

**ETEC JARDIM ÂNGELA**

**DESENVOLVIMENTO DE SISTEMAS**

JAKELINE DOMINGOS DO SANTOS

JULIA DA SILVA TEIXEIRA

THIAGO DA SILVA SANTOS

**ZAMP**

ZAMP

Trabalho apresentado a Etec Jardim Ângela como requisito para conclusão do curso de Desenvolvimento de Sistemas

Orientador: Prof. Fernando Faria

SÃO PAULO

2024

**DEDICATÓRIA**

Dedicamos este projeto primeiramente a Deus, aos nossos amigos e familiares, agradecemos a todos pelo auxílio e apoio dos mesmos.

Agradecemos também a nossos professores por terem nos ensinando e instruído até o presente momento.

Dedicamos também a nós por todo o esforço e dedicação e colocamos aqui.

Nossos mais sinceros agradecimentos a todos citados acima, sem vocês não chegaríamos aqui.

Obrigada.

# **RESUMO**

Este projeto propõe o desenvolvimento de um software para a empresa ZAMP, com o objetivo de automatizar o gerenciamento de pedidos entre a Matriz e suas filiais. O sistema irá controlar estoque, pedidos, trocas, devoluções e gerar notas fiscais, substituindo o atual método que é utilizando as planilhas no Excel. Com esse software eles terão acessos separados, registro de histórico de movimentações e maior organização dos processos logísticos. Com isso, espera-se melhorar o controle operacional e a eficiência da empresa.

**Palavras-chave:** Automação de processos; Sistema de gestão; Controle de estoque; Software;

# **ABSTRACT**

This project proposes the development of software for the company ZAMP, aiming to automate the order management process between the headquarters and its branches. The system will handle inventory control, orders, exchanges, returns, and the issuance of invoices, replacing the current method based on Excel spreadsheets. With this software, users will have separate access levels, historical tracking of operations, and improved organization of logistical processes. As a result, the company is expected to achieve better operational control and increased efficiency.

**Keywords:** Process automation; Management system; Inventory control; Software.

# **LISTA DE IMAGENS**

[Figura 1 Dashboar Matriz 4](#_Toc201169872)

[Figura 2 Gerenciamento de Pedidos Matriz 5](#_Toc201169873)

[Figura 3 Gerenciamento de Produtos Matriz 6](#_Toc201169874)

[Figura 4 Gerenciamento de Usuários Matriz 6](#_Toc201169875)

[Figura 5 Sistema de Pedidos Usuários 7](#_Toc201169876)

[Figura 6 Escolher opção Usuário 8](#_Toc201169877)

[Figura 7 Lista de Produtos Usuário 8](#_Toc201169878)

[Figura 8 Doar Produtos Usuário 9](#_Toc201169879)

[Figura 9 Troca de Produtos Usuários 10](#_Toc201169880)

[Figura 10 Histórico de Pedidos Usuários 10](#_Toc201169881)

[Figura 11. Tabela de tarefas 12](#_Toc201169882)

[Figura 12. Modelo conceitual 13](#_Toc201169883)

[Figura 13/ Modelo Lógico de Banco de Dados 14](#_Toc201169884)

[Figura 14/ Diagrama de Caso de Uso UML 15](#_Toc201169885)

**Sumário**

[**RESUMO** 2](#_Toc201172152)

[**ABSTRACT** 2](#_Toc201172153)

[**LISTA DE IMAGENS** 2](#_Toc201172154)

[**1. INTRODUÇÃO** 1](#_Toc201172155)

[**2. OBJETIVO GERAL** 2](#_Toc201172156)

[**2.1 OBJETIVO ESPECÍFICOS** 2](#_Toc201172157)

[**3 METODOLOGIA** 3](#_Toc201172158)

[**3.1 FERRAMENTAS UTILIZADAS** 3](#_Toc201172159)

[ **3.1.1 HTML5** 3](#_Toc201172160)

[ **3.1.2 CSS3** 3](#_Toc201172161)

[ **3.1.3 Bootstrap** 3](#_Toc201172162)

[ **3.1.4 JavaScript** 3](#_Toc201172163)

[ **3.1.5 PHP** 3](#_Toc201172164)

[ **3.1.6 Python** 3](#_Toc201172165)

[ **3.1.7 MySQL** 3](#_Toc201172166)

[ **3.1.8 GitHub** 3](#_Toc201172167)

[ **3.1.9 Visual Studio Code** 3](#_Toc201172168)

[ **3.1.10 Google Colab** 3](#_Toc201172169)

[ **3.1.11 TasksBoard** 3](#_Toc201172170)

