

Distribuidora Ferreira

Software de gerência de distribuidoras de bebidas

Ana Flávia Carvalho Santos¹, Douglas Galvão Souza Machado²,
Gabriel Ramos Ferreira³, João Pedro Silva Braga³, João Vitor Romero Sales⁴,
Júlia Moreira Nascimento⁵

¹Departamento de Engenharia de Software
Pontifícia Universidade Católica de Minas Gerais (PUC-Minas)
Avenida Dom José Gaspar 500, Coração Eucarístico, Belo Horizonte, MG

Abstract. *This paper presents a software system solution for Distribuidora Ferreira, providing an integrated management system that covers everything from inventory management to financial control. By using Angular for the front-end, Java for the back-end, and SQL Server for data, the project aims to automate inventory processes, optimize delivery logistics, and integrate financial management. Agile methodology and DevOps practices ensure fast and reliable deliveries. The project is expected to reduce operational costs, increase efficiency and customer satisfaction, and improve resource management. The digital transformation will position the company to face new challenges and growth opportunities.*

Resumo. *Este trabalho apresenta uma solução de sistema de software para a Distribuidora Ferreira, oferecendo um sistema de gestão integrado que abrange desde a gestão de estoque até o controle financeiro. Utilizando Angular para o front-end, Java no back-end, e SQL Server para dados, o projeto visa automatizar processos de estoque, otimizar a logística de entregas e integrar a gestão financeira. A metodologia ágil e as práticas DevOps garantem entregas rápidas e confiáveis. Espera-se reduzir custos operacionais, aumentar a eficiência e a satisfação dos clientes e melhorar a gestão de recursos. A transformação digital posicionará a empresa para enfrentar novos desafios e oportunidades de crescimento.*

Contents

1	Apresentação do Problema	4
2	Stakeholders	5
2.1	Personas	5
2.2	Outros Stakeholders	6
2.3	Potenciais Conflitos de Interesse	7
2.4	Feedback e Comunicação	7
3	Proposta da Solução	8
3.1	Escopo da Solução	8
3.1.1	Fora de Escopo	8
3.2	Requisitos de Software	8
3.2.1	Requisitos Funcionais	9
3.2.2	Requisitos Não Funcionais	9
3.2.3	Técnicas de Elicitação de Requisitos Utilizadas	10
3.3	Diagrama de Caso de Uso	11
3.4	Diagrama de Classe	12
4	Projeto da Solução	13
4.1	Descrição Técnica da Solução	13
4.1.1	Arquitetura do Software	13
4.1.2	Angular Material	13
4.1.3	MVC (Model-View-Controller)	13
4.1.4	Spring Boot	13
4.1.5	Tecnologias Envolvidas	13
4.1.6	Controle de Versão com Git e GitHub	13
4.1.7	Banco de Dados PostgreSQL	14
4.1.8	Método de Desenvolvimento Scrum	14
4.2	Planejamento das Telas	14
4.2.1	Login Page	15
4.2.2	Home Page	16
4.2.3	Página de Comandas	17
4.2.4	Página de Estoque	18

4.2.5	Página de Vendas	19
5	Artefatos Principais	20
6	Conclusão	21
6.1	Principais Resultados e Contribuições	21
6.2	Lições Aprendidas	21
7	Referências	22

1. Apresentação do Problema

A Distribuidora Ferreira é uma empresa de médio porte localizada em Belo Ho, conhecida por sua base de clientes fiel. A gestão eficiente de vendas e estoque é crucial para o sucesso e crescimento da distribuidora. Atualmente, a empresa enfrenta vários desafios devido à gestão manual desses processos.

O principal problema é o gerenciamento ineficiente de vendas e estoque. Todas as transações e registros de estoque são realizados manualmente através de planilhas. Este método manual é não só demorado e propenso a erros, mas também dificulta a análise rápida e precisa dos dados. Erros na entrada de dados podem levar a discrepâncias no inventário, resultando em perda de vendas ou excesso de estoque.

Além disso, a distribuidora planeja expandir seu negócio para incluir um sistema de comandas, permitindo que os clientes consumam no local. Esta expansão adiciona uma camada de complexidade ao gerenciamento do negócio, exigindo uma solução capaz de rastrear vendas, estoque e comandas individuais dos clientes de maneira integrada e eficiente.

Os impactos negativos da gestão manual são significativos:

- **Erros Humanos:** A propensão a erros na entrada manual de dados pode resultar em problemas de inventário e perda de vendas.
- **Ineficiência:** O tempo excessivo gasto em tarefas manuais reduz a produtividade e a capacidade de resposta da equipe.
- **Dificuldade na Análise de Dados:** A obtenção de uma visão precisa e rápida do estado do estoque e das vendas é dificultada pela falta de automação.

Esses problemas não só afetam a eficiência operacional da distribuidora, mas também impactam negativamente a satisfação dos clientes, que podem enfrentar atrasos ou indisponibilidade de produtos. A expansão planejada aumenta ainda mais a necessidade de uma solução automatizada que possa lidar com a complexidade adicional de um sistema de comandas.

Portanto, é evidente a necessidade de uma solução eficiente e automatizada para os processos de vendas e estoque. A implementação de um sistema integrado e automatizado não apenas reduzirá os erros e aumentará a eficiência, mas também melhorará a satisfação do cliente e permitirá que a empresa se posicione melhor para enfrentar novos desafios e oportunidades de crescimento.

2. Stakeholders

Os stakeholders são todas as partes interessadas no desenvolvimento e uso do sistema de gerenciamento da Distribuidora Ferreira. A seguir, detalhamos as principais personas envolvidas, suas expectativas, interesses, influências e interações com o sistema.

