

Almorganzado – Sistema para Controle e Gestão de Almoxarifados

Peterson Bugatti Costa Faria, Daniela Marques

Tecnólogo em Análise e Desenvolvimento de Sistemas - Instituto Federal de Educação, Ciência e Tecnologia de São Paulo, Campus Hortolândia (IFSP) Avenida Thereza Ana Cecon Breda, s/n, Vila São Pedro - Hortolândia – SP – Brasil,

peterson.b@aluno.ifsp.edu.br, marquesdaniela@ifsp.edu.br

Abstract. *Due to the adversities found in the warehouse sector of companies, which are reported in scientific articles and related works, the present work consists of developing a prototype of an app to help in the stock control of warehouses. As a result, it is expected that the study and development of this project will provide a product that can collaborate with the management of some services performed by the warehouses, thereby easing the problems of disorganization, loss of goods and others.*

Resumo. *Devido as adversidades encontradas no setor de almoxarifado das empresas, que são relatados em artigos científicos e trabalhos correlatos, o presente trabalho consiste em desenvolver um protótipo de um aplicativo para ajudar no controle de estoque dos almoxarifados. Como resultado espera-se que o estudo e o desenvolvimento desse projeto forneçam um produto que consiga colaborar com o gerenciamento de alguns serviços exercidos pelos almoxarifados, com isso amenizando os problemas de desorganização, perdas de bens e outros.*

1. Introdução

Segundo a Agência Brasil, no ano de 2021 dados estatísticos apontam um crescimento de cerca de 3,2% de empresas ativas no país, chegando num valor aproximado de 19.907.733 firmas abertas, onde 85% desse valor é representado por empresas micro e de pequeno porte e 15% delas por empresas de médio a grande porte segundo dados retirados do JusBrasil.

Almoxarifado é um setor de uma empresa e é nesse local onde o volume de materiais adquiridos para suprir as necessidades operacionais dos demais setores da organização são armazenados e assegurados. No entanto, há a possibilidade do surgimento de divergências no controle desses itens, justamente por conta das demandas de serviços requeridos e pelos erros de gerenciamento do local e dos respectivos materiais (MACHADO, 2011, p.11).

Baseando-se em algumas pesquisas realizadas de artigos como “Problemas relacionados com a falta de material em almoxarifado da administração pública” (MACHADO, 2011) e/ou “Proposta de melhoria contínua em um almoxarifado utilizando a ferramenta 5W2H” (GROSELLI, 2014) e notícias sobre o assunto, o número de casos de falta de bens e desorganização neste ambiente de trabalho ainda é um obstáculo para as companhias e isso é prejudicial à elas, porque ao perder materiais e/ou itens consequentemente ela tem prejuízo financeiro (MACHADO, 2011, p.11). Esses problemas podem ocorrer por diversos motivos e concomitante a isso, não podemos deixar de mencionar a importância do gerenciamento do almoxarifado, pois ele exerce uma função significativa para o sucesso da cadeia de suprimentos de qualquer empresa (RODRIGUES, 2018, p.2).

Com base no cenário descrito, o objetivo do presente trabalho consiste em desenvolver um protótipo de um sistema para almoxarifados, desfrutando de ferramentas *Open-Source* para a sua construção.

Esse artigo irá apresentar na Seção 2 um conjunto de trabalhos correlatos ao tema estudado, com o intuito de conhecer outros sistemas semelhantes para a realização de uma analogia. Na Seção 3 são apresentados os principais termos e conceitos abordados durante o artigo para uma maior compreensão dos leitores. A Seção 4 vai descrever qual será a metodologia usada para o desenvolvimento do trabalho e na sequência temos a Seção 5 que mostra todas as etapas do modelo cascata que foram base para o desenvolvimento do protótipo. Por fim, a Seção 6 contém as considerações finais desse trabalho assim como a abordagem das melhorias e *upgrades* futuros.

2. Trabalhos correlatos

Esta Seção apresenta sistemas correlatos e comparações entre eles.

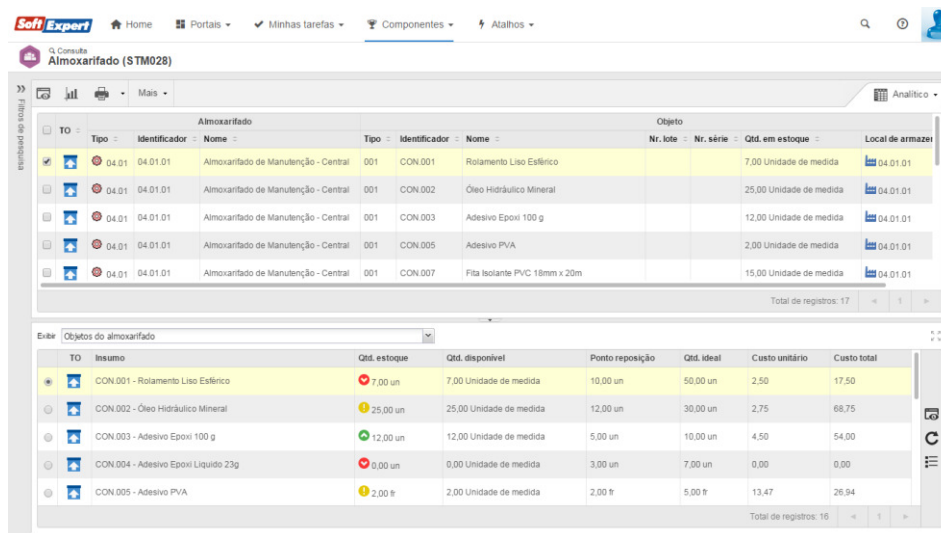
Ao procurar por sistemas de almoxarifado na internet, é possível encontrar vários sistemas disponíveis para utilização em uma empresa, desde programas gratuitos até aqueles que são pagos.

Com o intuito de fazer uma análise e trazer informações para a pesquisa, selecionamos 4 sistemas aleatórios encontrados na internet para compará-los entre si e através disso mapear quais são as funcionalidades comuns.

2.1. SoftExpert Almoxarifado

O primeiro escolhido foi o “SoftExpert Almoxarifado”, um sistema de propriedade da (SoftExpert, 2021) focado no gerenciamento e controle de estoques, almoxarifados e inventários. Ele possui algumas funcionalidades para quem almeja fazer a administração dos serviços realizados em um almoxarifado, para exemplificar temos o histórico de eventos de entrada e saída de produtos, alertas para quando a quantidade estiver abaixo do especificado, geração de relatórios e outros.