[ **3.1.12 Scrum** 3](#_Toc201172171)

[**4. DESENVOLVIMENTO** 5](#_Toc201172172)

[**4.1 MATRIZ** 5](#_Toc201172173)

[**4.2 Gerenciamento de pedidos** 6](#_Toc201172174)

[**4.3 Gerenciamento de Produtos** 7](#_Toc201172175)

[**4.4 Gerenciamento de usuário** 7](#_Toc201172176)

[**4.1.1 Usuário** 8](#_Toc201172177)

[**4.1.2. Escolha uma opção** 9](#_Toc201172178)

[**4.1.3. Fazer pedido** 9](#_Toc201172179)

[**4.1.4. Doar produto** 10](#_Toc201172180)

[**4.1.5. Trocar produtos** 11](#_Toc201172181)

[**4.1.6 Histórico de pedidos** 11](#_Toc201172182)

[**5. Imagens** 13](#_Toc201172183)

[**5.1. ESTRUTURA DO TRABALHO** 13](#_Toc201172184)

[**5.2. Modelo Conceitual do Banco de Dados** 14](#_Toc201172185)

[**5.3. Modelo Lógico do Banco de Dados** 15](#_Toc201172186)

[**5.4 Diagrama de Caso de Uso UML** 16](#_Toc201172187)

[**8. CONCLUSÃO** 17](#_Toc201172188)

# **1. INTRODUÇÃO**

O projeto consiste em fazer a implementação de um software para a empresa ZAMP com o objetivo de automatizar o gerenciamento de pedidos feitos pelas filiais da empresa. O objetivo desse software é ter um controle sobre estoque, pedidos de produtos, trocas, devoluções e gerar notas fiscais.

Com isso vamos precisar cadastrar as filiais e Matriz, para que cada uma tenha acesso a uma parte determinada do site. A Matriz recebera os pedidos por meio do sistema, os responsáveis por separar as solicitações e enviar terão acesso a cada pedido. Com isso facilitaremos os envios, as trocas e as devoluções dos produtos que foram solicitados de maneira incorreta, pois foi constatado que a empresa apresenta dificuldades no gerenciamento dos pedidos, devido à forma com que controla suas entradas e saídas de itens.

Atualmente os pedidos são controlados pelo programa Excel, é armazenado em nuvem no qual todos os gerentes de filiais têm acesso, podendo solicitar o envio, a troca e a devolução de produtos não constando um histórico ou nota fiscal para controle de entrada e saída. Em razão disso, o software fará a implementação de um histórico buscando armazenar as informações de acesso dos gerentes de cada filial para controle de entrada e saída do estoque.

A implementação desse novo modelo de operação visa não apenas melhorar a forma que os pedidos são feitos, mas também proporcionar uma maneira de trabalho mais organizada, responsável, completa e adequada para o trabalho, devido as condições que se encontram o controle de armazenamento. Ao longo deste projeto, será analisado as necessidades específicas da empresa, as maneiras mais adequadas para a automação dos processos e os impactos esperados dessa transformação no funcionamento geral da organização.

# **2. OBJETIVO GERAL**

Facilitar o processo de pedido de produtos da empresa, fazendo com que a solicitação de produtos, troca e devolução sejam feitos e uma maneira eficaz.

## **2.1 OBJETIVO ESPECÍFICOS**

* Desenvolver um software dinâmico que integre todos os requisitos necessários para solicitação de produtos, troca e devolução.
* Facilitar o controle de estoque.
* Editar o estoque
* Efetuar o cadastro de todas as filiais.
* Administrar as filiais cadastradas

# **3 METODOLOGIA**

Com base em uma pesquisa de campo, verificou-se que os funcionários da matriz enfrentavam dificuldades na administração dos pedidos, uma vez que os colaboradores das filiais possuíam acesso a todos os pedidos — tanto os da própria unidade quanto os das demais —, o que resultava em alterações indevidas em solicitações de outras filiais, acúmulo dos pedidos, causando transtorno e demora na realização dos pedidos. Diante dessas informações e após a realização de uma reunião com um colaborador, identificou-se a necessidade de desenvolver um software que facilitasse o trabalho da equipe. Para isso, utilizamos as seguintes