2.1. Personas

Personas são representações fictícias de usuários ideais de um sistema ou produto, criadas com base em dados e pesquisa de usuários reais. Elas ajudam a equipe de desenvolvimento a entender melhor as necessidades, comportamentos, objetivos e desafios dos diferentes tipos de usuários que irão interagir com o sistema. As personas são utilizadas para orientar decisões de design e desenvolvimento, garantindo que o produto final seja útil e relevante para seus usuários [Sommerville 2011].

Ao criar personas, a equipe de desenvolvimento ganha insights valiosos sobre como diferentes usuários interagem com o sistema, permitindo a criação de uma solução mais intuitiva e eficaz.

No contexto do sistema de gerenciamento da Distribuidora Ferreira, as personas ajudam a entender os diferentes tipos de usuários que utilizarão o sistema, desde o dono da distribuidora até os funcionários e clientes [Pressman and Maxim 2014].

- **Nilto Ferreira**

- **Idade:** 50 anos
- **Profissão:** Dono da Distribuidora de Bebidas
- **Objetivos:**
 - * Melhorar a eficiência das operações.
 - * Expandir o negócio para incluir um sistema de comandas.
 - * Ter controle em tempo real sobre o estoque e as finanças.
- **Dores:**
 - * Gerenciamento ineficiente de vendas e estoque.
 - * Necessidade de adaptação à expansão do negócio.
 - * Falta de informações precisas para tomada de decisão.
- **Interações com o sistema:**
 - * Acessar relatórios de vendas e estoque.
 - * Monitorar o desempenho financeiro.
 - * Configurar parâmetros de negócio e autorizar grandes transações.
- **Interesse e Influência:** Alta influência na tomada de decisões e interesse crítico nos resultados do sistema.
- **Expectativas:** Sistema intuitivo, relatórios precisos, melhoria na eficiência operacional
- **João Silva**
 - * **Idade:** 30 anos
 - * **Profissão:** Funcionário da Distribuidora de Bebidas
 - * **Objetivos:**
 - Dificuldade no registro e gestão de comandas sem uma ferramenta adequada.
 - Tempo excessivo gasto na atualização manual de estoque.
 - Erros frequentes devido à entrada manual de dados.

- * **Interações com o sistema:**
 - Registrar vendas e comandas no sistema.
 - Atualizar o estoque em tempo real.
 - Gerar relatórios de desempenho diário.
 - * **Interesse e Influência:** Médio interesse e influência; essencial para a
 - * **Expectativas:** Sistema fácil de usar, integração fluida de tarefas diárias.
- **Maria Costa**
- * **Idade:** 35 anos
 - * **Profissão:** Cliente regular da Distribuidora de Bebidas
 - * **Objetivos:**
 - Consumir suas bebidas favoritas no local com conveniência.
 - Experimentar um serviço rápido e eficiente.
 - * **Dores:**
 - Preocupação com a eficiência e usabilidade do novo sistema de comandas.
 - Longas esperas para atendimento.
 - Falta de disponibilidade de produtos devido a má gestão de estoque.
 - * **Interações com o Sistema:**
 - Realizar pedidos diretamente no sistema de comandas.
 - Receber atualizações sobre o status do pedido.
 - Fazer pagamentos de forma rápida e segura.
 - * **Interesse e Influência:** Alto interesse no uso do sistema, baixa influência na tomada de decisões.
 - * **Expectativas:** Interface amigável, serviço rápido e confiável.

2.2. Outros Stakeholders

Além das personas detalhadas acima, outros stakeholders incluem fornecedores, contadores e gestores de logística, cada um com interesses específicos e expectativas quanto ao sistema [International Organization for Standardization 2011].

- **Interesse e Influência:** Alta influência na implementação e manutenção do sistema.
- **Expectativas:** Sistema estável, documentação completa, facilidade de manutenção.
- **Fornecedores:**
 - **Interesse e Influência:** Médio interesse em previsões precisas de estoque; influência na cadeia de suprimentos.
 - **Expectativas:** Visibilidade clara sobre as necessidades de reabastecimento, integração eficiente com o sistema de compras.
- **Contadores:**
 - **Interesse e Influência:** Alto interesse em relatórios financeiros precisos; influência nas conformidades fiscais.
 - **Expectativas:** Relatórios detalhados, precisão contábil, conformidade regulatória.

- **Gestores de Logística:**
 - **Interesse e Influência:** Alto interesse na eficiência logística; influência na operação do fluxo de produtos.
 - **Expectativas:** Monitoramento em tempo real do fluxo de produtos, otimização de rotas de entrega.

2.3. Potenciais Conflitos de Interesse

Embora todos os stakeholders estejam interessados no sucesso do sistema, podem surgir conflitos de interesse, tais como:

- **Nilton Ferreira vs. Funcionários:** O desejo de Nilton por uma expansão rápida pode colidir com a necessidade dos funcionários por mais treinamento e adaptação.
- **Clientes vs. Equipe de TI:** Clientes podem exigir funcionalidades adicionais que complicam a manutenção do sistema.

2.4. Feedback e Comunicação

A comunicação contínua e o feedback são cruciais para o sucesso do projeto:

- **Reuniões Semanais:** Com a equipe de desenvolvimento e principais stakeholders para atualizar o progresso e alinhar expectativas.
- **Sessões de Feedback:** Com usuários finais (funcionários e clientes) para coletar sugestões e identificar problemas.
- **Relatórios Mensais:** Para a direção, detalhando o progresso, desafios e próximos passos.

