

Desenvolvimento e funcionalidades do Sistema Web “Caketter” para Pequenos Empreendimentos no setor de Confeitaria

Ádrian Rabelo Mendes¹, Cledenilson Souza Martins²

Instituto Federal de Educação, Ciência e Tecnologia de Rondônia (IFRO)
76820-441 – Porto Velho – RO – Brasil

adrianrabelo10@gmail.com¹, cledenilson.martins@ifro.edu.br²

Abstract. *This paper aims to present the solution developed at the conclusion of research that sought to leverage technology as a means of improving the organization and fulfillment of orders in a small bakery, with the goal of strengthening local production arrangements. The process of creating the system, named “Caketter,” was divided into phases. In the first phase, a requirements analysis was conducted; due to time constraints, this analysis was limited to order management. Following this assessment, the system’s functionalities were defined. Using agile methodologies, iterative and incremental development cycles were established, allowing for rapid implementation of various parts of the system. Modern technologies and frameworks were used for the user interface, and robust programming languages were employed to develop the system logic and integrate with the database. At the end of the development phase, the system was tested by the developer to ensure its stability, security, and usability. The results demonstrated that the “Caketter” system is an effective and intuitive tool to assist bakery professionals in managing their orders, contributing to the optimization of processes and improvement in the quality of services provided.*

Resumo. *Este artigo tem por objetivo apresentar a solução desenvolvida ao final da pesquisa que buscou utilizar a tecnologia como forma de colaborar com a organização e melhoria do atendimento a pedidos de uma pequena confeitaria, com vistas ao fortalecimento dos arranjos produtivos locais. O processo para criação do sistema denominado “Caketter”, foi dividido em fases na primeira foi feita a análise de requisitos, em virtude dos prazos, delimitou-se essa análise a gestão de pedidos. Após esse levantamento definiu-se as funcionalidades. Utilizando metodologias ágeis, foram definidos ciclos de desenvolvimento iterativos e incrementais, o que permite uma rápida implementação de partes do sistema. Utilizou-se tecnologias e frameworks modernos para a interface do usuário, e linguagens de programação robustas, para o desenvolvimento da lógica do sistema e integração com banco de dados. Ao final da fase de desenvolvimento, o sistema foi testado pelo desenvolvedor para garantir sua estabilidade, segurança e usabilidade. Os resultados obtidos demonstraram que o sistema “Caketter” é uma ferramenta eficaz e intuitiva para auxiliar profissionais da área de confeitaria na gestão de seus pedidos, contribuindo para a otimização de processos e melhoria da qualidade dos serviços oferecidos.*

1. Introdução

É notório que a tecnologia vem evoluindo com o passar dos anos, facilitando a vida e tornando trabalhos e atividades diárias mais eficientes. Segundo o Instituto de Pesquisa Econômica Aplicada (IPEA), um dos principais motivos do avanço da ciência é a curiosidade humana, que não se compromete com resultados concretos. Ainda segundo o IPEA, outro motor do avanço científico é a solução de problemas que afligem a humanidade, viver com mais tempo e saúde, trabalhar menos são desafios e aspirações humanas que a ciência e a tecnologia têm contribuído.

Uma das áreas que mais se beneficia com esses avanços tecnológicos é a área do empreendedorismo, que aproveita de sistemas de baixo custo ou gratuitos para alavancarem seu negócio e empreendimento, especialmente em relação à gestão de pedidos na área de gastronomia, como pode ser visto em pequenos restaurantes e lanchonetes. Conforme descrito pela Delage (2024), os sistemas existentes frequentemente seguem um padrão voltado para pedidos imediatos e com categorias de pedido padrão. No entanto, áreas como a confeitaria, que operam de maneira semelhante a um consultório, onde os pedidos requerem um agendamento prévio, podendo serem marcados para entrega em dias posteriores.