A Figura 1 mostra a interface do sistema que contém informações dos materiais cadastrados nos determinados almoxarifados, também possui dados da quantidade de estoque e quantidade disponível desse produto e quais os custos dele. Um ponto muito interessante dessa tabela é que ela, além de controlar todos os aspectos mencionados acima, também tem a funcionalidade de pontuar os produtos que necessitam de reposição, quantos são necessários para repor e obter o ideal e quanto custaria para realizar essa ação.



The screenshot shows the 'Almoxarifado (STM028)' interface. It features a table with columns for 'Almoxarifado' (Tipo, Identificador, Nome) and 'Objeto' (Tipo, Identificador, Nome, Nr. lote, Nr. série, Qtd. em estoque, Local de armazem). Below this is a summary table 'Objetos do almoxarifado' with columns: TO, Insumo, Qtd. estoque, Qtd. disponível, Ponto reposição, Qtd. ideal, Custo unitário, and Custo total.

TO	Insumo	Qtd. estoque	Qtd. disponível	Ponto reposição	Qtd. ideal	Custo unitário	Custo total
CON.001	Rolamento Liso Esférico	7,00 un	7,00 Unidade de medida	10,00 un	50,00 un	2,50	17,50
CON.002	Óleo Hidráulico Mineral	25,00 un	25,00 Unidade de medida	12,00 un	30,00 un	2,25	68,75
CON.003	Adesivo Epoxi 100 g	12,00 un	12,00 Unidade de medida	5,00 un	10,00 un	4,50	54,00
CON.004	Adesivo Epoxi Líquido 23g	0,00 un	0,00 Unidade de medida	3,00 un	7,00 un	0,00	0,00
CON.005	Adesivo PVA	2,00 un	2,00 Unidade de medida	2,00 un	5,00 un	13,47	26,94

Figura 1. Controle de Materiais

Fonte: [SoftExpert, 2021]

2.2. CPT Almoarifado

O “CPT Almoarifado”, pensado e criado pela (CPT Softwares, 2021), tem por função realizar o gerenciamento de almoxarifados que segundo a empresa possui características modernas, ágeis e seguras. O CPT apresenta funcionalidades para gerenciamento e controle dos dados do setor como fornecedores, produtos, lotes, usuários, funcionários, como por exemplo temos o cadastro de produtos na Figura 2. Também faz o controle do próprio estoque com informações de entradas, perdas, consumo e código de barras, além de realizar vários alertas para casos de vencimentos, empréstimos e manutenções.

Figura 2. Cadastro de Produto

Fonte: [CPT Softwares, 2021]

2.3. GestãoClick

O terceiro sistema selecionado para estudo e pesquisa é o (GestãoClick, 2021), desenvolvido especificamente para controle de almoxarifados. O GestãoClick possui mais de 30 funcionalidades diferentes, além de ser uma espécie de ERP *online*. Dentre as diversas funcionalidades existentes podemos citar: Cadastros, Controle de estoque, Controle financeiro, Fluxo de caixa, Emissão de relatórios, Emissão de Notas Fiscais, Gestão de contratos, Agenda *online* e muitos outros.

A Figura 3 mostra um resumo geral da aplicação e algumas das visualizações que ela possui, como gráficos, *dashboards* e outros.

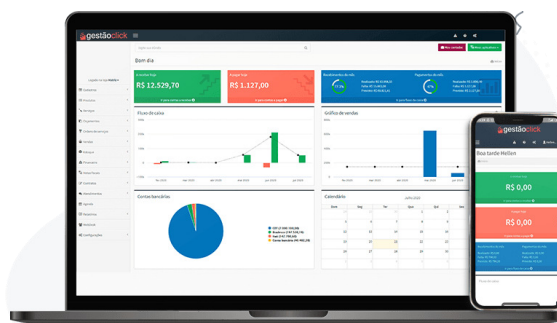


Figura 3. GestãoClick

Fonte: [GestãoClick, 2021]

2.4. Sistema de Almoxarifado-DQL

Por último, temos o sistema de Almoxarifado-DQL (Figura 4), o qual foi desenvolvido pela empresa (DQL Sistemas, 2021) e assim como os outros também tem por objetivo facilitar e simplificar o controle de entrada e saída dos itens de um almoxarifado. O sistema é todo controlado por senhas para garantir a segurança da informação, além de entregar serventias ao cliente, desde cadastro de pessoas (consumidores, fabricantes, representantes e fornecedores), departamentos e outros, até mesmo com o auxílio da emissão de pedidos para saída de produtos, geração de relatórios e gráficos e informações do estoque.

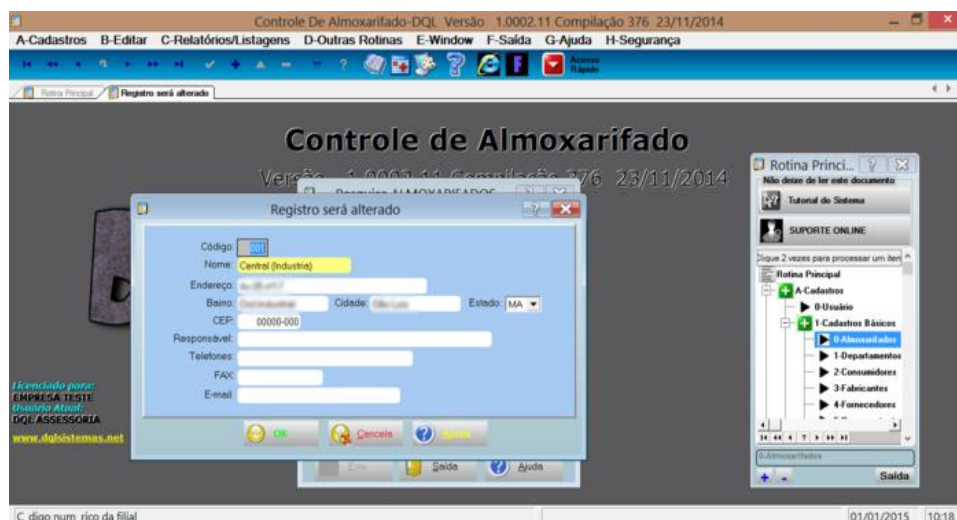


Figura 4. Tela principal e secundárias

Fonte: [Download.it, 2021]

2.5. Análise das funcionalidades

Baseando-se nos 4 sistemas analisados e estudados, foram elencadas 10 funcionalidades no Quadro 1, a fim de relacionar e identificar quais são comumente utilizadas.

