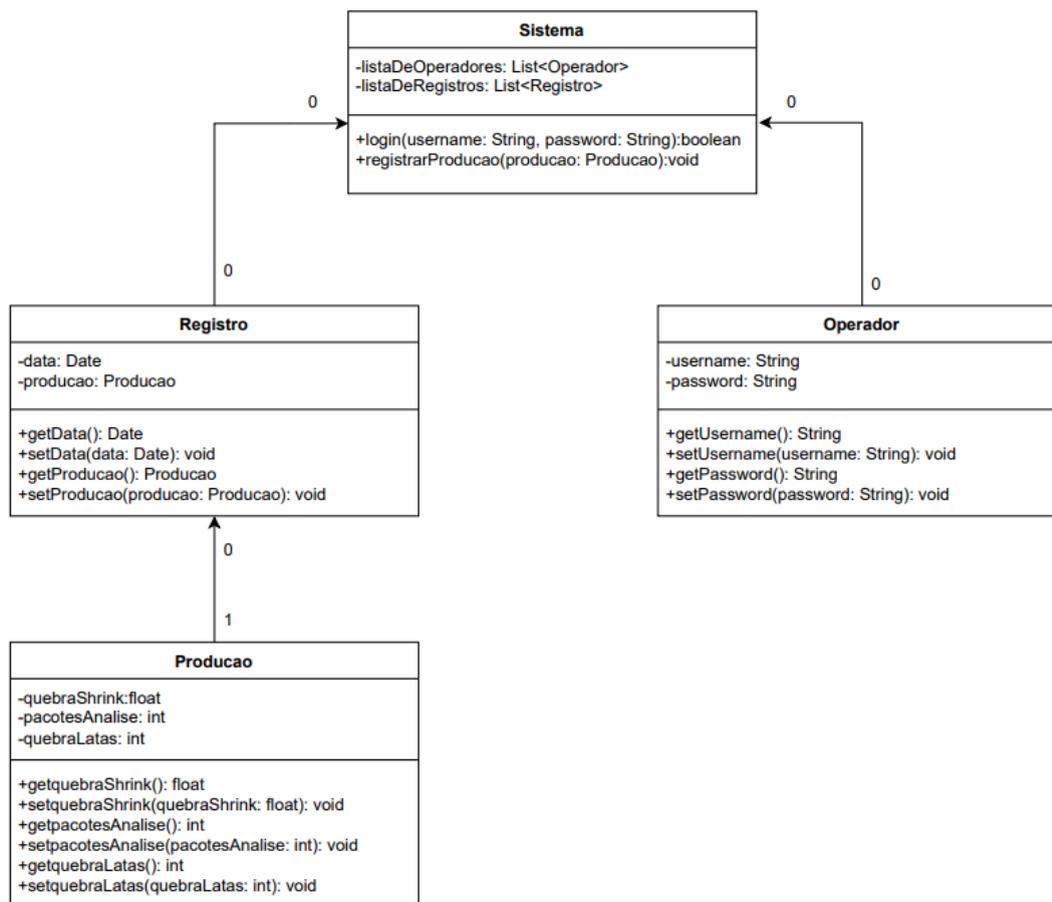


Figura 3: Diagrama de classe
 Fonte: Autoria própria

3.6 DIAGRAMA ENTIDADE RELACIONAMENTO

Aponta Magalhães e Neto (2010), Diagrama Entidade Relacionamento (DER), utiliza elementos gráficos para descrever o modelo de dados de um sistema com alto nível de abstração. O DER parte do princípio de que o mundo real é formado por um conjunto de objetos chamados entidades e por relacionamentos.

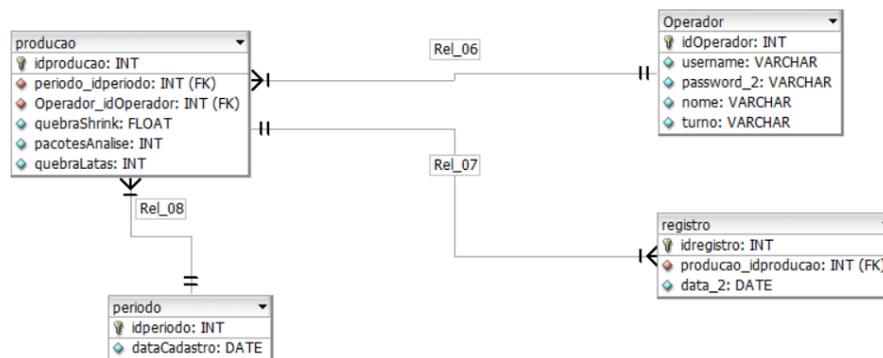


Figura 4: Diagrama entidade relacionamento
Fonte: Autoria própria

4. DESENVOLVIMENTO DO SISTEMA

4.1 MÉTODO SALVAR

```
58     this.service
59         .salvar(this.operador)
60         .subscribe( response => {
61             this.success = true;
62             this.operador = response;
63         }, errorResponse =>{
64             this.errors = errorResponse.error.errors;
65         })
66     }
67 }
```

Figura 5: Método Salvar
Fonte: Autoria própria

Este código faz parte de um componente Angular e está relacionado ao salvamento dos dados de um operador. A função “*salvar*” é chamada através de um serviço que lida com a comunicação HTTP entre o código FrontEnd e BackEnd. O método “*subscribe*” é usado para lidar com a resposta dessa operação.