Pessoas que perderam o emprego devido à pandemia provocada pelo Covid-19 buscaram um meio de trabalhar em casa. Diante disso, de acordo com Torta (2021) a área da confeitaria teve uma crescente de 200% visto que é um trabalho que pode ser realizado em casa, mas ainda exige tempo para preparação dos pedidos. Porém, são escassos os sistemas que auxiliem os autônomos dessa área, fazendo com que eles utilizem papel e caneta para organizar seus pedidos, gerando custos de produção de papéis personalizados onde incluem características dos pedidos como nome do cliente, data de entrega e demais informações necessárias.

Com foco nesta área de atuação, este trabalho tem como objetivo a implementação de um sistema web gratuito capaz de facilitar a gestão de pedidos de profissionais autônomos na área de confeitaria, que garanta a segurança de dados do usuário bem como uma interface familiar e intuitiva que o incentive a descobrir as funcionalidades do site.

2. Fundamentação Teórica

Para a construção do sistema foram utilizadas tecnologias com código fonte aberto livre, visando modernidade, usabilidade na construção do banco de dados, comunicação dinâmica entre os componentes do sistema e redução do custo para o profissional. De acordo com o SEBRAE (2013), o uso de software livre apresenta uma vantagem significativa sobre o software proprietário, especialmente para empresas ao evitar o pagamento de licenças, que representam um elevado custo anual para a utilização e atualização desses programas.

A fim de adequar essas características, TypeScript, em conjunto com seus frameworks Prisma e NestJS serão responsáveis pela comunicação e gerenciamento do banco de dados. A biblioteca de código aberto Prisma é um ORM (Mapeador de Relação de Objetos). É utilizada como alternativa na construção de consultas ou acesso a sistemas gerenciadores de banco de dados SQL. ORM é uma técnica que facilita a relação entre a representação dos objetos e as informações armazenadas, mapeando objetos para dados em um banco de dados relacional. A escolha da utilização de um ORM visa a simplicidade no desenvolvimento do aplicativo, concentrando-se na idealização da lógica de negócios do sistema [Rodrigues 2023].

O Prisma ainda oferece suporte à diversas linguagens de consulta estruturada (SQL), e dessas, foi escolhido o sistema de gerenciamento de banco de dados PostgreSQL. Segundo Luis (2020), o PostgreSQL usa e estende a linguagem SQL combinada com muitos recursos que armazenam e escalam com segurança as cargas de trabalho de dados mais complicadas e, por ser gratuito e de código aberto, o PostgreSQL é altamente extensível e adaptável para cada necessidade e para Adams:

Os bancos de dados desempenham um papel fundamental na persistência e recuperação de dados em sistemas de software. Eles fornecem um mecanismo estruturado para armazenar, organizar e recuperar informações de maneira eficiente. Ao escolher um banco de dados para um aplicativo, é crucial considerar fatores como escalabilidade, desempenho, segurança e facilidade de manutenção, para garantir uma base sólida para o armazenamento e manipulação de dados

Dado que a usabilidade também pode ser considerada como a qualidade de uso de um produto e/ou sistema a partir de um determinado usuário [Maia et al. 2020], para atingir maior eficiência, melhor produtividade, redução de erros, redução de treinamentos, melhor aceitação, benefícios apontados por Bevan (1998), foram utilizadas as seguintes ferramentas para a construção de uma aplicação web: a ferramenta de construção de projetos Vite, o *framework*, React e a biblioteca de estilos Tailwind CSS. Um *framework*, de acordo com Baltieri (2024) é um conjunto de bibliotecas que fornece funcionalidades e estruturas para o desenvolvimento de aplicações, permitindo a reutilização de código e oferecendo soluções par problemas de um domínio específico.

Vite é uma ferramenta de construção de projetos *front-end*, ou seja, destinada ao desenvolvimento da parte visível e interativa dos sites e aplicativos que os usuários veem e interagem diretamente. Ele facilita a configuração de ambiente, possuindo um carregamento rápido, permitindo um desenvolvimento ágil. A escolha dessa ferramenta se deu por seus benefícios como desempenho aprimorado, recarregamento rápido, configuração simplificada, tamanho de pacote menor e ecossistema crescente [Seguins 2023].

