

SISTEMA DE GESTÃO DE PEDIDOS PARA UM RESTAURANTE

YGOR SOUSA MACHADO^{1 15}HEITOR RAMOS NOGUEIRA²

RESUMO

Este artigo apresenta o desenvolvimento de um sistema de gestão de pedidos para restaurantes, com o objetivo de aprimorar o processo de gerenciamento desde a efetivação até a entrega dos pedidos. A aplicação proporciona interatividade entre funcionários e clientes, permitindo que os usuários façam pedidos com confiança e acompanhem o status em tempo real através de notificações ativas. Os administradores dispõem de ferramentas de controle para gerenciamento de estoque, finanças e análise de desempenho, facilitando a tomada de decisões estratégicas. Funcionalidades adicionais incluem personalização de pedidos, opções variadas de pagamento e geração de relatórios detalhados. A metodologia adotada compreende o levantamento de requisitos, a arquitetura de design, codificação e testes de implantação. Os resultados obtidos indicam uma melhoria significativa na eficiência operacional, redução de erros nos pedidos e aumento da satisfação dos clientes. Conclui-se que o sistema desenvolvido oferece uma solução robusta e escalável, capaz de atender às demandas específicas do setor de restaurantes, promovendo uma gestão mais eficaz e segura dos processos envolvidos.

Palavras-chave: Controle. Eficiência. Restaurante. Gestão de estoque. Sistema.

ABSTRACT

¹ Graduando em [ADS] pela Fatec Dr Thomaz Novelino – Franca/SP. Endereço eletrônico: [\[ygor.machado@fatec.sp.gov.br\]](mailto:ygor.machado@fatec.sp.gov.br).

² Graduando em [ADS] pela Fatec Dr Thomaz Novelino – Franca/SP. Endereço eletrônico: [\[heitor.nogueira2@fatec.sp.gov.br\]](mailto:heitor.nogueira2@fatec.sp.gov.br).

This article presents the development of an order management system for restaurants, aimed at enhancing the process from order placement to delivery. The application provides interactivity between staff and customers, allowing users to place orders with confidence and track their status in real time through active notifications. Administrators have access to control tools for inventory management, finances, and performance analysis, facilitating strategic decision-making. Additional features include order customization, diverse payment options, and detailed report generation. The adopted methodology includes requirements gathering, design architecture, coding, and deployment testing. The results obtained indicate a significant improvement in operational efficiency, a reduction in order errors, and an increase in customer satisfaction. It is concluded that the developed system offers a robust and scalable solution capable of meeting the specific demands of the restaurant sector, promoting more effective and secure management of the processes involved.

Keywords: Efficiency. Order control. Restaurant. Stock management. System.

1 Introdução

A indústria da restauração enfrenta desafios crescentes na gestão de pedidos, nomeadamente devido à necessidade de processos mais rápidos e eficientes. Questões como erros em pedidos, atrasos nas entregas e falta de comunicação eficaz com os clientes impactaram negativamente a experiência do consumidor e a eficiência operacional. À medida que a competitividade aumenta e os serviços se tornam digitais, é crucial que os restaurantes adotem soluções tecnológicas que simplifiquem e otimizem a gestão de pedidos.

Pesquisas recentes mostram que a implementação de um sistema de gestão digital pode reduzir significativamente o tempo de processamento de pedidos e melhorar a precisão, aumentando assim a satisfação e a fidelidade do cliente. No entanto, muitas soluções existentes carecem de integração abrangente entre capacidades de gestão de inventário, diversas opções de pagamento e relatórios detalhados.

Este projeto visa preencher essas lacunas, fornecendo uma abordagem mais robusta e escalável.

O principal objetivo do projeto era desenvolver um site que permitisse uma gestão eficiente dos pedidos, proporcionasse uma melhor experiência aos clientes e ajudasse a aumentar as vendas. Adicionalmente, o projeto visa criar uma plataforma que agilize o funcionamento interno do restaurante e fortaleça a sua presença online. A justificativa para a implementação do sistema foi a necessidade de aumentar o reconhecimento da marca, melhorar a eficiência operacional e proporcionar uma experiência de usuário mais satisfatória.

Para atingir esses objetivos, será utilizada uma abordagem baseada na coleta de requisitos, no desenho da arquitetura, na codificação e na implementação de testes.