Quadro 1. Quadro das funcionalidades comuns entre os sistemas

Funcionalidades	SoftExpert Almoxarifado	CPT Almoxarifado	GestãoClick	Almoxarifado-DQL
Cadastro Cliente	X	X	X	X
Cadastro Fornecedor	X	X	X	X
Cadastro Departamento	X	X		X
Cadastro Produto	X	X	X	X
Cadastro Transportadora	X			
Relatório de entrada/saída de produtos	X	X	X	X
Controle de Estoque	X	X	X	X
Ordens/Pedidos	X		X	X
Agenda de compromissos		X	X	
Dados financeiros	X	X	X	

Fonte: Elaborada pelo autor

O Quadro 1 mostra que o Cadastro de Cliente, Cadastro de Fornecedor, Cadastro do Produto, Controle de Estoque e o Relatório, estão presentes em todos os sistemas desta pesquisa, além disso também é notória a participação de diferentes tipos de usuários desses sistemas, onde cada um possui determinadas funcionalidades para o seu perfil.

3. Referencial teórico

Para maior compreensão do artigo, esta Seção tem o propósito de descrever de maneira resumida os conceitos e significados dos termos empregados no projeto.

3.1. Almoxarifado

É local/espço reservado nas empresas que tem por função armazenar e conservar materiais, itens, produtos, equipamentos e/ou qualquer outro objeto necessário que seja guardado. Além disso, nesse setor deve-se assegurar a qualidade e quantidade exata do material, realizar o gerenciamento de inventário/estoque dos itens, evitar perdas de qualquer natureza e manter as instalações e recursos corretos para o deslocamento e distribuição desses materiais de forma ágil e eficaz (UFPI, 2013, p.2).

3.2. Modelo Cascata

O Modelo Cascata é um modelo de desenvolvimento de *software* sequencial que representa as atividades fundamentais do processo, como especificação, desenvolvimento, validação e evolução, na forma de fases de processo distintas, como a especificação de requisitos sejam eles funcionais ou não-funcionais, projeto de *software*, implementação e testes. (SOMMERVILLE, 2019, p.31-33).

- Análise e definição dos requisitos: essa é a fase que determina os requisitos do software, normalmente são os objetivos e funcionalidades que ele irá possuir, além de realizar o estudo de viabilidade e documentar tudo que for discutido.
- Projeto do sistema e do *software*: Essa é a fase que antecede a codificação, logo essa etapa tem por objetivo analisar os requisitos e projetar os processos em quatro atributos: a estrutura de dados, arquitetura de *software*, caracterização das interfaces e detalhes procedimentais.
- Implementação: Etapa na qual é realizado a programação do sistema, ou seja, quando os programas são criados (codificação), no entanto ainda nessa etapa já é realizado os testes unitários de cada pedaço do código desenvolvido.
- Integração e Teste do sistema: Essa fase onde são realizados os testes globais do sistema, com o intuito de averiguar as lógicas internas e as funcionalidades do sistema. Essa etapa é essencial, pois ela evidencia os erros do comportamento do *software* que podem ser corrigidos antes de ir para uso.
- Operação e manutenção: Após subir o sistema em ambiente de produção, serão corrigidos os erros encontrados que não foram antes achados nos testes, dessa forma é possível melhorar o sistema com o suporte do mesmo.

3.3. Protótipo

Normalmente ao se utilizar o modelo de desenvolvimento cascata é gerado uma primeira versão de um sistema que pode ser melhorado no futuro, esse tipo de sistema pode ser identificado como um protótipo, que nada mais é do que versões simplificadas de um *software*. Eles são desenvolvidos de forma rápida e normalmente como a primeira versão de

uma aplicação que implementa algumas funcionalidades e características, no entanto a ideia é que ele sofra alterações de melhorias solicitadas pelo usuário, como os requisitos da interface, por exemplo (SOARES, 2007, p. 1 e 2).

3.4. Requisitos Funcionais e Não-Funcionais

Já na primeira etapa do modelo cascata é levantando e analisado os requisitos do sistema em questão, no entanto existem dois diferentes tipos de requisitos, podendo ser do tipo funcional ou não-funcional. Os requisitos funcionais são aqueles que determinam as funcionalidades e/ou os serviços de um sistema ou até também o que o sistema não deve fazer, além de definir como o sistema deve reagir a entradas específicas e como o sistema deve se comportar em determinadas situações. (SOMMERVILLE, 2007, p.94). Os requisitos não-funcionais por sua vez, definem como o sistema realizará determinada ação, no entanto não estão diretamente relacionados às funcionalidades de um sistema, mas sim pensando no quesito qualidade. Podem ser restrições sobre os serviços ou as funções oferecidas pelo sistema, como, por exemplo, a segurança, disponibilidade, robustez e outros (SOMMERVILLE, 2007, p.94).

3.5. Diagrama de Caso de Uso

Ainda pensando no modelo cascata, foi dito que na primeira etapa do cascata é importante documentar os requisitos levantados para o sistema e uma das formas de fazer isso, é utilizar diagramas. O diagrama de caso de uso é provindo do *Unified Modeling Language* (UML) e tem o intuito de descrever as funcionalidades dos sistemas orientado a objetos.

Um caso de uso desenha um cenário para o levantamento de requisitos, além de identificar o tipo da interação e os atores envolvidos (SOMMERVILLE, 2007, p.116).

Esse diagrama possui 4 diferentes representações, sendo elas (RIBEIRO, 2012):

- Cenário: Sequência de eventos da interação do usuário com o sistema.
- Ator: Usuário do sistema, ou melhor, um tipo de usuário.
- Caso de uso: É um requisito ou uma funcionalidade realizada pelo ator.
- Comunicação: é a interação que liga um ator com um caso de uso.

3.6. Banco de Dados

Durante o desenvolvimento, será necessário o armazenamento dos dados que serão utilizados no sistema, para isso na área de tecnologia da informação, existe os bancos de dados que são um conjunto de informações e/ou dados que são armazenados de forma organizada e estruturada em um sistema de gerenciamento de banco de dados (SGBD) e que se relacionam entre si com o propósito de criar algum sentido (Oracle, 2021).