## **3.1 FERRAMENTAS UTILIZADAS**

* **3.1.1 HTML5** (HyperText Markup Language): utilizado para a estruturação das páginas, incluindo todo o conteúdo textual e os botões de navegação;
* **3.1.2 CSS3** (Cascading Style Sheets): empregado na estilização geral do site, tanto por meio de codificação manual quanto com o auxílio do framework Bootstrap;
* **3.1.3 Bootstrap**: utilizado para tornar o sistema responsivo, adaptando-o a diferentes tamanhos de tela e dispositivos;
* **3.1.4 JavaScript**: utilizado para o desenvolvimento de funcionalidades dinâmicas e interativas, como o funcionamento do carrinho de compras, abertura e fechamento do menu lateral, mensagens de alerta (message boxes), entre outras animações e interações com o usuário;
* **3.1.5 PHP**: empregado na manipulação de dados, tanto na recepção de informações do banco de dados (como produtos, usuários e pedidos), quanto no envio de novos dados, como o cadastro de pedidos, usuários e produtos;
* **3.1.6 Python**: utilizado para automatizar o processo de formatação do estoque da matriz. Como os dados foram fornecidos em planilhas do Excel, foi desenvolvido um script em Python para leitura desses arquivos e geração de comandos SQL INSERT, facilitando a inserção dos produtos no banco de dados;
* **3.1.7 MySQL** (via Laragon): escolhido pela facilidade de uso e praticidade para o desenvolvimento e testes em ambiente local;
* **3.1.8 GitHub**: utilizado como ferramenta de versionamento de código, permitindo o registro de atualizações por meio de commits frequentes e garantindo acesso a todas as modificações realizadas, promovendo melhor organização e colaboração;
* **3.1.9 Visual Studio Code** (VSCode): utilizado como ambiente de desenvolvimento para a escrita e organização do código-fonte, por oferecer recursos como extensões, integração com o GitHub e suporte a múltiplas linguagens;
* **3.1.10 Google Colab**: utilizado para a execução do código em Python responsável pela leitura das planilhas do Excel e conversão dos dados em comandos SQL INSERT, facilitando a integração com o banco de dados;
* **3.1.11 TasksBoard**: utilizado como ferramenta de organização e acompanhamento das tarefas da equipe, funcionando como um cronograma visual e colaborativo;
* **3.1.12 Scrum**: metodologia ágil adotada para o gerenciamento do projeto. Por meio dessa abordagem, foram realizadas reuniões frequentes para definir, acompanhar e revisar as etapas do desenvolvimento, promovendo maior organização, divisão de responsabilidades e entregas contínuas ao longo do processo.

# **4. DESENVOLVIMENTO**

Após a coleta e análise das informações obtidas por meio de pesquisas e diálogos com os responsáveis pelo sistema, estabeleceu-se a forma mais adequada para o desenvolvimento do software, a qual será descrita a seguir.

O site terá dois logins: matriz e usuário.

## **4.1 MATRIZ**

Interface gráfica do usuário, Aplicativo, Site

O conteúdo gerado por IA pode estar incorreto.

Figura Dashboar Matriz

Para acessar a conta da matriz, os administradores deverão utilizar um endereço de e-mail e uma senha. Após a autenticação, o administrador será redirecionado à tela do painel de controle (dashboard), a qual será composta pelos seguintes elementos:

## **4.2 Gerenciamento de pedidos**

Figura Gerenciamento de Pedidos Matriz

**Interface gráfica do usuário, Texto, Aplicativo, Email

O conteúdo gerado por IA pode estar incorreto.**

Na seção de gerenciamento de pedidos, o administrador terá acesso detalhado a todos os pedidos das filiais, organizados em ordem decrescente de data, do mais recente ao mais antigo. Essa seção contará com filtros que permitirão a categorização dos pedidos conforme o seu status. Os filtros disponíveis serão: todos os pedidos, novos pedidos, em processo, finalizados, requisição, troca, doação e devolução. A utilização desses filtros possibilitará um controle mais eficiente e prático dos pedidos, conforme a preferência e necessidade do usuário.

## Interface gráfica do usuário, Aplicativo O conteúdo gerado por IA pode estar incorreto.**4.3 Gerenciamento de Produtos**

Figura Gerenciamento de Produtos Matriz

Na seção de gerenciamento de produtos, a Matriz terá acesso completo ao estoque, podendo visualizar todos os itens cadastrados no sistema. Será possível realizar ações como a exclusão, a adição e a edição das informações dos produtos, conforme a necessidade da organização. Os itens estarão organizados por categorias, como **SKU** (Stock Keeping Unit) e tipo de produto, o que tornará a navegação mais intuitiva e contribuirá para um controle de inventário mais eficiente.