A solução proposta visa não só facilitar a gestão de encomendas, mas também proporcionar canais de comunicação eficazes e aumentar a fidelização dos

clientes, com especial enfoque nos públicos mais jovens e mais propensos a utilizar plataformas digitais para realizar encomendas.

2 Referencial teórico e trabalhos correlatos

Atuando no ramo alimentício, a empresa cliente fornece produtos e serviços de alta qualidade para satisfazer seus clientes.

A empresa pretende ser reconhecida como uma referência em inovação, sustentabilidade e transparência na indústria de alimentos, valorizando princípios como orientação para resultados e ética em tudo o que faz.

Seu objetivo é continuar crescendo como uma empresa de sucesso no setor alimentício.

2.1 Termo de Abertura do Projeto [TAP]

O Termo de Abertura do Projeto (TAP) é um documento fundamental que formaliza o início formal de um projeto. Ele contém informações essenciais e detalhes importantes sobre o projeto, estabelecendo as bases para todo o seu desenvolvimento, (PRESSMAN, 2016).

No TAP é estabelecido quem é o gerente, o histórico da empresa para quem será desenvolvido o projeto, o objetivo deste projeto e sua justificativa, além de uma descrição detalhada, as funcionalidades, riscos, custos e prazos estimados. Este documento representa um acordo inicial entre as partes interessadas, definindo critérios de sucesso e as bases para a condução e controle do projeto.

O TAP apresenta os seguintes tópicos, (WAZLAWEICK, 2023):

Objetivo: Criação de um site para gerenciamento de pedidos e fidelização de clientes.

Justificativa: Criar um site para melhor visibilidade da marca da empresa e oferecer uma melhor experiência ao cliente, além de tudo, aumentar as vendas.

Descrição: União das forças, fraquezas, oportunidades, ameaças, viabilidade e todos os processos incluídos ao longo do projeto para um melhor desempenho e entendimento das dificuldades.

Funcionalidades: O site terá como funcionalidades o gerenciamento de pedidos, visando promover uma interface de fácil uso para o usuário.

Riscos: A concorrência e a precificação dos produtos.

2.2 Estrutura Analítica do Projeto (EAP)

A EAP, ou Estrutura Analítica do Projeto, é uma ferramenta de gerenciamento de projetos que desmembra um projeto em partes menores e mais gerenciáveis. Ela fornece uma visão hierárquica e estruturada do escopo do projeto, identificando e organizando os principais elementos do trabalho a ser realizado, (ARTIA, 2023).

Representada graficamente em forma de árvore, com a "raiz" sendo o projeto em si e os "galhos" representando as principais fases, entregas ou componentes do projeto, (ARTIA, 2023).

Cada ramificação subsequente detalha as subdivisões desses elementos em partes menores e mais específicas. Cada nível na hierarquia da EAP representa uma camada de detalhe crescente importante para: Decomposição do Projeto; Atribuição de Responsabilidades; Estimativa de Custos e Recursos e Controle de Mudanças.

Este sistema de gerenciamento de pedidos de um restaurante é fundamental para otimizar processos, melhorar o serviço e aumentar a eficiência, (ALMEIDA, 2023).

Baseados em conceitos de sistemas de informação, esses sistemas permitem a integração dos pedidos com outras áreas como estoque e finanças, agilizando os serviços e minimizando erros operacionais.

De acordo com Silva (2015), “os pedidos digitais podem agilizar os tempos de atendimento e reduzir custos operacionais”.

“A tecnologia de autoatendimento pode proporcionar aos clientes uma experiência mais prática e ágil, aumentando assim a satisfação com os gastos”, como mostrado por Oliveira e Brito (2017).

Santos e Almeida (2019), mostram “que a integração de sistemas especialistas e soluções em nuvem pode ajudar a melhorar o controle de estoque e o fluxo de caixa, aumentando assim a eficiência operacional”.

3 Material e Métodos ou desenvolvimento

O desenvolvimento deste projeto foi realizado utilizando diversas tecnologias robustas e modernas, com foco em eficiência e qualidade.

O PHP serviu como a linguagem de programação backend, proporcionando a base para a construção de funcionalidades dinâmicas.