3.7. MVC – Modelo, Visualização e Controle

Pensando ainda no desenvolvimento do sistema, é trivial pensar em como todo o código do sistema será organizado, sendo assim foi pensado em utilizar o MVC que em resumo se trata de um padrão de arquitetura de *software* que surgiu na década de 80 que apoia a apresentação dos dados de maneiras diferentes e separa as interações em 3 partes, que são o “Modelo” que possui a função de modelar os dados e o seu comportamento, a “Visualização” que é a parte de interface com o usuário e toda a parte dos elementos de visualização e o “Controlador” que é a parte intermediária entre as duas outras partes, por isso ela controla e mapeia as ações das demais classes. Essa separação surgiu com a ideia de facilitar o gerenciamento do código,

reduzindo o acoplamento e gerando maior entendimento das classes, além de simplificar muito a troca de informações entre as interfaces e o banco de dados (SOMMERVILLE, 2007, p.297).

3.8. Testes

Por fim, durante e ao finalizar a criação dos programas é essencial realizar os testes para a validação das funcionalidades que eles exercem. O teste é uma etapa do modelo cascata, pois é nesse momento onde serão realizados os testes do *software* com o intuito de verificar se o que foi desenvolvido está de fato funcionando conforme deveria. “Quando são realizados testes adequados e dentro das melhores práticas, contribui-se também para a melhoria da qualidade e redução dos custos dos produtos finais, pela diminuição do retrabalho nos estágios posteriores de testes, durante a implantação e na manutenção.” (RIOS, E; MOREIRA, T., 2013).

4. Metodologia

Para a elaboração desse trabalho, esta pesquisa baseou-se no estudo e análise dos sistemas correlatos abordados na Seção 2 com o intuito de reconhecer quais as principais funcionalidades integradas nesses sistemas. O Quadro 1 apresenta as funcionalidades comuns entre os sistemas e algumas delas servem como base para a definição dos requisitos deste trabalho.

O modelo cascata foi utilizado como modelo de desenvolvimento, porque pensando no contexto protótipo e quando os processos já estão bem definidos, esse modelo de desenvolvimento se encaixa muito bem. O primeiro estágio foi a definição de requisitos que foi feita com base nos estudos dos sistemas correlatos conforme explicado anteriormente. O diagrama de caso de uso foi utilizado para documentar os requisitos do sistema. Posteriormente foi feita a abstração dos requisitos a nível de análise seguido pela implementação e testes.

À medida que as funcionalidades forem desenvolvidas foram integradas ao sistema. O estágio de implementação e testes foi desenvolvido por caso de uso sequencialmente uma vez que só há um desenvolvedor para o sistema.

5. Desenvolvimento

A seguir será apresentado o desenvolvimento do sistema divididos em fases de acordo com o modelo cascata.

5.1. Análise e definição dos requisitos

Partindo dos sistemas correlatos na Seção 2 e toda a análise realizada, foi criado e modelado o diagrama de caso de uso que demonstra as funcionalidades e os atores envolvidos no sistema (Figura 5).

O ator Administrador é o que tem permissão total no sistema, interage com os casos de uso Efetuar Login, Movimentar Estoque que pode ser estendido para o caso de uso Gerar Relatório que tem por função emitir o histórico de ações realizadas no estoque (entrada e saída de produtos), além do caso de uso Manter Produto e Manter Usuário, que fazem as operações básicas tanto para os usuários quanto para os produtos, como inserir, alterar e excluir.

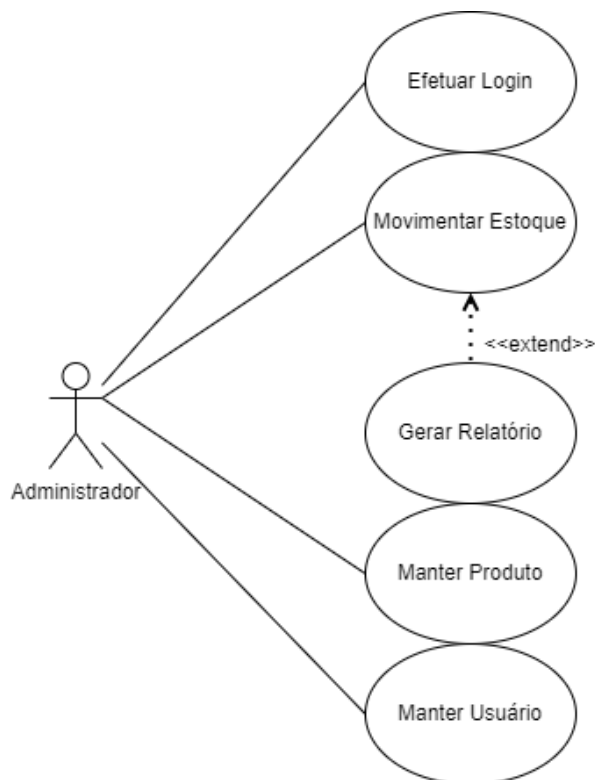


Figura 5. Diagrama de Caso de Uso

Fonte: Elaborada pelo autor.

5.2. Projeto do sistema e do software

5.2.1. Prototipação em papel

Nesta Seção é apresentada a prototipação do sistema, conforme explicado anteriormente, o protótipo deve ser feito de forma rápida e simples. O protótipo foi desenhado em papel como forma de demonstrar o funcionamento total do sistema.

A Figura 6 apresenta o protótipo proposto a primeiro momento para o caso de uso Manter Produto englobando todas as operações possíveis.

Figura 6. Tela Manter Produto

Fonte: Elaborada pelo autor.

Outra tela que foi pensada e desenhada é a tela que representa o caso de “Manter Usuário”, basicamente são os funcionários da empresa que trabalham no setor de almoxarifado. A Figura 7 mostra o *design* inicial pretendido.

ALMORGANIZADO - □ X

CADASTRO FUNCIONÁRIO

NOME: CPF:
 TELEFONE: DATA NASCIMENTO:
 E-MAIL: CEP:
 SEXO: MASCULINO ENDEREÇO:
 FEMININO

ID	NOME	TELEFONE	E-MAIL	SEXO	CPF	DATA NASC.	CEP	ENDEREÇO

USUÁRIO: ADMINISTRADOR

Figura 7. Tela do Layout dos cadastros

Fonte: Elaborada pelo autor.