O método “*salvar*” é chamado com o objeto “*operador*” como argumento. Este objeto contém os dados que devem ser salvos no servidor. A chamada desse método faz uma requisição para o código BackEnd.

A função “*response*” é executada quando a operação de salvamento é bem-sucedida. A propriedade “*success*” é definida como “*true*”, indicando que o salvamento foi realizado.

A função “*errorResponse*” é executada se ocorrer algum erro durante a operação. Neste caso, a propriedade “*errors*” é preenchida com as mensagens de erro retornadas pelo servidor (“*errorResponse.error.errors*”).

4.2 MÉTODO ATUALIZAR

```
onSubmit(){
  if(this.id){
    this.service
      .atualizar(this.operador)
      .subscribe(response => {
        this.success = true;
        this.errors = null;
      }, errorResponse =>
        this.errors = ['Erro ao atualizar'])
  }
}
```

Figura 6: Método Atualizar

Fonte: Autoria própria

Este método é chamado quando o usuário envia o formulário. Ele verifica os dados que vão ser atualizados e, em seguida, chama o serviço para fazer a atualização.

O código verifica se existe um “*id*” associado ao objeto que está sendo manipulado. Se possuir, isso indica que o objeto existe e precisa ser atualizado.

O método “*atualizar*” é chamado, passando o “*operador*” como argumento. Este método faz uma requisição HTTP ao backend para atualizar as informações.

Em caso de sucesso, a propriedade “*success*” é definida como “*true*”, indicando que a atualização foi realizada com sucesso.

Caso ocorra um erro, a propriedade “*errors*” é preenchida com uma mensagem genérica, informando um problema ao atualizar.

4.3 MÉTODO DELETAR

```
deletarOperador(){
  this.service
  .deletar(this.operadorSelecioneado)
  .subscribe(
    response => {
      this.mensagemSucesso = 'Operador Deletado com sucesso!',
      this.ngOnInit();
    },
    erro => this.mensagemErro = 'Ocorreu um erro ao deletar o operador.'
  )
}
```

Figura 7: Método deletar
Fonte: Autoria própria

A função “*deletarOperador()*” é responsável por deletar um operador. Ela chama o serviço “*this.service*” para deletar o operador que está selecionado “*this.operadorSelecioneado*”.

O método “*subscribe*” possui dois parâmetros principais:

- “*response =>*” é executado quando a operação é bem-sucedida.
- “*this.mensagemSucesso = ‘Operador Deletado com sucesso’*” é a mensagem de retorno caso o operador seja deletado.
- “*this.ngOnInit()*” é chamado para atualizar o estado do componente após a deleção.
- “*erro => this.mensagemErro = ‘Ocorreu um erro ao deletar o operador’*” é executado quando ocorre algum erro na deleção. Uma mensagem de erro é atribuída a “*this.mensagemErro*”.

4.4 OPERADOR SERVICE

```
salvar( operador : Operador ) : Observable<Operador> {
    return this.http.post<Operador>(`${this.apiUrl}`, operador);
}

atualizar( operador : Operador ) : Observable<any> {
    return this.http.put<Operador>(`${this.apiUrl}/${operador.id}`, operador);
}

getOperadores() : Observable<Operador[]>{
    return this.http.get<Operador[]>(this.apiUrl);
}

getOperadorById(id : number) : Observable<Operador> {
    return this.http.get<any>(`${this.apiUrl}/${id}`);
}

deletar( operador : Operador ) : Observable<any> {
    return this.http.delete<any>(`${this.apiUrl}/${operador.id}`);
}
```

Figura 8: Operador service

Fonte: Autoria própria

O código acima mostra o serviço que interage com a API para realizar as operações de CRUD (Create, Read, Update, Delete).

O método “*salvar*” tem como objetivo criar um operador. Ele efetua uma requisição POST para a URL da API “*this.apiUrl*” com os dados do operador a ser criado.

O método “*atualizar*” tem como objetivo atualizar um operador existente. Ele efetua uma requisição PUT para a URL da API, verificando o ID do operador a ser atualizado “*operador.id*”.

O método “*getOperadores()*” traz uma lista completa de todos os operadores criados. Ele realiza uma requisição GET, retornando uma lista de operadores como resposta.

O método “*deletar*” deleta um operador. Ele realiza uma requisição DELETE, informando o ID do operador a ser deletado.