3. Proposta da Solução

3.1. Escopo da Solução

A solução proposta é uma aplicação web com um dashboard que inclui um gerenciador de estoque, permitindo o registro de entrada e saída de produtos no estoque, bem como a baixa no estoque quando um produto é vendido. Todas essas movimentações serão informadas junto com a visualização de preços e descrições adicionais. Também será registrado as vendas no local e pedidos agendados, contendo as informações principais da venda (data, horário, produtos, valor da compra e método de pagamento), criando um histórico de vendas para a empresa. Além disso, haverá um sistema de comandas para que os funcionários possam controlar o consumo das mesas do estabelecimento, gerando um registro de venda ao final do serviço.

O sistema tratará da abertura e fechamento de caixa e do controle de vendas, tanto no balcão quanto nas mesas.

3.1.1. Fora de Escopo

Fora de escopo refere-se a todas as funcionalidades e características que não estão incluídas no projeto e que não serão abordadas ou implementadas. Especificar o que está fora de escopo é fundamental para definir claramente os limites do projeto, evitando mal-entendidos, expectativas irreais e escopo indeterminado.

No contexto da solução descrita, os itens fora de escopo são:

- **Integração com aplicativos externos:** O sistema não será integrado com plataformas de entrega ou venda externas, como Ifood e Rappi. Isso significa que qualquer venda ou pedido realizado por esses aplicativos não será automaticamente registrado ou gerenciado pelo sistema desenvolvido.
- **Acesso dos clientes:** Os clientes do estabelecimento não terão acesso ao sistema. Isso implica que a aplicação é exclusivamente para uso interno dos funcionários e gerentes do estabelecimento.
- **Emissão de comprovantes fiscais:** O sistema não será responsável pela emissão de comprovantes fiscais de compra. Ou seja, qualquer necessidade de gerar recibos ou notas fiscais será tratada por outro software ou processo, fora do escopo deste projeto específico.

Definir claramente o que está fora do escopo ajuda a garantir que todos os envolvidos no projeto tenham uma compreensão comum do que será entregue e o que não será, permitindo um gerenciamento de expectativas mais eficiente e evitando demandas adicionais que possam comprometer o cronograma ou o orçamento do projeto.

3.2. Requisitos de Software

Requisitos de software são especificações detalhadas que definem as capacidades e restrições de um sistema de software. Eles são fundamentais para a construção de um software de alta qualidade, pois orientam todo o processo de desenvolvimento, desde o planejamento até a implementação e manutenção. Esses requisitos abrangem todos os aspectos do sistema, incluindo funcionalidades, desempenho, segurança, usabilidade e compatibilidade. Eles são tipicamente documentados em um documento de especificação

de requisitos, que serve como um guia para os desenvolvedores e uma referência para o teste e validação do software [Sommerville 2011] .

3.2.1. Requisitos Funcionais

Requisitos funcionais descrevem o comportamento específico ou funções que um sistema de software deve executar. Eles detalham as tarefas, serviços e funções que o sistema deve oferecer, como cálculos, manipulação de dados, processamento de transações e outras atividades específicas do usuário. Os requisitos funcionais são essenciais para garantir que o software atenda às necessidades e expectativas dos usuários finais e das partes interessadas. Eles são geralmente representados em forma de histórias de usuário, casos de uso ou descrições detalhadas das funções do sistema [Pressman and Maxim 2014].

ID	Descrição do Requisito	Prioridade	Complexidade
RF01	Usuário faz login como administrador ou garçom	Essencial	Alta
RF02	Usuário gerencia o produto	Essencial	Alta
RF03	Usuário registra quantidade de produtos	Essencial	Média
RF04	Sistema gera aviso de baixo estoque	Importante	Média
RF05	Usuário visualiza produtos no estoque	Essencial	Alta
RF06	Usuário gerencia a venda	Essencial	Alta
RF07	Usuário gerencia um pedido agendado	Desejável	Média
RF08	Sistema registra o histórico de vendas	Essencial	Média
RF09	Sistema filtra as vendas no histórico	Essencial	Alta
RF10	Usuário gerencia as vendas do dia no caixa	Essencial	Alta
RF11	Usuário gerencia a comanda	Essencial	Média
RF12	Sistema gera relatório de vendas do dia	Importante	Alta
RF13	Sistema gera estatísticas de venda, lucro, saída e entrada de estoque	Desejável	Alta

3.2.2. Requisitos Não Funcionais

Requisitos não funcionais especificam critérios que podem ser usados para julgar a operação de um sistema, ao invés de comportamentos específicos. Eles incluem atributos de qualidade do sistema, como desempenho, segurança, escalabilidade, usabilidade, confiabilidade e conformidade normativa. Esses requisitos são fundamentais para garantir que o sistema não apenas funcione corretamente, mas também ofereça uma experiência de uso satisfatória e segura, além de ser capaz de suportar a carga e as demandas futuras. Requisitos não funcionais são frequentemente expressos em termos de métricas ou condições que o sistema deve cumprir [International Organization for Standardization 2011].

ID	Descrição do Requisito	Prioridade	Complexidade
RN01	Usuários devem saber usar o sistema intuitivamente (usabilidade)	Essencial	Média
RN02	Sistema deve gerenciar o acesso (responsabilidade)	Essencial	Alta
RN03	Sistema deve fazer o backup (confiabilidade)	Essencial	Alta
RN04	Uso do PostgreSQL como banco de dados	Essencial	Alta
RN05	Uso de Angular Framework para o Front-end	Essencial	Alta
RN06	Uso do Sprint Boot	Essencial	Alta
RN07	Uso do Spring Security	Essencial	Alta
RN08	Uso do Sha512 para criptografia	Essencial	Alta

3.2.3. Técnicas de Elicitação de Requisitos Utilizadas

Para elicitar requisitos de forma eficaz, foram utilizadas varias técnicas, incluindo entrevistas estruturadas com os proprietarios do estabelecimento, prototipagem rápida e análise de cenários. As entrevistas estruturadas permitem obter informações detalhadas e precisas diretamente dos principais interessados, garantindo que suas necessidades e expectativas sejam claramente compreendidas.