Optamos pelo Laravel devido à sua abordagem moderna e organizada. Ele oferece recursos como sistemas de rotas intuitivo, ORM Eloquent para manipulação de banco de dados, e um sistema muito bem estruturado em torno do padrão MVC, além de acelerar o desenvolvimento por conta das várias ferramentas que ele oferece.

Blade foi a ferramenta escolhida para a construção das views, por ser nativamente integrado ao Laravel, com uma sintaxe simples e também podendo criar interfaces dinâmicas e organizar o código através de componentes.

No front-end, o Tailwind CSS foi essencial para o design da interface, por estarmos trabalhando com Blade Components, o tailwind foi a melhor escolha para estilizar os componentes e manter o código organizado.

Para o versionamento do código, utilizamos o git, com o repositório do projeto hospedado no GitHub, facilitando o controle das alterações

feitas no projeto. Como ambiente de desenvolvimento, utilizamos o PHPStorm como IDE, que nos proporcionou ferramentas avançadas para depuração e escrita de um código eficiente. Para os testes trabalhamos com um ambiente local para simular o de produção.

O processo de desenvolvimento de um projeto é uma jornada estruturada que envolve várias etapas, desde a concepção da ideia até a entrega do produto. Cada fase do desenvolvimento requer a criação de artefatos específicos, contribuindo para a compreensão, documentação e implementação eficaz do projeto.

No início do processo, a concepção da ideia do projeto é fundamental. Nessa fase, é elaborado um documento inicial que descreve a visão geral do projeto, seus objetivos, requisitos preliminares e o escopo geral. Este artefato, muitas vezes chamado de "Documento de Visão", estabelece as bases para o desenvolvimento subsequente.

Uma etapa crucial é o levantamento de requisitos, onde os detalhes funcionais e não funcionais do projeto são identificados. O artefato gerado nessa fase é o "Documento de Requisitos", que serve como guia para o desenvolvimento, descrevendo as características e funcionalidades esperadas do produto.

Com os requisitos, a fase de análise aprofunda as especificações do projeto. Modelos conceituais, como diagramas de caso de uso, diagramas de classe e diagramas de sequência, são desenvolvidos para representar visualmente a estrutura e o comportamento do sistema. Estes são artefatos cruciais para a compreensão do design do projeto.

A codificação é a fase em que o código-fonte é desenvolvido com base nos requisitos e no design previamente estabelecidos. O artefato principal aqui é o próprio código-fonte, organizado de maneira lógica e eficiente para garantir a implementação adequada das funcionalidades do projeto.

Os testes são realizados para validar o funcionamento do sistema em relação aos requisitos estabelecidos. Isso envolve a criação de "Casos de Teste" que verificam se o sistema atende às expectativas. Os resultados dos testes são documentados em relatórios, indicando quaisquer problemas identificados e suas soluções.

Após a fase de testes bem-sucedida, o projeto está pronto para ser implantado no ambiente de produção. Um "Plano de Implantação" é elaborado, detalhando os passos necessários para instalar e configurar o sistema. As "Instruções de Instalação" também são desenvolvidas para orientar os usuários na implementação.

A fase de manutenção inicia-se após a implantação, onde relatórios de erros e atualizações são gerenciados. Artefatos como "Relatórios de Erros" documentam problemas encontrados, enquanto "Atualizações de Software" descrevem as modificações e melhorias implementadas ao longo do tempo.

As figuras apresentadas mostram a arquitetura e design do front-end e back-end do sistema de gerenciamento. O front-end contém a interface gráfica para interação do usuário, destacando elementos de usabilidade e navegação, enquanto o back-end cuida do processamento de dados, lógica de negócios e integração de banco de dados.