O React é uma biblioteca de código aberto criada pelo time do Facebook com a finalidade de otimizar a atualização do feed de notícias da rede social. Dada à sua eficiência também no Instagram, o React teve sua biblioteca aberta à comunidade, dando início à sua crescente popularidade no mercado de tecnologia. React permite dividir uma página inteira em partes chamadas de componentes, trabalhando cada um destes de forma individual e independente. Para criar estes componentes de forma mais dinâmica e com estilos flexíveis, foi utilizado o Tailwind, que é um *framework* CSS visando a utilidade, fornecendo várias classes utilitárias, opinativas e de propósito único, sendo possível utilizar diretamente dentro da marcação de texto para projetar um elemento [Simionato 2022].

Em conjunto com as tecnologias mencionadas anteriormente, o Docker foi utilizado para ser responsável por facilitar o gerenciamento e a criação do banco de dados. Docker é uma plataforma de código aberto que permite automatizar a implantação, o dimensionamento e a gestão de aplicativos em contêineres, encapsulando um sistema com todas as suas dependências, garantindo que ele funcione de maneira consistente em qualquer ambiente [Merkel 2014]. A utilização do Docker no sistema web se deu na criação desses contêineres que empacotam todas as dependências do sistema, incluindo o banco de dados PostgreSQL, as APIs construídas com NestJS e Prisma, e a aplicação *front-end* baseada em React. Vitalino e Castro (2016) mencionam que os

de um site onde um usuário é capaz de gerenciar dados de clientes, produtos e pedidos de forma acessível e intuitiva, facilitando a compreensão dos dados a partir da disponibilização de informações de maneira simples e objetiva aplicando conceitos de usabilidade ao site. O sistema conta com um mecanismo de criação e armazenamento de contas, com a implementação de uma criptografia para proteger as senhas dos usuários durante a autenticação e registro.

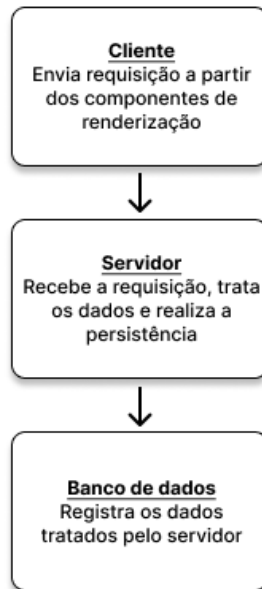


Figura 2. Fluxo de comunicação do sistema.

Ao se cadastrar no sistema, a senha informada pelo usuário é submetida a um processo de *hash* que utiliza da biblioteca “argon2”, um algoritmo de *hash* de senha que transforma a senha do usuário em uma sequência de caracteres que é armazenada no banco de dados no lugar da original. O sistema, antes de realizar o cadastro do usuário, valida as informações inseridas por ele, não permitindo cadastrar o mesmo *e-mail* mais de uma vez, e sugerindo o usuário a criar senhas seguras que contenham letras maiúsculas e minúsculas, caracteres especiais e números. Após o cadastro, é necessário fazer o *login* (Figura 3) no sistema. Durante este processo, a senha fornecida pelo usuário é comparada com o *hash* armazenado no banco de dados, e com a comparação sendo correta, a autenticação retorna como bem-sucedida.



Figura 3. Página de login.

Após a autenticação bem-sucedida, um token JWT (JSON Web Token) é gerado e retornado ao cliente. O JWT é um padrão (RFC-7519) que define como transmitir e armazenar objetos JSON de forma segura e compacta entre diferentes aplicações [Nascimento 2018].