A prototipagem rapida, por sua vez, envolve a criação de modelos preliminares do sistema, permitindo que os proprietarios visualizem e avaliem funcionalidades propostas, fornecendo feedback imediato e ajudando a refinar os requisitos. Já a análise de cenários envolve a criação de situações hipotéticas baseadas no uso do sistema, ajudando a identificar possíveis problemas e necessidades adicionais, assegurando que todos os aspectos do projeto sejam considerados.

Essas técnicas combinadas garantem uma compreensão abrangente e precisa dos requisitos do sistema

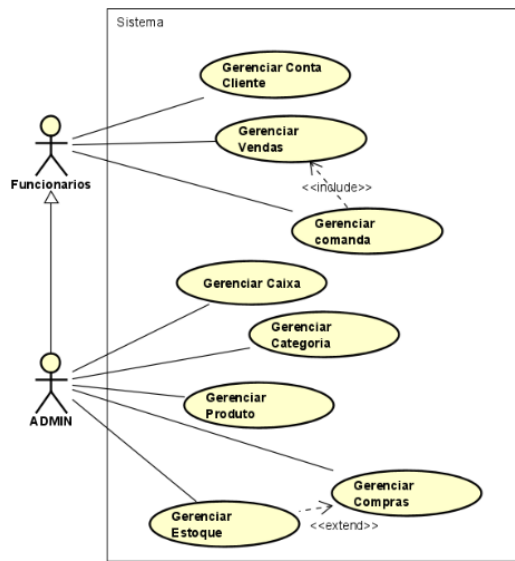


Figure 1. Diagrama de Caso de Uso

3.3. Diagrama de Caso de Uso

O diagrama de caso de uso é uma representação gráfica que descreve as interações entre os atores (usuários ou outros sistemas) e o sistema em desenvolvimento. Ele mostra as funcionalidades oferecidas pelo sistema, chamadas de casos de uso, e como os atores utilizam essas funcionalidades para atingir seus objetivos. Este diagrama é fundamental para capturar os requisitos funcionais de um sistema e para entender como os usuários interagem com ele [Fowler 2004].

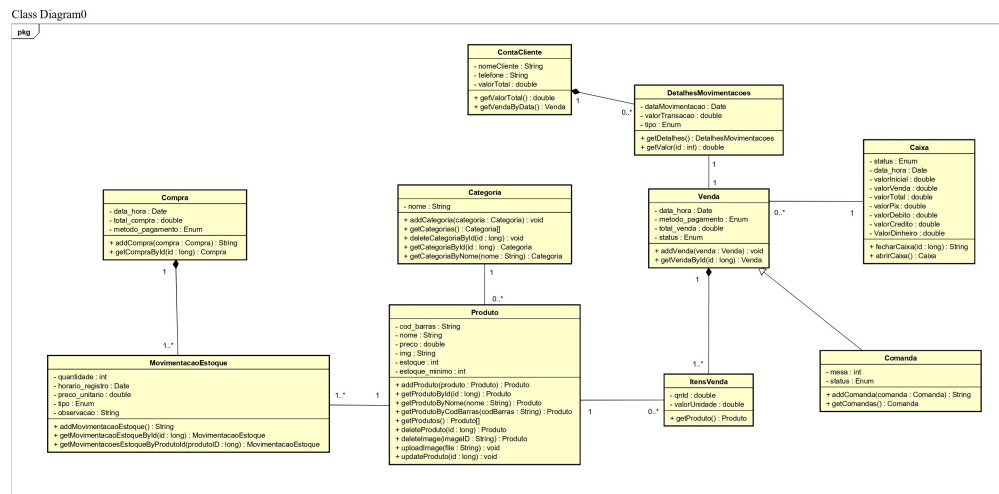


Figure 2. Diagrama de Classe

3.4. Diagrama de Classe

O diagrama de classes é uma representação gráfica que descreve a estrutura estática de um sistema, mostrando suas classes, atributos, métodos e os relacionamentos entre elas. Ele é essencial para o desenvolvimento de software orientado a objetos, pois ajuda a identificar as principais entidades do sistema e suas interações. Este diagrama é usado para modelar a arquitetura do sistema e para guiar a implementação do código [Fowler 2004].

4. Projeto da Solução

4.1. Descrição Técnica da Solução

4.1.1. Arquitetura do Software

A arquitetura de software define a estrutura fundamental de um sistema, incluindo seus componentes, suas interações e os princípios que guiam seu design e desenvolvimento. No contexto do projeto em questão, utilizamos tecnologias específicas como Angular Material, MVC (Model-View-Controller) e Spring Boot para construir uma aplicação web robusta e escalável.

4.1.2. Angular Material

Angular Material é uma biblioteca de componentes UI implementada para Angular e projetada com os princípios de Material Design. Ela oferece uma série de componentes prontos para uso, como botões, caixas de diálogo, menus, entre outros, que seguem as diretrizes de design estabelecidas pelo Google. Esses componentes ajudam os desenvolvedores a criar interfaces de usuário consistentes e agradáveis, além de facilitar a interação com os usuários através de uma experiência visual intuitiva [ang].

4.1.3. MVC (Model-View-Controller)

O padrão MVC é uma arquitetura de software que separa uma aplicação em três componentes principais: Model (Modelo), View (Visão) e Controller (Controlador). O Modelo representa os dados e a lógica de negócios, a Visão exibe a interface com o usuário e o Controlador gerencia as interações entre o Modelo e a Visão. Este padrão promove a separação de preocupações e facilita a manutenção, testabilidade e escalabilidade de aplicações [Reenskaug 1987].