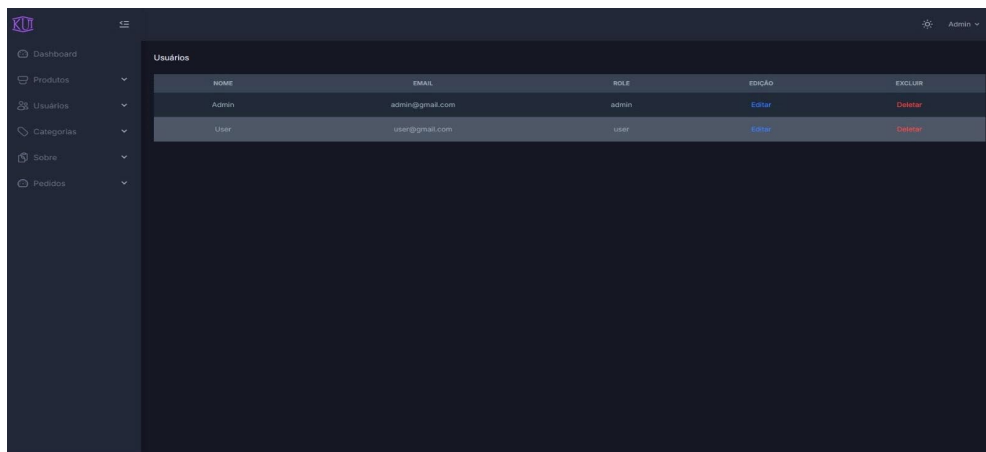
Essas representações visuais são essenciais para entender como os dois lados do sistema funcionam juntos para garantir uma experiência tranquila e eficiente para o usuário.

Durante o desenvolvimento, enfrentamos os seguintes desafios abaixo:

Integração com o banco de dados, para evitar inconsistências em transações críticas como pagamentos, optamos por escolher o método DB::Transaction do próprio Laravel.

Segurança, implementamos a criptografia das senhas para manter os dados do usuário seguros, além de criar várias validações utilizando o FormRequest do Laravel.

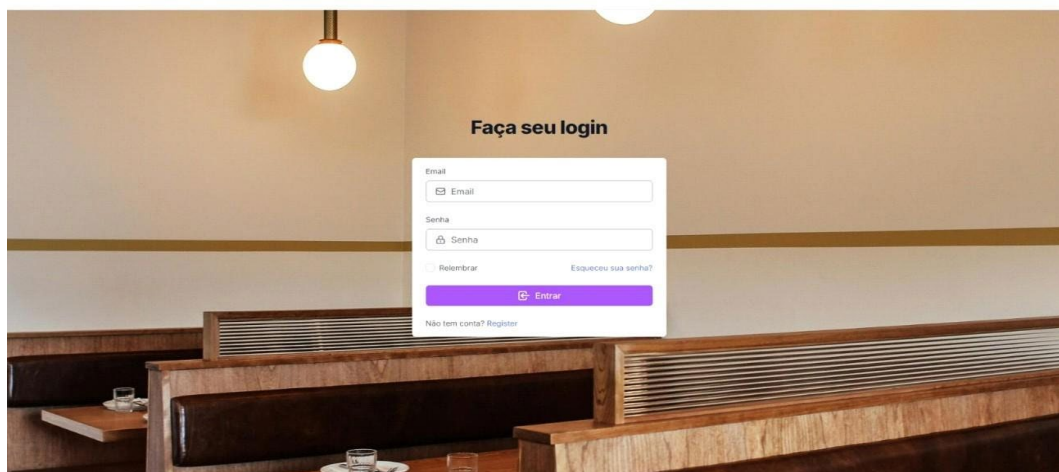
A figura 1 exibe a interface da administração do sistema, especificamente a seção de "Usuários", onde é possível visualizar e gerenciar os usuários cadastrados. A tabela apresenta três colunas principais: Nome, Email e Role (Função), além das colunas para editar e excluir os usuários, permitindo realizar as operações de CRUD (Create, Read, Update, Delete).



Fonte: Autores

A figura 2 exibe a interface da página de login do site, mostrando também o menu de navegação para facilitar a navegação entre as páginas do site, no formulário tem o botão “esqueceu a senha” para o caso de o usuário esquecer sua senha, e para caso não tenha o login tem o botão para ser redirecionado para a página de registrar.

Figura
2:
Front-end



Fonte:

Autores

A figura 3 é uma função responsável por criar um usuário no sistema. Primeiro ele valida os dados recebidos através de um FormRequest chamado StoreUserRequest. Após a validação, a senha é criptografada usando o método Hash::make(), garantindo que ela seja armazenada de forma segura no banco de dados. Em seguida o usuário é criado com os dados fornecidos no método create()

da classe User. Após a criação, o sistema redireciona para uma página de listagem de usuários e exibe uma mensagem de sucesso.