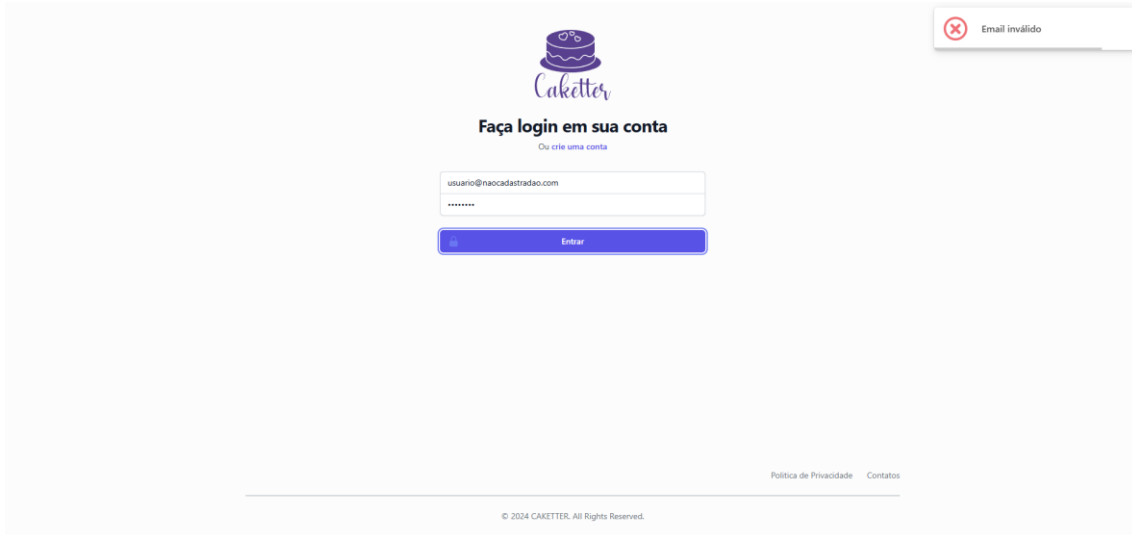


Figura 4 - Notificação de erro ao inserir e-mail inválido.

Este token contém informações importantes sobre o usuário, como ID e e-mail, e é assinado com uma chave secreta [Nascimento 2018]). Essas informações são necessárias para armazenar dados de usuário localmente, que serão posteriormente utilizados durante o fluxo do sistema. O token JWT é então utilizado pelo cliente para realizar requisições ao servidor, autenticando-se, então, como o usuário correto. O sistema conta com um sistema de notificações que retorna ao usuário quando há sucesso ou falha na autenticação e registro

Para garantir a segurança e privacidade dos dados, os registros são privados para cada usuário. O sistema usa uma estratégia de segregação, criando um identificador universal e exclusivo utilizado para identificação [Paixão 2018], *uuid*, conforme mostrado na Figura 5 a seguir, único para cada usuário quando o processo de autenticação é concluído. Mesmo que os usuários estejam no mesmo sistema, eles não podem acessar registros de outros usuários dessa maneira.

id	uuid	name	email	login	password	created_at	updated_at
1	534022b8-a04d-47d0-af67-97c0bc186548	master	master@master.com	master	\$argon2id\$v=19\$m=4096,t=3,p=157/QA3UdmyC0s6ArePzy4wASTKa/UPH5/aZavGeEH6jKZnLHAm+4kmuVKEWfTLQ	2024-01-23 23:01:20.794	2024-01-23 23:01:20.794
2	93834089-f697-42a1-b9a9-d04b7da191f2	Joao	joao@joao.com	Joao	\$argon2id\$v=19\$m=4096,t=3,p=15uokFXEfv444V7QHGCNwStkQr+sz2b46PjLNxKHHeu4rH6s1oe7UJ766CSAK	2024-01-24 00:32:46.778	2024-01-24 00:32:46.778
3	c6d40365-aa31-40f1-ab0a-6a556112f0d6	Pedro	pedro@pedro.com	Pedro	\$argon2id\$v=19\$m=4096,t=3,p=15/q8D4YDwSS6y4/QcYm6tqz\$Yp8WdeAgvWD4XGyCMkOekipQJCw/8Kf6zMrn8/lyk	2024-01-24 00:41:00.086	2024-01-24 00:41:00.086
4	d4401bf5-2e70-4sd1-81ae-72533b281c32	Testes	teste@teste.com	Testes	\$argon2id\$v=19\$m=4096,t=3,p=15DeeL/RR6jycKdvp5M78bg5q94f99lt3LTDLPUnroP8BshpTGacvdVpH5a3Ug	2024-02-01 15:51:43.231	2024-02-01 15:51:43.231
5	3994adf0-e4be-48db-8cb9-bece6f67809	Rabelo	rabelo@dev.com	Rabelo	\$argon2id\$v=19\$m=4096,t=3,p=15RzolzMQHA22zhQMwul+ZASveI3ec7wVWmE/saYntQmbGBWv7WwpCum+/7tZqOMz0	2024-04-19 21:33:47.421	2024-04-19 21:33:47.421

Figura 5. Como os usuários estão armazenados no banco.