4.1.4. Spring Boot

Spring Boot é um framework de código aberto para o desenvolvimento de aplicações Java, focado na simplicidade e produtividade. Ele facilita a configuração de aplicações baseadas em Spring, fornecendo funcionalidades pré-configuradas e prontas para uso, como injeção de dependência, configuração automática e um servidor embutido. Spring Boot é amplamente adotado devido à sua capacidade de acelerar o desenvolvimento e simplificar a manutenção de sistemas robustos [Wallace 2016].

4.1.5. Tecnologias Envolvidas

4.1.6. Controle de Versão com Git e GitHub

O controle de versão do código-fonte foi gerenciado utilizando Git, um sistema de controle de versão distribuído amplamente utilizado. Git permite o controle eficiente das

mudanças no código, oferecendo recursos como ramificações (branches) e merge para facilitar o trabalho colaborativo entre desenvolvedores [git a].

GitHub, por sua vez, é uma plataforma baseada em Git que facilita o gerenciamento e colaboração em projetos de software. Além de hospedar repositórios Git, o GitHub oferece ferramentas para revisão de código, rastreamento de problemas (issues), integração contínua e entrega contínua (CI/CD), entre outras funcionalidades que promovem a colaboração eficiente em equipes de desenvolvimento [git b].

4.1.7. Banco de Dados PostgreSQL

O banco de dados utilizado na aplicação é o PostgreSQL, um sistema de gerenciamento de banco de dados relacional de código aberto e altamente confiável. PostgreSQL é conhecido por sua robustez, suporte a ACID (Atomicidade, Consistência, Isolamento e Durabilidade) e extensibilidade, sendo amplamente utilizado em aplicações que requerem armazenamento seguro e escalável de dados [Momjian and et al. 2001].

4.1.8. Método de Desenvolvimento Scrum

Scrum é uma metodologia ágil de desenvolvimento de software que enfatiza a colaboração, flexibilidade e entrega contínua de valor ao cliente. Baseado em iterações curtas chamadas de sprints, Scrum promove a auto-organização de equipes multifuncionais e uma abordagem iterativa e incremental para o desenvolvimento de produtos [Schwaber and Sutherland 2012].

4.2. Planejamento das Telas

O planejamento das telas é uma etapa crucial no desenvolvimento do sistema de gestão da Distribuidora Ferreira, pois define como os usuários interagirão com a aplicação. Este processo envolve a criação de wireframes e protótipos detalhados para cada funcionalidade principal, garantindo que a interface seja intuitiva e eficiente. Considerando as necessidades específicas dos diferentes tipos de usuários, como administradores, funcionários e clientes, o design das telas busca proporcionar uma experiência de uso fluida e agradável. A escolha do framework Angular, juntamente com Angular Material, permite a construção de interfaces modernas e responsivas, assegurando que o sistema seja acessível e funcional em diferentes dispositivos e plataformas.

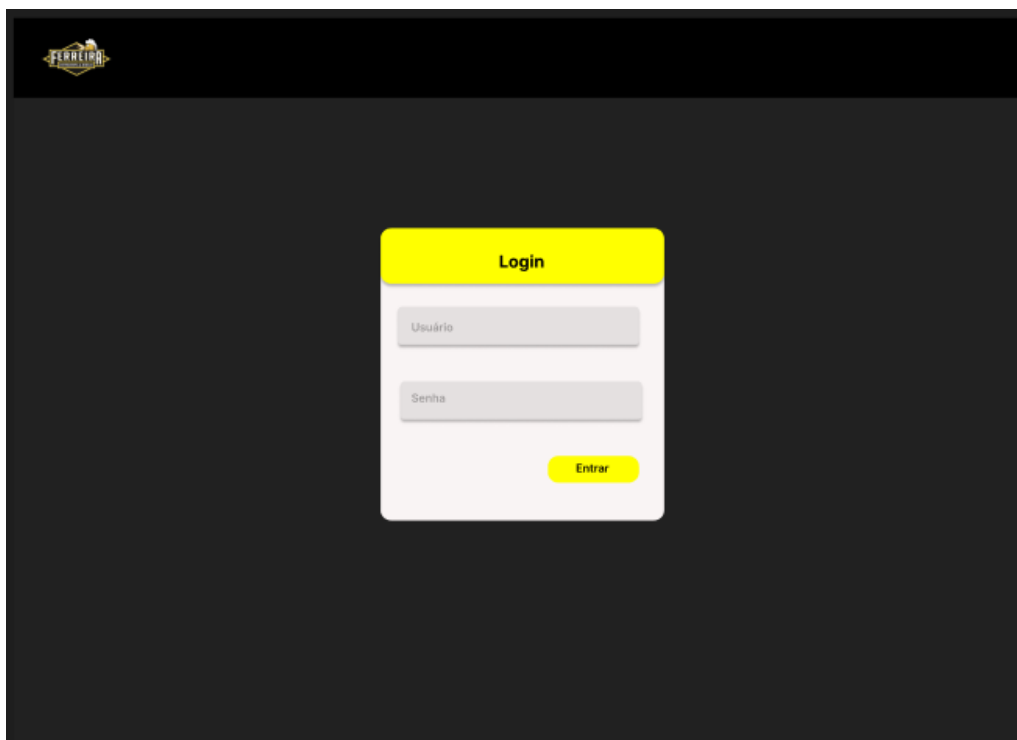


Figure 3. Login page

4.2.1. Login Page

Primeira interface que é apresentada ao usuário quando ele entra no sistema. Após um login bem sucedido, o usuário será direcionado para a Home Page. Como o sistema não é acessível pelos clientes, somente os funcionários e administradores da Distribuidora Ferreira vão ter acesso.