5.2.2. Modelagem do banco de dados

Após a definição das funcionalidades do sistema, um modelo de banco de dados em linguagem SQL foi construído com suas respectivas tabelas, tendo em vista que para o sistema será necessário o armazenamento dos dados de usuários, produtos e movimentações do estoque. A Figura 9 demonstra parte do código na linguagem SQL e como está sendo feito a parametrização dos dados e criação das tabelas.

Em adendo, a Figura 8 mostra o Diagrama de Banco de Dados gerado automaticamente no MySQL WorkBench a partir do código do banco.

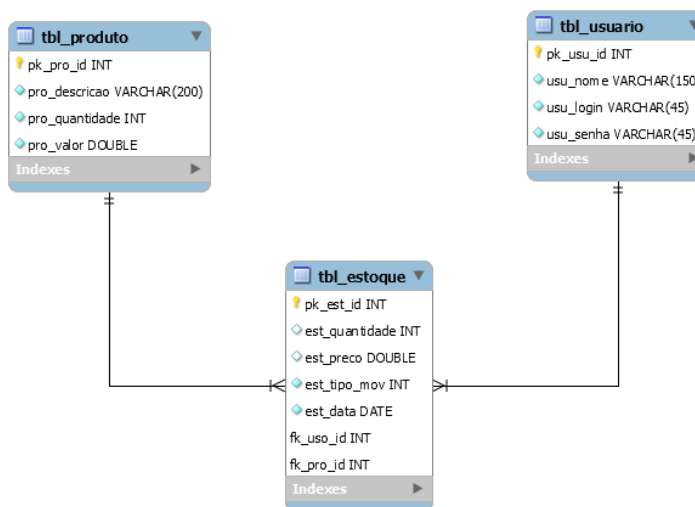


Figura 8. Diagrama de Banco de Dados

Fonte: Elaborada pelo autor.

```

CREATE SCHEMA IF NOT EXISTS `dbestoque` DEFAULT CHARACTER SET utf8 ;
USE `dbestoque` ;

CREATE TABLE IF NOT EXISTS `dbestoque`.`tbl_produto` (
  `pk_pro_id` INT NOT NULL AUTO_INCREMENT,
  `pro_descricao` VARCHAR(200) NOT NULL,
  `pro_quantidade` INT NOT NULL,
  `pro_valor` DOUBLE NOT NULL,
  PRIMARY KEY (`pk_pro_id`))
ENGINE = InnoDB;

CREATE TABLE IF NOT EXISTS `dbestoque`.`tbl_usuario` (
  `pk_usu_id` INT NOT NULL AUTO_INCREMENT,
  `usu_nome` VARCHAR(150) NOT NULL,
  `usu_login` VARCHAR(45) NOT NULL,
  `usu_senha` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`pk_usu_id`))
ENGINE = InnoDB;

CREATE TABLE IF NOT EXISTS `dbestoque`.`tbl_estoque` (
  `pk_est_id` INT NOT NULL AUTO_INCREMENT,
  `est_quantidade` INT NULL,
  `est_preco` DOUBLE NULL,
  `fk_uso_id` INT NOT NULL,
  `fk_pro_id` INT NOT NULL,
  PRIMARY KEY (`pk_est_id`, `fk_uso_id`, `fk_pro_id`),
  INDEX `fk_tbl_estoque_tbl_usuario_idx` (`fk_uso_id` ASC) ,
  INDEX `fk_tbl_estoque_tbl_produto1_idx` (`fk_pro_id` ASC) ,
  CONSTRAINT `fk_tbl_estoque_tbl_usuario`
  FOREIGN KEY (`fk_uso_id`)

```

Figura 9. Código referente a criação do banco de dados

Fonte: Elaborada pelo autor.

5.3. Implementação

Dando sequência para o desenvolvimento do sistema, partimos do pressuposto de estruturar as classes e pacotes do projeto utilizando MVC (Model, View e Controller), a Figura 10 apresenta essa organização. Outra utilização importante, é o pacote DAO o qual foi criado para armazenar todas as classes responsáveis pelo encapsulamento do acesso com o banco de dados, permitindo que outras camadas da aplicação consigam se comunicar com o banco.

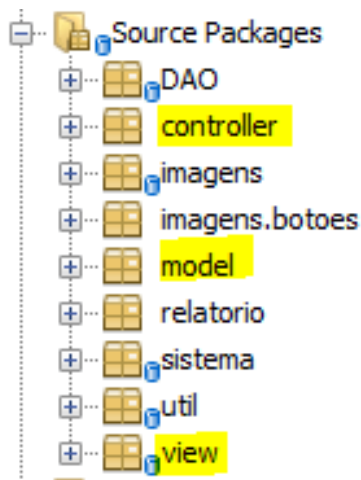


Figura 10. Arquitetura de software MVC

Fonte: Elaborada pelo autor.

Ainda neste estágio, foram construídas as telas do sistema por meio da tecnologia Swing utilizando a ferramenta Apache NetBeans, as classes que representam essas interfaces estão presentes no pacote de View (visualização) conforme o padrão MVC.

A primeira funcionalidade implementada no sistema foi o caso de uso Efetuar Login (Figura 11) para autenticar o usuário e permitir que somente usuários com cadastros tenham acesso ao sistema.



Figura 11. Tela de Login

Fonte: Elaborada pelo autor.

Após realizada a autenticação e o acesso do usuário no sistema, o mesmo é direcionado para um menu inicial (Figura 12) que contém uma paleta de opções.