O banco de dados do projeto foi construído a partir do comando de execução padrão do prisma, que gerará um novo diretório na pasta raiz do projeto com um arquivo `prisma.schema`, onde será escrito toda a estrutura do banco, com suas tabelas, tipos de

dados e relacionamentos. Depois de definir e salvar na sessão do usuário seus dados básicos, o usuário será redirecionado para uma página estática que apresenta uma breve descrição das funções disponíveis. Esta página contém um resumo de instruções sobre como usar o sistema e permite acessar as principais seções, como cadastro de clientes, bolos e pedidos. No topo da página estão os 4 botões, um para Sair do sistema (destacado em vermelho) e acesso aos módulos e telas de disponíveis no sistema: Clientes, Bolos e Pedidos.

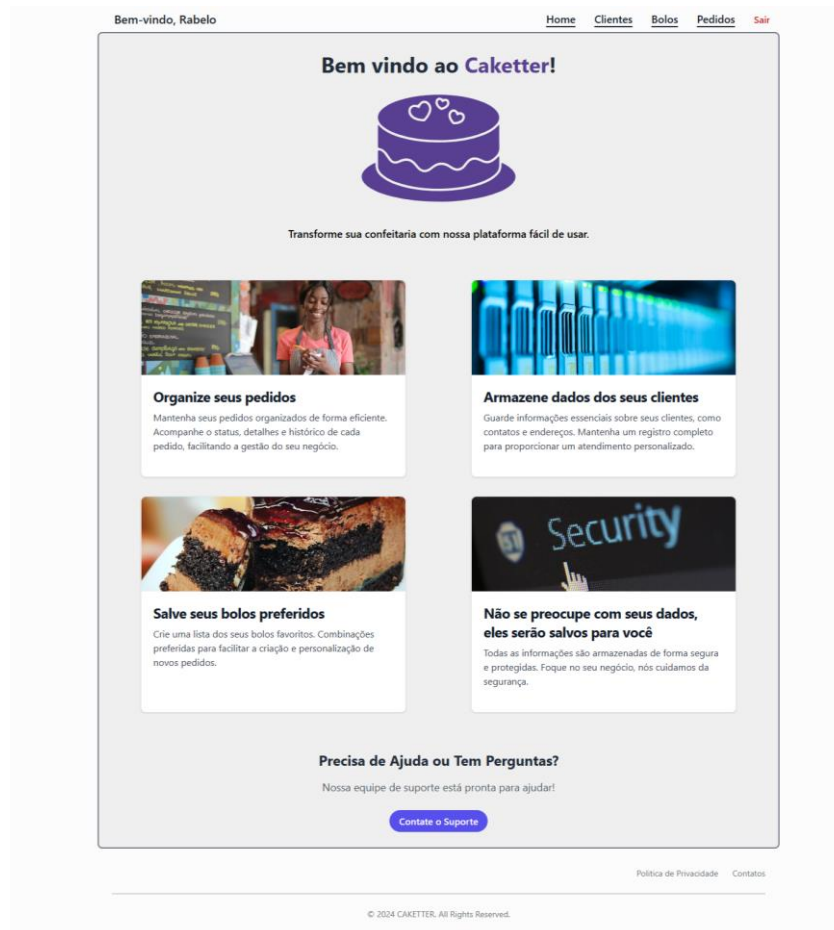


Figura 6. Página principal da aplicação.

Cada página foi projetada para atender às necessidades de organização dos registros dos clientes, visando otimizar a experiência do usuário. Oliveira (2015) destaca que o conceito de design vai além da estética da tela e abrange aspectos como a correção tempo de resposta, dificuldade de uso, desempenho em tarefas-chave, nível de erros, facilidade de aprendizado, fadiga do usuário, e a adaptação para necessidades especiais. Com isso em mente, foram criadas duas colunas de registro no banco de dados, que salvam a data de criação e a data de alteração de um registro, mantendo uma fidelidade e auditoria na base de dados.