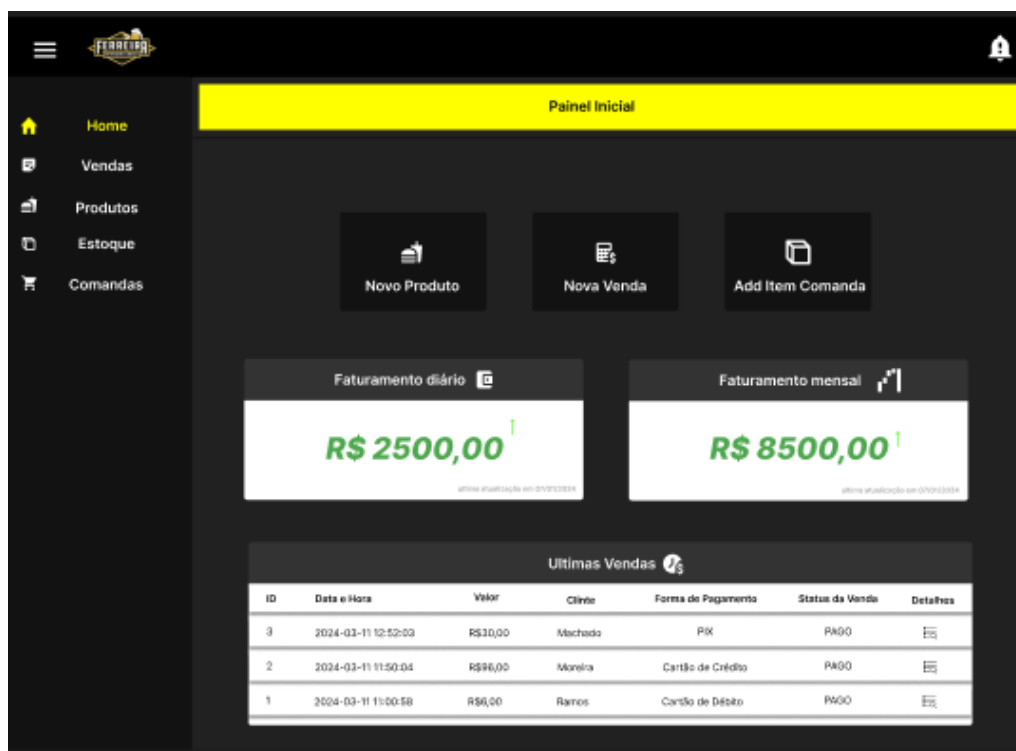


Figure 4. Home Page

4.2.2. Home Page

A página inicial da Distribuidora Ferreira é o ponto de entrada para funcionários e clientes acessarem as principais funcionalidades e informações relevantes do sistema. Projetada para ser intuitiva e informativa, esta tela exibe um resumo das últimas atualizações, como novos produtos, promoções em andamento e avisos importantes. Os funcionários podem acessar links rápidos para as páginas de comandas, estoque, vendas e relatórios diretamente da página inicial, facilitando a navegação e o acesso rápido às áreas essenciais do sistema. A interface é responsiva, adaptando-se a diferentes dispositivos para garantir uma experiência de usuário consistente e acessível.

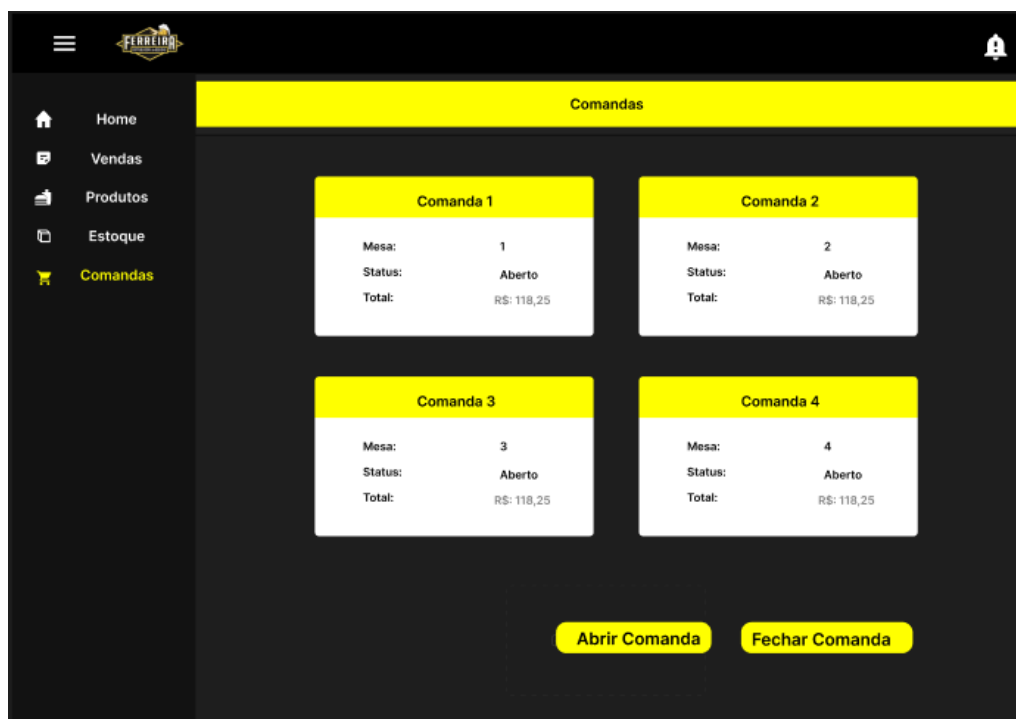


Figure 5. Página das comandas

4.2.3. Página de Comandas

A página de comandas oferece uma interface intuitiva para que os funcionários possam gerenciar as mesas e pedidos dos clientes no estabelecimento. A tela exibe todas as mesas disponíveis, permitindo que os garçons visualizem e atualizem os pedidos conforme necessário. Cada mesa é representada visualmente, indicando o status do pedido (aberto, em andamento, fechado) e detalhes como os itens pedidos, quantidades e status de preparação. Os botões e controles são projetados para facilitar a adição, modificação ou exclusão de itens do pedido, garantindo uma interação rápida e eficiente. Além disso, a tela inclui recursos para fechar o pedido, calcular a conta final e processar o pagamento de forma integrada.