Figura

3:

back-end

```
Ygor-Machado
public function store(StoreUserRequest $request)
{
    $data = $request->validated();

    $data['password'] = Hash::make($data['password']);

    User::create($data);

    return redirect()->route('route: 'users.index')->with('success', 'Usuário criado com sucesso!');
}
```

Fonte: Autores

A figura 4 define um método `getOrdersByDate` passando uma variável como parâmetro, responsável por buscar pedidos finalizados em uma data específica, convertendo a data em um objeto Carbon, depois é filtrado por data e que tem os status finalizado, retornando os pedidos que atendem esse critério.

Figura

4:

back-end

```
public function getOrdersByDate($date)
{
    $carbonDate = Carbon::parse($date);

    return Order::whereDate('created_at', $carbonDate)
        ->where('status', OrderStatusEnum::FINALIZADO)
        ->get();
}
```

Fonte:

Autores

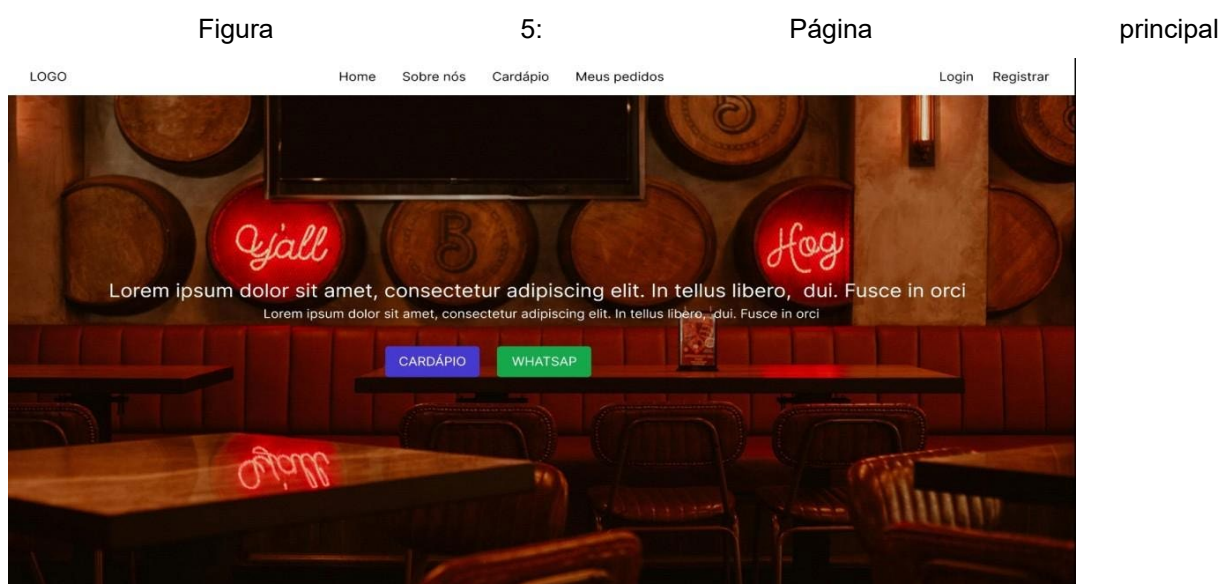
4 Resultados e discussões

A implementação do sistema de controle de pedidos do restaurante resultou em uma interface intuitiva e eficiente. O sistema foi projetado para otimizar o fluxo de trabalho e facilitar o gerenciamento de pedidos para funcionários e clientes. As figuras 5, 6 e 7 ilustram os principais recursos, como seleção de itens nos menus, monitoramento do status do pedido.

As figuras 5, 6 e 7 apresentam algumas das principais interações, destacando um layout simplificado para otimizar o fluxo de trabalho do agente e do

cliente. É fundamental para visualizar e ajustar os elementos da interface para garantir que o sistema atenda às necessidades operacionais do restaurante de forma prática e eficiente.