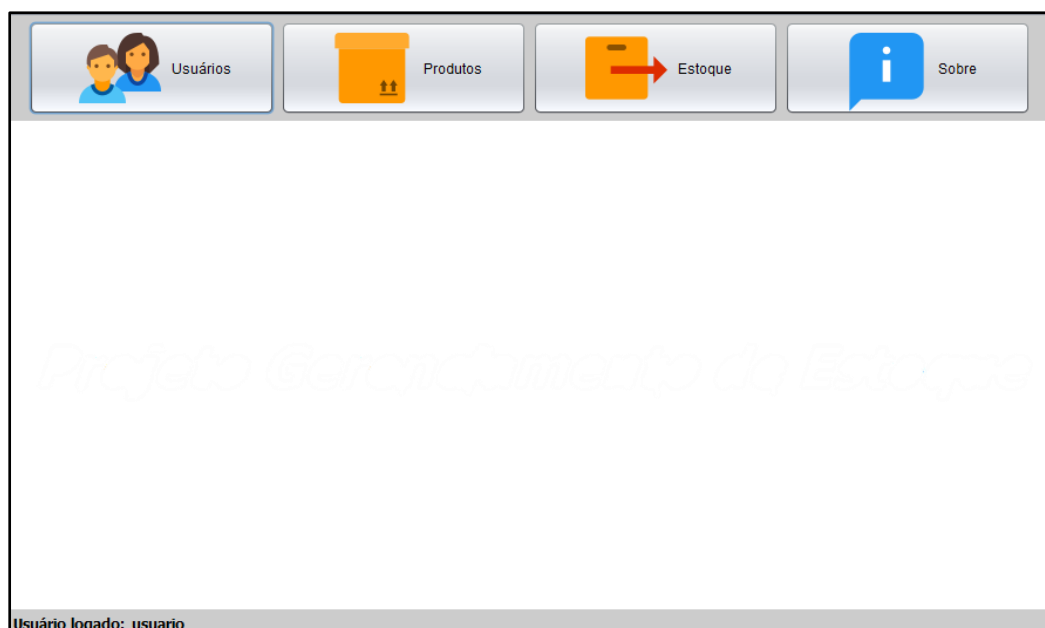


Figura 12. Tela de Menu Inicial

Fonte: Elaborada pelo autor.

A primeira opção Usuários diz respeito ao caso de uso Manter Usuário (Figura 13), a qual mostra a interface desenvolvida para fazer todas as operações básicas dos usuários.

Código	Nome	Login
1	Adminsitrador	admin
2	Peterson	petbcf
3	Daniela	dani
4	André	deca

Figura 13. Tela do Cadastro de Usuário

Fonte: Elaborada pelo autor.

Com o intuito de apresentar parte do desenvolvimento, a Figura 14 mostra o código em JAVA desenvolvido para realizar toda a parte da regra de negócio do funcionamento da interface do caso de uso Manter Produto, onde será possível inserir, excluir e consultar produtos. Seguindo o padrão MVC, no pacote *View* temos a classe *viewLogin.java* que é responsável por toda a configuração dos elementos/componentes da interface. No pacote *Controller* temos a classe *ControllerProduto.java* a qual possui os métodos de validação do preenchimento dos campos (linha 228), os métodos da ação que será executada ao clicar em um botão, por exemplo salvar ou alterar um produto e outros.

```

219 private void jbSalvarActionPerformed(java.awt.event.ActionEvent evt) {
220
221     modelProduto.setProDescricao(this.jtfDescricao.getText());
222     modelProduto.setProQuantidade(0);
223     modelProduto.setProValor(0.0d);
224     if (!this.estaVazio(this.jtfValor)) {
225         modelProduto.setProValor(FORMATADOR.converterVirgulaParaPonto(this.jtfValor.getText()));
226     }
227
228     if (CONTROLLER_PRODUTO.validarProdutoFormController(modelProduto)) {
229         if (salvarAlterar.equals("salvar")) {
230             //salvar um novo produto
231             if (CONTROLLER_PRODUTO.salvarProdutoController(modelProduto)) {
232                 JOptionPane.showMessageDialog(this, Mensagens.getMessage("PRD_SALVO"), Mensagens.getMessage("INFO"), JOptionPane.INFORMATION_MESSAGE);
233                 this.carregarProdutos();
234                 this.limparCampos();
235             } else {
236                 JOptionPane.showMessageDialog(this, Mensagens.getMessage("PRD_ERRO_SALVAR"), Mensagens.getMessage("ERRO"), JOptionPane.ERROR_MESSAGE);
237             }
238         } else if (salvarAlterar.equals("alterar")) {
239             //alterar um produto cadastrado
240             if (CONTROLLER_PRODUTO.alterarProdutoController(modelProduto)) {
241                 JOptionPane.showMessageDialog(this, Mensagens.getMessage("PRD_ALTERADO"), Mensagens.getMessage("INFO"), JOptionPane.INFORMATION_MESSAGE);
242                 this.carregarProdutos();
243                 this.limparCampos();
244             } else {
245                 JOptionPane.showMessageDialog(this, Mensagens.getMessage("PRD_ERRO_ALTERAR"), Mensagens.getMessage("ERRO"), JOptionPane.ERROR_MESSAGE);
246             }

```

Figura 14. Trecho do Código da Tela Cadastro de Produto

Fonte: Elaborada pelo autor.

A tela referente ao caso de uso Manter Produto pode ser vista na Figura 15, interface na qual está seguindo um padrão conforme as outras telas mostradas acima, um tema muito discutido na disciplina de Iteração Humano-Computador (IHC), que preza pela consistência entre as interfaces. Essa tela, portanto, tem por função realizar toda a parte de cadastramento de novos produtos, além de também possuir funções para atualização ou para a exclusão de um produto caso necessário.

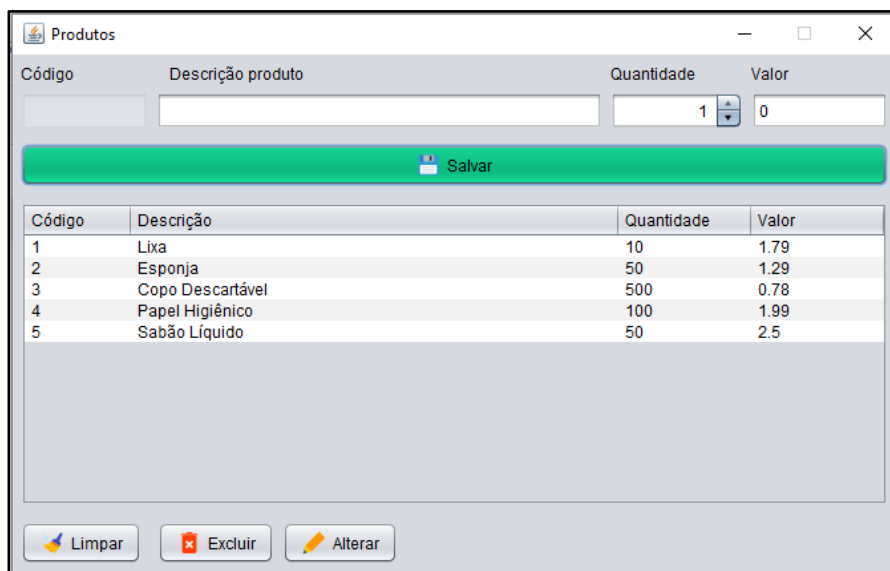


Figura 15. Tela do Cadastro de Produtos

Fonte: Elaborada pelo autor.