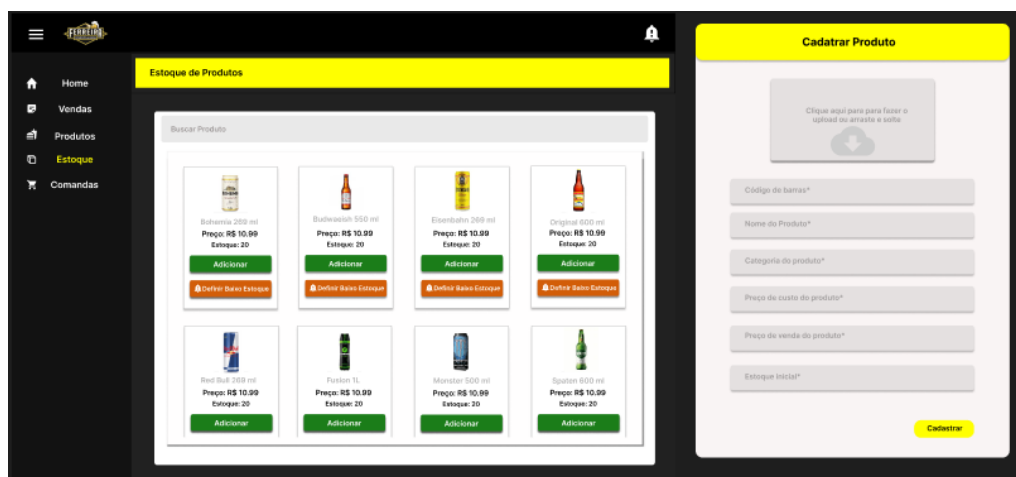


Figure 6. Página de estoque

4.2.4. Página de Estoque

A página de estoque é o centro de controle para gerenciar todos os produtos disponíveis na Distribuidora Ferreira. Nesta tela, os administradores e gerentes podem visualizar um resumo completo do inventário atual, incluindo quantidades disponíveis, níveis mínimos e máximos, fornecedores e custos associados. Cada produto é apresentado com detalhes específicos, como código de identificação, descrição, categoria e preço unitário. A interface permite adicionar novos produtos, atualizar informações existentes e ajustar os níveis de estoque conforme as necessidades operacionais.

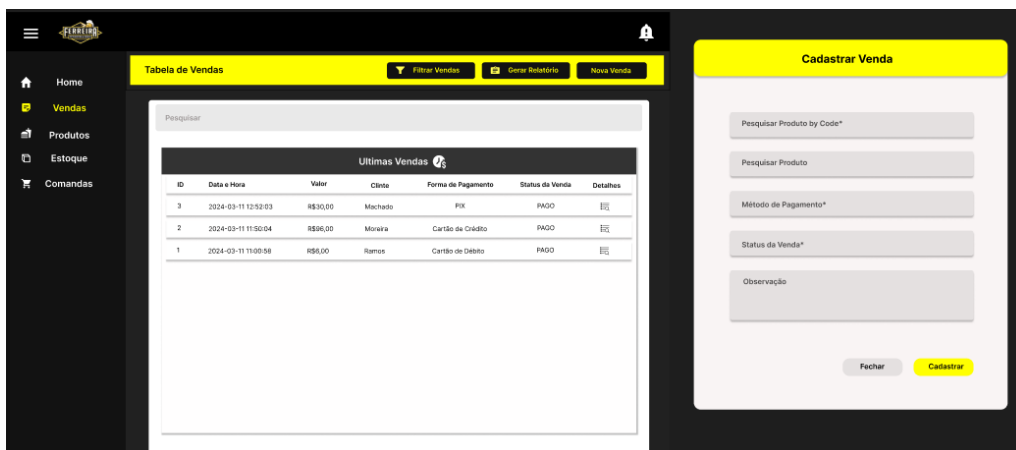


Figure 7. Página de Vendas

4.2.5. Página de Vendas

A página de vendas oferece uma visão abrangente das transações comerciais realizadas pela Distribuidora Ferreira. Nesta tela, os usuários podem acompanhar em tempo real as vendas diárias, semanais ou mensais, visualizar pedidos pendentes, histórico de vendas por cliente e produtos mais vendidos. A interface é projetada para facilitar o processo de registro de novas vendas, com opções para adicionar itens ao carrinho de compras, aplicar descontos, selecionar métodos de pagamento e gerar faturas ou recibos para os clientes. Além disso, a página inclui ferramentas para análise de desempenho de vendas, permitindo identificar padrões de consumo, calcular margens de lucro e ajustar estratégias de marketing com base nos dados coletados.