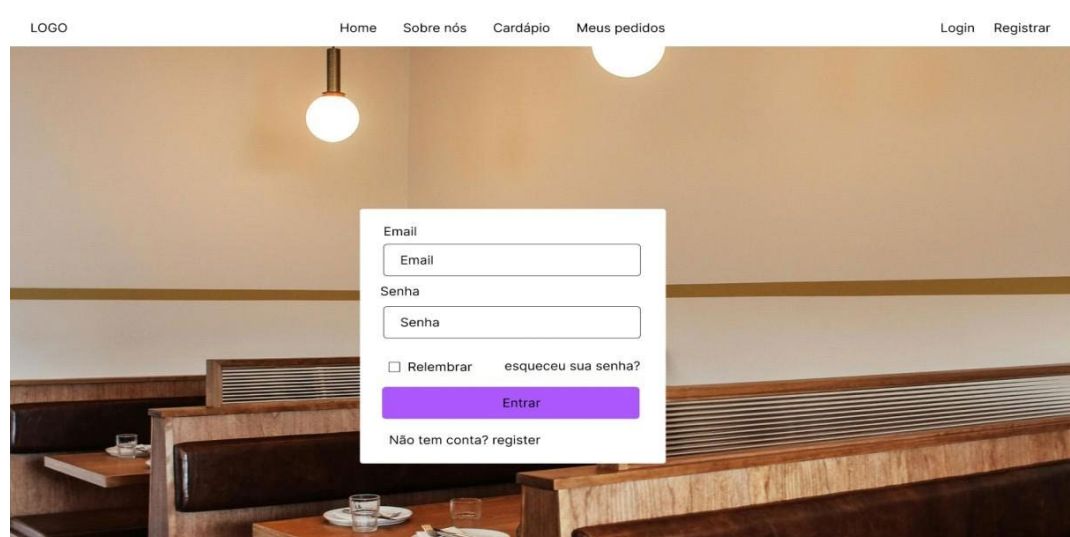
A figura 5 apresenta a 1ª seção do site do restaurante com layout moderno. Há um menu na parte superior com as opções, seguido por uma grande imagem de fundo mostrando o interior de um restaurante com mesas de madeira e cadeiras de couro e baldes decorativos na parede.



Fonte: Autores

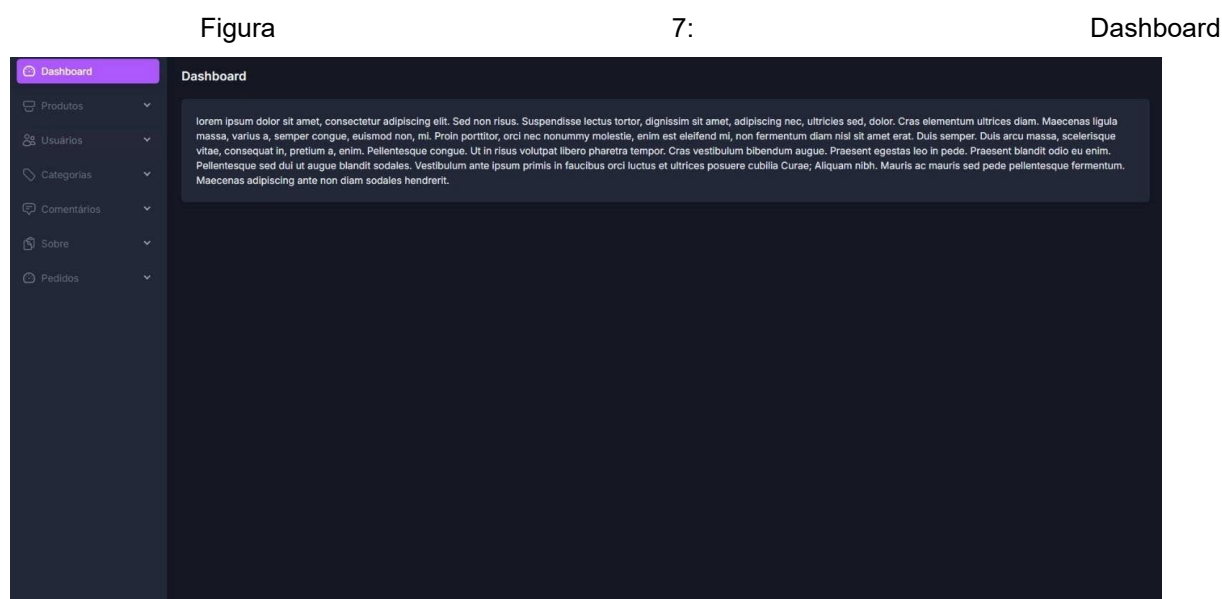
A figura 6 mostra uma página de login de um site com um layout simples e intuitivo. Há um menu na parte superior e botões Login e Cadastro no canto superior direito. Há um formulário de login no centro da tela com campos de e-mail e senha, uma opção “Lembrar” e um link “Esqueceu sua senha?”. Há um botão roxo abaixo do formulário que contém a palavra “Login” e um link para registrar uma nova conta.