## **4.4 Gerenciamento de usuário**

**Interface gráfica do usuário, Texto, Aplicativo, Email

O conteúdo gerado por IA pode estar incorreto.**

Figura Gerenciamento de Usuários Matriz

Na seção de gerenciamento de usuários, a matriz possui acesso a todas as filiais, organizadas em ordem sistemática. É permitido à administração realizar alterações nas informações das filiais, tais como: nome do usuário, CNPJ, responsável, endereço, CEP, bairro, cidade e unidade federativa (UF). Além disso, é possível adicionar novas filiais ou excluir as existentes, o que contribui para uma gestão mais eficiente e centralizada das unidades.

### **4.1.1 Usuário**

Interface gráfica do usuário, Aplicativo

O conteúdo gerado por IA pode estar incorreto.

Figura Sistema de Pedidos Usuários

Para que as filiais acessem suas respectivas contas, cada unidade contará com um e-mail e uma senha individualizados. Após a autenticação, o usuário será direcionado à tela do sistema de pedidos, a qual será composta pelos seguintes elementos:

### **4.1.2. Escolha uma opção**

Interface gráfica do usuário, Aplicativo, Teams

O conteúdo gerado por IA pode estar incorreto.

Figura Escolher opção Usuário

Ao acessar a página de realização de pedidos, o usuário será redirecionado para uma nova interface, intitulada “Escolha uma opção”. Nessa tela, estarão disponíveis três funcionalidades principais: fazer pedidos, doar produtos e trocar produtos. Ao selecionar a opção "fazer pedidos", o sistema redirecionará o usuário para uma nova página contendo a lista de produtos disponíveis, possibilitando, assim, a seleção dos itens desejados para concluir o pedido.