A página de clientes possui uma tabela com todos os clientes que o usuário registrou, há também um botão simples que permite a adição de novos clientes. Estes botões são exibidos acima da tabela, e dentro de cada elemento da tabela, contendo ícones para editar e excluir. Considerando que um gesto semelhante ao que faríamos no mundo real e permitindo uma experiência sensorial e rica ao usuário (Oliveira, 2015), optou-se

por utilizar de botões que com ícones fáceis de entender e uma descrição apenas ao passar o mouse por cima, incentivando a curiosidade do usuário a descobrir qual ação cada botão deverá executar, aumentando o grau de satisfação e diminuindo o esforço para aprendizado do usuário, como pode ser observado na Figura 7. Para evitar confusão do usuário, uma mensagem é exibida na tabela informando ao usuário quando não há clientes cadastrados.

Bem-vindo, Rabelo [Home](#) [Clientes](#) [Bolos](#) [Pedidos](#) [Sair](#)

Clientes

Pesquisar

[+](#)

#	NOME	CONTATO	E-MAIL	ENDEREÇO	DATA DE CRIAÇÃO	ÚLTIMA ALTERAÇÃO		
1	Cliente Teste 2	(69) 99371-3611	cliente.teste.2@gmail.com	Av. Margaridas, 1234, Bairro Planalto	23/04/2024, 16:24	23/04/2024, 16:46	✎	🗑️
2	Cliente Teste 3	(69) 98271-9988	cliente.teste.3@gmail.com	Av. Margaridas, 1234, Bairro Planalto	23/04/2024, 16:24	23/04/2024, 16:47	✎	🗑️
3	Cliente Teste 7	(69) 97425-2705	cliente.teste.7@gmail.com	Av. Marechal Deodoro, 1234, Bairro Serraria	23/04/2024, 16:25	23/04/2024, 16:47	✎	🗑️
4	Cliente Teste 8	(69) 98971-8603	cliente.teste.8@gmail.com	Av. Marechal Deodoro, 1234, Bairro Serraria	23/04/2024, 16:26	23/04/2024, 16:48	✎	🗑️
5	Cliente Teste 9	(69) 97118-9025	cliente.teste.9@gmail.com	Av. Presidente Dutra, 1234, Bairro Centro	23/04/2024, 16:26	23/04/2024, 16:47	✎	🗑️

[Anterior](#) [Próxima](#)

5 por página ▾

[Política de Privacidade](#) [Contatos](#)

© 2024 CAKETER. All Rights Reserved.

Figura 7. Tabela de clientes.

Para facilitar a compreensão temporal dos dados e acompanhar a evolução de dados para o usuário, a tabela (Figura 7) apresenta os dados em ordem ascendente, isto é, o registro mais antigo é exibido primeiro na tabela, seguindo a data de sua criação.

O usuário pode clicar no botão com um símbolo de adição que está posicionado acima da tabela. Este botão o direciona ao formulário de registro de clientes. A figura 8 mostra como o formulário utiliza técnicas de usabilidade para melhorar a experiência do usuário, agrupando os campos de forma lógica, facilitando a compreensão e o preenchimento das informações necessárias. Cada campo do formulário possui um *placeholder*, que é um texto breve exibido dentro do campo para indicar ao usuário o tipo de informação esperada. No final do formulário, há um botão de ação “Cadastrar Cliente” claro e descritivo, indicando a ação que deverá ser realizada ao clicá-lo.

The screenshot shows a web page with a navigation bar at the top containing 'Bem-vindo, Rabelo' on the left and 'Home', 'Clientes', 'Bolos', 'Pedidos', and 'Sair' on the right. A purple cake icon is centered above the heading 'Cadastre um novo cliente'. Below the heading is a registration form with the following fields: 'Nome do cliente', '() ____-__', 'E-mail', 'Avenida' (with a 'Número' field to its right), and 'Bairro'. A blue button labeled 'Cadastrar Cliente' is positioned below the form. At the bottom right, there are links for 'Política de Privacidade' and 'Contatos'. The footer contains the text '© 2024 CAKETTER. All Rights Reserved.'