Ao fundo, a figura mostra a área da mesa de um restaurante com iluminação suave.



Fonte: Autores

A figura 7 apresenta a interface central onde o administrador tem o controle total sobre as operações do sistema. A interface está organizada com um menu lateral à esquerda, que oferece fácil acesso às diferentes seções, como produtos, usuários, categorias, comentários e pedidos. Cada uma dessas seções permite ao administrador gerenciar os respectivos itens, possibilitando ações como visualizar, adicionar, remover e atualizar as informações disponíveis no sistema.



Fonte:

Autores

Considerações finais

No final do projeto foi necessário revisitar os objetivos que traçamos no início para avaliar o quanto conseguimos alcançar e medir o sucesso dos objetivos declarados. Desde o início, o nosso principal objetivo foi criar uma solução eficiente e prática para a gestão de restaurantes, e ao longo do processo de desenvolvimento atingimos marcos importantes, desde a integração do sistema com a cozinha até à implementação de uma área de feedback para os clientes.

Durante a execução do projeto, focamos no desenvolvimento de uma plataforma intuitiva para os usuários, permitindo pedidos customizados, controle de estoque em tempo real e geração automática de relatórios para administradores.

Essas funcionalidades não só atendem às necessidades originais da empresa, mas também agregam valor ao negócio, proporcionando aos clientes melhor experiência e maior eficiência operacional.

Contudo, neste processo, também encontramos desafios. Esses obstáculos exigiram um esforço extra de nossa parte e, apesar das dificuldades, geraram valiosas lições aprendidas e oportunidades de crescimento e melhoria que, sem dúvida, nos ajudarão em projetos futuros.

No futuro, existem oportunidades para expandir a solução, como a inclusão de um sistema de fidelização mais robusto, o fornecimento de integração com aplicações de entrega externas e a implementação de análises preditivas de vendas com base no comportamento do cliente. Além disso, acompanhar as tendências tecnológicas e as preferências do consumidor é fundamental para impulsionar a melhoria contínua.

Os desafios tornam-se oportunidades de aprendizagem e as conquistas nos inspiram a continuar explorando novas possibilidades. Estamos entusiasmados com os próximos passos e prontos para continuar inovando e agregando mais valor às soluções que desenvolvemos.

Referências bibliográficas

ARTIA. Como fazer EAP na gestão de projetos. Artia, São Paulo, 2023. Disponível em: <https://artia.com/blog/como-fazer-eap-na-gestao-de-projetos/>. Acesso em: 05 set. 2023.

ALMEIDA, Vinicius Nóbile. A importância da gestão de dados nos processos de negócios das organizações modernas. 2023. 150 folhas. Dissertação

(Mestrado em Administração) - Faculdade de Administração, Universidade de São Paulo, 2023.

OLIVEIRA, Marcos; BRITO, Andréa. A tecnologia no autoatendimento de restaurantes: experiência do cliente e agilidade no serviço. São Paulo: Saraiva, 2017. 240 p.

PRESSMAN, Roger S. Engenharia de software - produtos - Vol. 1: Volume 1. 8. ed. Porto Alegre: AMGH, 2016. 752 p.

SANTOS, Rodrigo; ALMEIDA, Fernanda. Sistemas POS e a automação de pedidos no setor de alimentos e bebidas. Brasília: Universidade de Brasília, 2019. 200 p.

SILVA, João Carlos. Sistemas de informação em restaurantes: impactos na eficiência e redução de custos operacionais. Rio de Janeiro: Editora FGV, 2015. 182 p.

WAZLAWEICK, Raul. Engenharia de software: fundamentos e práticas. 11. ed. São Paulo: Editora XYZ, 2023. p. 137-145.