5. Artefatos Principais

Os principais artefatos criados para solucionar os problemas de gerenciamento de vendas e estoque da Distribuidora Ferreira incluem diversas ferramentas e documentações essenciais. A seguir, uma descrição detalhada dos artefatos desenvolvidos:

1. Software de Gestão Integrado:

- **Página de Login:** Interface inicial para autenticação dos usuários, garantindo a segurança e controle de acesso ao sistema;
- **Página Inicial (Home Page):** Dashboard central que apresenta um resumo das principais métricas e informações relevantes, como status do estoque, vendas recentes e notificações importante;
- **Página de Comandas:** Interface intuitiva para gerenciar mesas e pedidos de clientes no local. Permite visualizar o status dos pedidos, adicionar, modificar e excluir itens, fechar pedidos e processar pagamentos;
- **Página de Estoque:** Central de controle para gerenciamento de produtos. Apresenta detalhes como quantidades disponíveis, níveis mínimos e máximos, fornecedores e custos. Permite adicionar novos produtos, atualizar informações e ajustar níveis de estoque;
- **Página de Vendas:** Proporciona uma visão abrangente das transações comerciais, permitindo acompanhar vendas em tempo real, visualizar pedidos pendentes, histórico de vendas por cliente e produtos mais vendidos. Facilita o registro de novas vendas, aplicação de descontos, seleção de métodos de pagamento e geração de faturas

2. Especificações de Requisitos:

- **Requisitos Funcionais:** Detalham as funcionalidades que o sistema deve oferecer, como o gerenciamento de estoque, registro de vendas, e controle de comandas;
- **Requisitos Não Funcionais:** Incluem aspectos como desempenho, usabilidade, segurança e compatibilidade, garantindo que o sistema atenda a padrões de qualidade e eficiência;

3. Documentação sobre o desenvolvimento:

- **Arquitetura do Software:** Descrição técnica da solução, incluindo a estrutura do sistema, componentes principais, e a integração entre eles. Detalha a utilização de tecnologias como Angular para o front-end, Spring Boot para o back-end e PostgreSQL como banco de dados
- **Controle de Versão com Git e GitHub:** Práticas de versionamento e colaboração utilizando Git e GitHub, garantindo que todas as alterações no código sejam rastreáveis e gerenciáveis;
- **Metodologia de Desenvolvimento Scrum:** Abordagem ágil adotada para o desenvolvimento do projeto, permitindo entregas incrementais e feedback contínuo dos stakeholders.

Esses artefatos foram cuidadosamente desenvolvidos para criar uma solução robusta, eficiente e escalável, que aborda os principais problemas de gerenciamento de vendas e estoque enfrentados pela Distribuidora Ferreira. A implementação dessa solução não apenas reduz erros e aumenta a eficiência, mas também melhora a satisfação do cliente e posiciona a empresa para enfrentar novos desafios e oportunidades de crescimento.

6. Conclusão

Este trabalho apresentou uma solução abrangente para a gestão da Distribuidora Ferreira, abordando os principais problemas de gerenciamento de vendas e estoque por meio de um sistema automatizado e integrado. A implementação de tecnologias modernas, como Angular e Spring Boot, aliada à metodologia ágil e práticas DevOps, permitiu a entrega de um sistema robusto, eficiente e escalável.

6.1. Principais Resultados e Contribuições

- **Redução de Erros Humanos:** A automação de processos diminuiu a propensão a erros na entrada de dados.
- **Aumento de Eficiência:** A eliminação de tarefas manuais redundantes permitiu que os funcionários se concentrassem em atividades de maior valor.
- **Melhoria na Satisfação do Cliente:** A otimização do gerenciamento de estoque e vendas resultou em um serviço mais rápido e confiável.
- **Controle Financeiro Acurado:** A integração de funções financeiras facilitou o controle e a análise dos resultados operacionais.

6.2. Lições Aprendidas

- **Importância da Comunicação:** Manter uma comunicação contínua com os stakeholders foi essencial para alinhar expectativas e ajustar o desenvolvimento conforme necessário.
- **Flexibilidade na Implementação:** A adoção de práticas ágeis permitiu responder rapidamente a mudanças de requisitos e prioridades.
- **Valorização da Experiência do Usuário:** Focar na usabilidade e na experiência do usuário final resultou em um sistema mais intuitivo e eficaz.

A solução desenvolvida posiciona a Distribuidora Ferreira para enfrentar desafios futuros e aproveitar oportunidades de crescimento, consolidando-a como uma empresa moderna e eficiente.

7. Referências

References

- Angular material. <https://material.angular.io/>. Acessado em: 17 de junho de 2024.
- Git - fast version control system. <https://git-scm.com/>. Acesso em: 17 de junho de 2024.
- Github. <https://github.com/>. Acessado em 17 de junho de 2024.
- Fowler, M. (2004). *UML Distilled: A Brief Guide to the Standard Object Modeling Language*. Addison-Wesley Professional, 3rd edition.
- International Organization for Standardization (2011). Iso/iec 25010:2011. systems and software engineering – systems and software quality requirements and evaluation (square) – system and software quality models.
- Momjian, B. and et al. (2001). *PostgreSQL: Introduction and Concepts*. Addison-Wesley Professional.
- Pressman, R. S. and Maxim, B. R. (2014). *Software Engineering: A Practitioner's Approach*. McGraw-Hill, 8 edition.
- Reenskaug, T. (1987). *Applications Programming in Smalltalk-80(TM): How to use Model-View-Controller (MVC)*. Prentice Hall, New Jersey, USA.
- Schwaber, K. and Sutherland, J. (2012). *Software in 30 Days: How Agile Managers Beat the Odds, Delight Their Customers, And Leave Competitors In the Dust*. Wiley.
- Sommerville, I. (2011). *Software Engineering*. Addison-Wesley, 9 edition.
- Wallace, C. (2016). *Spring Boot in Action*. Manning Publications, New York, USA.