5. CONSIDERAÇÕES FINAIS

Houve muitas dificuldades no processo de desenvolvimento deste trabalho, pois precisei me aperfeiçoar em diversos assuntos nos quais não tinha muito conhecimento. Entretanto, a maioria dos objetivos foi alcançada graças a pesquisas e cursos na internet, além do apoio dos professores e colegas de sala, que colaboraram grandemente para a conclusão deste projeto.

Minha maior dificuldade foi desenvolver o FrontEnd e o Back-End, pois era algo em que eu não tinha grande conhecimento. No entanto, com o material das aulas dos meus professores e cursos encontrados na internet, consegui realizar o desenvolvimento.

Espero que este projeto ajude novos desenvolvedores e pessoas que se interessam pelo assunto e tema a desenvolver novos trabalhos. Com esta finalização, espera-se que este trabalho tenha algum valor para a comunidade.

5.1 TRABALHOS FUTUROS

Com o desenvolvimento deste trabalho, é possível expandi-lo para abranger o fechamento de todas as máquinas que compõem uma linha completa de produção. Dessa forma, todos os dados podem ser inseridos, resultando em uma lista que cobre toda a linha de produção. Cada máquina será detalhada com seus respectivos valores, proporcionando uma visão completa e precisa do processo. Com uma visão detalhada de cada etapa, será possível realizar análises mais precisas e implementar melhorias contínuas no sistema produtivo.

REFERÊNCIAS

ALCÂNTARA, Elisa Ferreira Silva de. (2020). Mapa Conceitual e Mapa Mental. Anais do VIII Simpósio de Pesquisa e de Práticas Pedagógicas.

ALMEIDA, Vinicius Coelho de. Uso da linguagem OCL no contexto de diagramas de classe da UML e programas em Java. 75 Pág. Dissertação (Mestre). Departamento - Universidade Federal de Minas Gerais. Minas Gerais, Belo Horizonte, 2006.

ANJOS, Luana Agnes dos. ANÁLISE DO MRP COMO UMA METODOLOGIA DE PLANEJAMENTO E CONTROLE DE ESTOQUE EM UMA INDÚSTRIA DE BEBIDAS DE MÉDIO PORTE. 2021. 32 Pág. Trabalho de conclusão de curso (Bacharel). Departamento – Universidade Tecnológica Federal do Paraná. Paraná, Londrina, 2021.

BAGLIOTTI, I. R.; GIBERTONI, D. REUSABILIDADE NO DESENVOLVIMENTO DE UM SISTEMA WEB UTILIZANDO O FRAMEWORK ANGULAR. Revista Interface Tecnológica, [S. l.], v. 17, n. 1, p. 192–204, 2020. DOI: 10.31510/infa.v17i1.826. Disponível em: <https://revista.fatectq.edu.br/interfacetecnologica/article/view/826>. Acesso em: 29 mar. 2024.

BARROS, João Paulo. Casos de uso e respectivos diagramas. Escola Superior de Tecnologia e Gestão Instituto Politécnico de Beja. Beja, Portugal. 2009.

CAMBRUZZI, Wesley Patrick. Aplicativo mobile para gerenciamento de reembolso de quilometragem. 2020. 39 Pág. Monografia de Especialização. Departamento - Universidade Tecnológica Federal do Paraná. Paraná, Pato Branco. 2020.

CARNEIRO, Alessandro Pinto; MOREIRA, Juliano Lucas; FREITAS, André Luís Castro de. TUNING-técnicas de otimização de banco de dados um estudo comparativo: mysql e postgresql. Universidade Federal do Rio Grande (FURG). RS, Rio Grande. 2009.

CARVALHO, Muriel Magno Teles de. A aplicação de teste de regressão na qualidade de software usando Postman e Newman. 2022. 55 Pág. Trabalho de Conclusão de Curso (Bacharel). Departamento - Centro Universitário Unidade de Ensino Superior Dom Bosco. Maranhão, São Luís. 2022.

GANEM, Alan Motta. ALGORITMOS DE APRENDIZADO DE MÁQUINAS APLICADOS NO DIMENSIONAMENTO E CONTROLE DE ESTOQUE NA INDÚSTRIA DE BEBIDAS. 2020. 76 Pág. Trabalho de Graduação. Departamento – Universidade Federal de São Carlos. São Paulo, São Carlos, 2020.

MAGALHÃES, Rafael L.; NETO, Michelle MF. ApreNDER: Ferramenta de apoio à construção de diagrama entidade relacionamento para deficientes visuais. Departamento - Instituto Federal Fluminense (IFF). Rio de Janeiro, Itaperuna, 2010.

MENDES, Douglas Rocha. Programação Java com Ênfase em Orientação a Objetos. 2009. NovaTec Editora LTDA. SP, São Paulo. 2009.

SANTOS, Adailson Pinho dos. JCodingTime: Gerador de Testes Unitários para Aplicações em Spring Boot Desenvolvidas pelo JHipster. 2021. 114 Pág. Trabalho de Conclusão de Curso (Bacharel). Departamento - Universidade de Brasília - UnB. Distrito Federal, Brasília, 2021.