### **4.1.3. Fazer pedido**

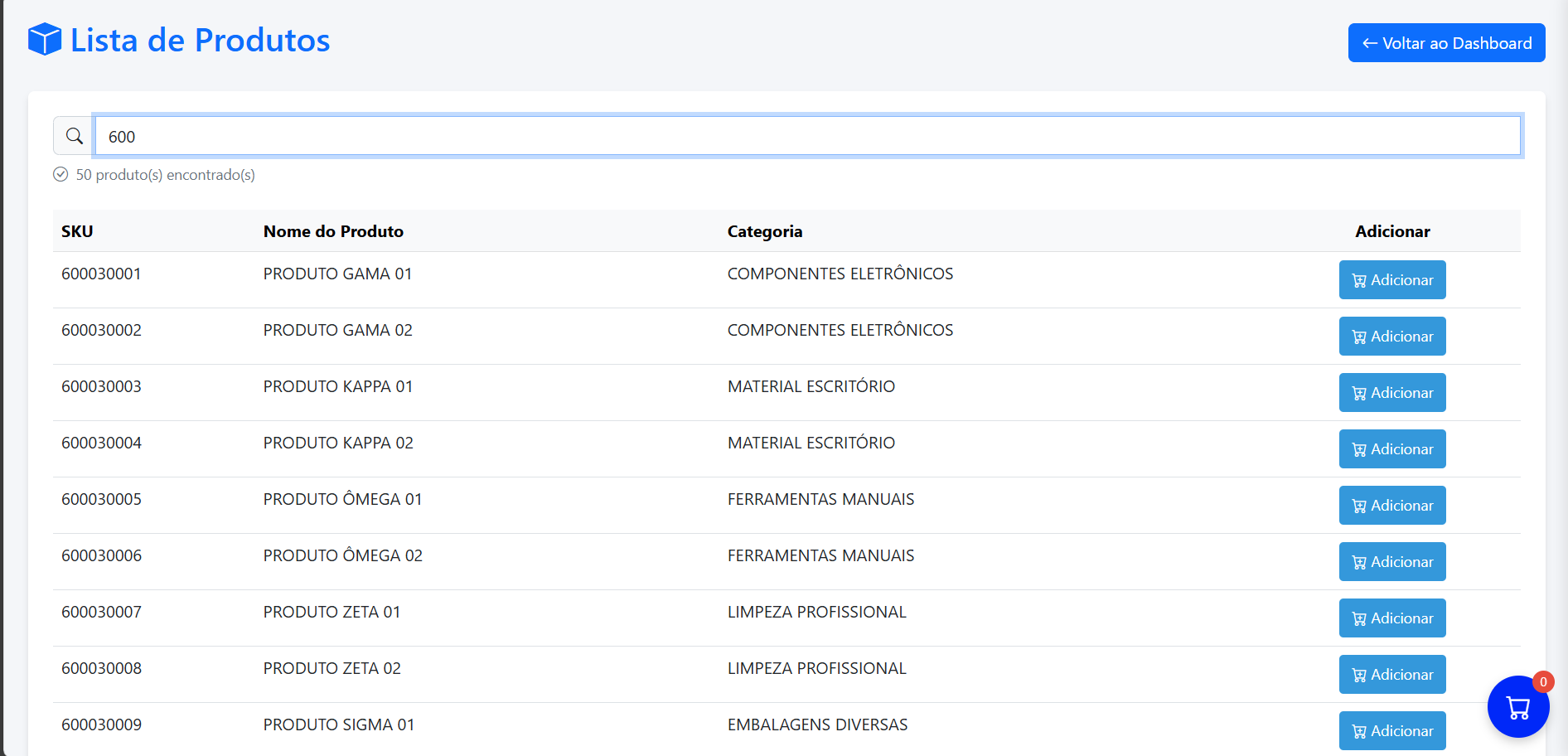


Figura Lista de Produtos Usuário

Ao clicar na opção "Fazer Pedido", será aberta uma nova página na qual o usuário poderá pesquisar o produto desejado por nome, categoria ou SKU. Após localizar o item, será possível adicioná-lo ao carrinho e modificar o que desejar, em seguida, finalizar o pedido.

### **4.1.4. Doar produto**

Interface gráfica do usuário, Aplicativo

O conteúdo gerado por IA pode estar incorreto.

Figura Doar Produtos Usuário

Ao selecionar a opção "Doar Produto", o usuário será direcionado para uma interface semelhante à da função "Fazer Pedido". A principal diferença estará na etapa final: após adicionar os produtos ao carrinho e selecionar a opção "Finalizar Doação", será exibida uma lista com todas as filiais disponíveis para receber a doação. Além disso, o sistema permitirá a inclusão de observações referentes aos produtos doados.

### **4.1.5. Trocar produtos**

Interface gráfica do usuário, Texto, Aplicativo, Email

O conteúdo gerado por IA pode estar incorreto.

Figura Troca de Produtos Usuários

Ao selecionar a opção "Trocar Produtos", o usuário será direcionado para a tela de troca, na qual será possível selecionar os produtos que serão enviados, bem como aqueles que estão sendo solicitados em retorno. Na parte inferior da página, estará disponível uma lista com as filiais, além de um campo para inserção de observações sobre a troca, permitindo ao usuário indicar a unidade com a qual deseja realizar a operação

### **4.1.6 Histórico de pedidos**

Interface gráfica do usuário, Aplicativo

O conteúdo gerado por IA pode estar incorreto.

Figura Histórico de Pedidos Usuários

Na seção de histórico de pedidos, a filial poderá visualizar todos os pedidos nos quais esteja envolvida, seja como unidade de origem ou de destino. Haverá a opção de visualizar os detalhes de cada pedido, incluindo informações como horário, produtos envolvidos, filial de destino e a possibilidade de devolução. Caso a filial tenha recebido um pedido de doação, será possível realizar a devolução total ou parcial dos itens, conforme a necessidade.

# **5. Imagens**

## **5.1. ESTRUTURA DO TRABALHO**



Figura . Tabela de tarefas

## **5.2. Modelo Conceitual do Banco de Dados**

Figura . Modelo conceitual

## **5.3. Modelo Lógico do Banco de Dados**

Diagrama

O conteúdo gerado por IA pode estar incorreto.

Figura / Modelo Lógico de Banco de Dados

## **5.4 Diagrama de Caso de Uso UML**

Gráfico

O conteúdo gerado por IA pode estar incorreto.

Figura / Diagrama de Caso de Uso UML

# **8. CONCLUSÃO**

Durante o desenvolvimento do projeto, foram encontradas diversas dificuldades, especialmente relacionadas à organização e estruturação do código-fonte. Devido à falta de experiência prévia da equipe, muitos trechos do sistema apresentavam mais de mil linhas de código contínuo, o que dificultava a localização de elementos específicos e comprometia a manutenção e a modelagem do projeto.

Diante desse cenário, buscou-se estudar e aplicar boas práticas de desenvolvimento, com o objetivo de tornar o código mais modular, legível e eficiente. Uma das estratégias adotadas foi a utilização de arquivos **include** para componentes recorrentes, como **header**, **menu lateral**, **footer** e **conexão com o banco de dados**. Essa abordagem permitiu dividir o código em partes reutilizáveis e independentes, reduzindo significativamente a quantidade de linhas por arquivo e facilitando tanto a organização quanto a manutenção do sistema.

Essa experiência contribuiu para o amadurecimento técnico da equipe, promovendo um maior entendimento sobre a importância da estrutura modular no desenvolvimento de sistemas web. Além disso, permitiu aplicar conceitos práticos de programação, como reaproveitamento de código, separação de responsabilidades e clareza na organização do projeto, que serão de grande valor em futuros projetos acadêmicos ou profissionais.