O caso de uso Movimentar Estoque (Figura 16), tem a principal função para o gerenciamento do almoxarifado, realiza movimentações de entrada e saída de produtos no estoque, dessa maneira é possível verificar quando um produto saiu ou entrou no estoque, qual a quantidade retirada ou inserida, o valor unitário desse produto, identificar quem foi o responsável por aquela transação, além de também poder realizar uma pesquisa nos dados.

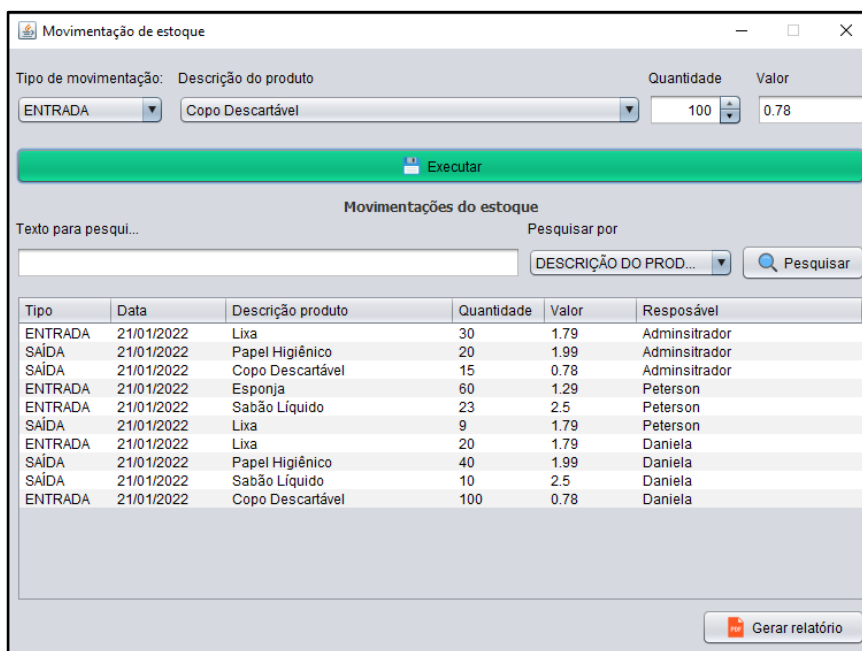


Figura 16. Tela Movimentação de Estoque

Fonte: Elaborada pelo autor.

Considerando a imagem apresentada na Figura 16 para inserir uma nova movimentação, basta você selecionar no componente *ComboBox* o tipo de movimentação, que pode ser uma entrada ou saída e na sequência ao selecionar o produto no *ComboBox* descrição do produto, o campo de valor já é preenchido automaticamente com o seu respectivo preço cadastrado anteriormente na tela de produtos, faltando apenas o preenchimento da quantidade por meio de um componente de interface de usuário do tipo *Spinner*.

Além disso, para essa interface também é possível realizar uma consulta dos dados utilizando diferentes filtros, com as opções de pesquisa por tipo de movimentação (entrada ou saída), por data da movimentação, pela descrição do produto e/ou pelo responsável da movimentação.

5.4 Testes

Foram realizados testes funcionais no sistema. São testes de caixa preta onde é verificado a funcionalidade desenvolvida. Neste trabalho, a cada caso de uso desenvolvido foi feita a verificação das operações básicas (inclusão, alteração, consulta e exclusão). Além de garantir o funcionamento das operações, foi verificado a gravação dos dados que foram apresentados anteriormente na Figura 16, no banco de dados conforme a Figura 17 mostra.

	pk est id	est quant	est preco	est tipo mov	est data	fk uso id	fk pro id
1	25	30	1.79	0	1642734000000	1	1
2	26	20	1.99	1	1642734000000	1	4
3	27	15	0.78	1	1642734000000	1	3
4	28	60	1.29	0	1642734000000	2	2
5	29	23	2.5	0	1642734000000	2	5
6	30	9	1.79	1	1642734000000	2	1
7	31	20	1.79	0	1642734000000	3	1
8	32	40	1.99	1	1642734000000	3	4
9	33	10	2.5	1	1642734000000	3	5
10	34	100	0.78	0	1642734000000	3	3

Figura 17. Tela do SQLite Studio com os Dados da Tela Movimentação de Estoque

Fonte: Elaborada pelo autor.

6. Considerações finais

Um dos problemas relacionados ao almoxarifado e o controle de estoque nas empresas é a perda de materiais durante a transição para o seu armazenamento, isso pode decorrer de uma gestão ineficiente durante as demandas. Desta forma, a presente pesquisa teve a intenção de comparar sistemas correlatos a este mesmo assunto e com os resultados desta analogia, desenvolver um protótipo de sistema que integrasse as funcionalidades mais comuns identificadas para que o mesmo ajude com o controle do estoque do setor de almoxarifado.

O *software* foi desenvolvido utilizando as ferramentas Apache NetBeans que oferece suporte para a codificação do aplicativo (regra de negócio) e toda a parte visual e *design* das telas (interfaces), além da utilização do SQLite Studio e MySQL Workbench para a construção e manuseio do banco de dados, ferramentas que por sua vez são *Open-Source* e permitem a sua utilização sem custo.

No desenvolvimento deste trabalho senti dificuldades em aplicar o método incremental, que foi o método escolhido inicialmente para trabalho, mas logo depois foi percebido o uso do modelo cascata, isso porque o desenvolvimento estava sendo feito de uma maneira sequencial, iniciando pela prototipação em desenho, construção do banco de dados e a programação do protótipo, sem pensar na divisão em incrementos e em melhorias em cada iteração conforme preza o modelo incremental.

Ainda durante o projeto tive dificuldades no início, pois eu não tinha uma máquina para trabalhar e realizar as atividades deste projeto, também em conseguir realizar o *setup* das ferramentas utilizadas quando consegui adquirir um computador pessoal, tendo em vista que ainda estamos na modalidade à distância do curso devido ao cenário pandêmico atual e que caso contrário poderíamos estar desfrutando das máquinas e instrumentos da instituição.

O objetivo do projeto foi desenvolver o protótipo de um sistema que atendesse as demandas de um setor empresarial, especificamente o almoxarifado. O sistema Almorganizado, desenvolvido parcialmente, atende a ideia inicial, pois o mesmo colabora com o gerenciamento dos dados de produtos e usuários, onde é possível realizar consultas, novas inserções, alterações e exclusões no banco de dados por meio do sistema. Além disso, também foi desenvolvido uma interface a qual é possível controlar as movimentações de entrada e saída de produtos, em outras palavras, realizar o controle de estoque. Entretanto, a funcionalidade para geração de relatórios de entrada e saída de produtos fica como trabalho futuro assim como poderá ser realizado a segregação das funções, podendo existir diferentes atores como funcionário, administrador, fornecedor e outros, conforme estudado na Seção 2 dos sistemas correlatos, além de novas funcionalidades como a validação da data de vencimento dos produtos, por exemplo.

Durante o desenvolvimento do trabalho de conclusão de curso é notória a utilização dos conhecimentos adquiridos no decorrer dos semestres, além de que, foi possível pôr em prática a maioria dos princípios e aprendizados do curso, como, por exemplo, conceitos das disciplinas: banco de dados para a criação e modelagem do banco utilizado no protótipo, engenharia de *software* e arquitetura de *software* para a construção dos diagramas e organização do código por meio do modelo de desenvolvimento MVC, linguagem de programação, programação orientada a objetos e estrutura de dados para todo o desenvolvimento do código do sistema, conexão com o banco de dados, métodos para as ações dos botões, salvamento no banco, entre outras disciplinas.

Referências Bibliográficas

Agência Brasil. Aumenta número de empresas abertas no país. Disponível em: <https://agenciabrasil.ebc.com.br/economia/noticia/2021-02/aumenta-numero-de-empresas-abertas-no-pais>. Acesso em: 5 jul. 2021.

BARBOSA, L. D. Criação de um layout otimizado para o novo almoxarifado de uma fábrica de produtos odontológicos. Paraná, 2017.

CPT Softwares. CPT Almoxarifado. Disponível em: <https://www.cptsoftwares.com.br/produto/cpt-almoxarifado-software-para-gerenciamento-de-almoxarifado/>. Acesso em: 28 jul. 2021.

DQL Sistemas. Software para almoxarifado de alta performance e baixo custo. Disponível em: https://www.dqlsistemas.com.br/Dql3_almoxarifado_prod.html. Acesso em: 28 jul. 2021.

- Engenharia Projetos e Construção. Noções Básicas de Almojarifado. Disponível em: <https://sites.google.com/site/engenhariaprojetoseconstrucao/suprimentos/nocoes-basicas-de-almojarifado>. Acesso em: 29 jun. 2021.
- GestãoClick. Programa para controle de almojarifado. Disponível em: <https://gestaoclick.com.br/programa-para-controle-de-almojarifado/>. Acesso em: 28 jul. 2021.
- GROSBELLI, A. C. Proposta de melhoria contínua em um almojarifado utilizando a ferramenta 5W2H. Paraná, 2014.
- JusBrasil. IBPT divulga censo sobre empresas do país: são 12,9 milhões. Disponível em: <https://crc-pr.jusbrasil.com.br/noticias/100417738/ibpt-divulga-censo-sobre-empresas-do-pais-sao-12-9-milhoes#:~:text=O%20Instituto%20Brasileiro%20de%20Planejamento,do%20pa%C3%AAs%20e%20o%20setores..> Acesso em: 5 jul. 2021.
- MACHADO, J. Problemas relacionados com a falta de material em almojarifado da administração pública. Brasília, 2011.
- Oracle. O Que É um Banco de Dados? Disponível em: <https://www.oracle.com/br/database/what-is-database/>. Acesso em: 5 jul. 2021.
- PRESSMAN, R. S. Engenharia de Software- (6ª edição), São Paulo, Ed. McGrawHill, 2006.
- RIBEIRO, L. O que é UML e Diagramas de Caso de Uso: Introdução Prática à UML. Disponível em: <https://www.devmedia.com.br/o-que-e-uml-e-diagramas-de-caso-de-uso-introducao-pratica-a-uml/23408>. Acesso em: 19 out. 2021.
- RIOS, E; MOREIRA, T. Teste de Software. 3. ed. Rio de Janeiro: Alta Books, c2013. 304 p. ISBN 978-85-7608-775-5.
- RODRIGUES, J. W, L. Estudo de layout e condições de trabalho em uma empresa de serviços industriais. Rio Grande do Sul, 2018.
- SOARES, B. C. Requisitos para utilização de prototipagem evolutiva nos processos de desenvolvimento de software para Web. Disponível em: <https://homepages.dcc.ufmg.br/~rodolfo/dcc823-2-07/Entrega3/Bruno3.pdf>. Acesso em: 27 jul. 2021.
- SoftExpert. Gestão de Almojarifado. Disponível em: <https://www.softexpert.com/pt-br/produto/gestao-almojarifado/>. Acesso em: 28 jul. 2021.
- SOMMERVILLE, I. Engenharia de Software. 10. ed. São Paulo: Pearson, c2019. 768 p. ISBN 9788543024974.
- SOMMERVILLE, I. Engenharia de Software. 8. ed. São Paulo: Pearson, c2007. 568 p. ISBN 9788588639287.
- SOMMERVILLE, I. Engenharia de Software. 9. ed. São Paulo: Pearson, c2011. 548 p. ISBN 9788579361081.
- Universidade Federal do Piauí. Manual do Almojarifado. Disponível em: https://www.ufpi.br/arquivos_download/arquivos/Parnaiba/fluxogramasemanuais/Manual_Almojarifado.pdf. Acesso em: 8 jul. 2021.
- WAZLAWICK, R. S. Análise e *design* orientados a objetos para sistemas de informação: modelagem com UML, OCL e IFML. 3. ed. Rio de Janeiro: Elsevier, c2015. 462 p. ISBN 9788535279849.

Documento Digitalizado Público

Anexo I - Artigo final

Assunto: Anexo I - Artigo final
Assinado por: Daniela Marques
Tipo do Documento: Projeto
Situação: Finalizado
Nível de Acesso: Público
Tipo do Conferência: Documento Digital

Documento assinado eletronicamente por:

- Daniela Marques, PROFESSOR ENS BASICO TECN TECNOLOGICO, em 27/03/2022 20:23:55.

Este documento foi armazenado no SUAP em 27/03/2022. Para comprovar sua integridade, faça a leitura do QRCode ao lado ou acesse <https://suap.ifsp.edu.br/verificar-documento-externo/> e forneça os dados abaixo:

Código Verificador: 927915

Código de Autenticação: fa019d8054