Figura 8. Formulário de registro de clientes.

Após o envio do formulário, o site fornece ao usuário um feedback visual através de notificações, informando sobre o sucesso ou falha de uma operação, considerando falhas de comunicação com o servidor, falta de preenchimento de campos necessários ou registros duplicados na base de dados.

This screenshot shows the same registration form as in Figure 8, but with a duplicate entry. The form fields are filled with: 'Cliente Duplicado', '(01) 33214-5668', 'email@duplicado.com', 'Av. Duplicado' (with '2222' in the 'Número' field), and 'Duplo'. A blue 'Cadastrar Cliente' button is at the bottom. A red error notification box in the top right corner contains a red 'X' icon and the text: 'Erro ao cadastrar cliente. Já existe um cliente com este contato cadastrado.'

Figura 9. Formulário de registro de clientes, erro ao cadastrar informação duplicada.

Quando o usuário clica em cadastrar cliente, o sistema valida as informações e envia a notificação de sucesso, redirecionando dinamicamente o usuário para a página da tabela de clientes (Figura 7).

Bem-vindo, Rabelo

Home Clientes Bolos Pedidos Sair

Cliente cadastrado com sucesso!

Clientes

Pesquisar

+

#	NOME	CONTATO	E-MAIL	ENDEREÇO	DATA DE CRIAÇÃO	ÚLTIMA ALTERAÇÃO		
1	Cliente Teste 2	(69) 99371-3611	cliente.teste.2@gmail.com	Av. Margaridas, 1234, Bairro Planalto	23/04/2024, 16:24	23/04/2024, 16:46		
2	Cliente Teste 3	(69) 98271-9988	cliente.teste.3@gmail.com	Av. Margaridas, 1234, Bairro Planalto	23/04/2024, 16:24	23/04/2024, 16:47		
3	Cliente Teste 7	(69) 97425-2705	cliente.teste.7@gmail.com	Av. Marechal Deodoro, 1234, Bairro Serraria	23/04/2024, 16:25	23/04/2024, 16:47		
4	Cliente Teste 8	(69) 98971-8603	cliente.teste.8@gmail.com	Av. Marechal Deodoro, 1234, Bairro Serraria	23/04/2024, 16:26	23/04/2024, 16:48		
5	Cliente Teste 9	(69) 97118-9025	cliente.teste.9@gmail.com	Av. Presidente Dutra, 1234, Bairro Centro	23/04/2024, 16:26	23/04/2024, 16:47		

Anterior Próxima

5 por página

Figura 10 - Notificação de sucesso ao cadastrar cliente.

Visando simplificar o processo de gerenciamento, o usuário pode pesquisar por um cliente utilizando um campo de texto localizado acima do botão de registro de novos clientes. Este campo recebe o dado inserido pelo usuário e envia uma requisição para o servidor com o termo digitado, realizando uma consulta e retornando uma lista de clientes cujos nomes contenham o termo fornecido pelo usuário. A tabela informa ao usuário quando não há registros similares com o termo fornecido.

A tabela de clientes inclui um sistema de paginação que facilita a navegação entre os registros, mantendo a experiência de usuário simples. Na parte inferior da tabela, há botões de controle que permitem ao usuário avançar ou retroceder entre as páginas de resultados. Além disso, o usuário pode escolher quantos registros serão mostrados por página, variando de 5 a 15 registros de clientes por página. Não é necessário que a página reinicie nem direcione o usuário a outra tela, apenas atualiza os registros da tabela dinamicamente.

Cada célula de registro de cliente na tabela, contém dois botões que são responsáveis pelo controle de dados dela. Um para editar um registro e outro para excluir o registro. Ao clicar no botão de editar, uma janela pop-up é exibida para o usuário que poderá alterar todos os registros do cliente que são apresentados em um formulário semelhante ao de registro. O usuário pode confirmar ou cancelar a alteração de registro.

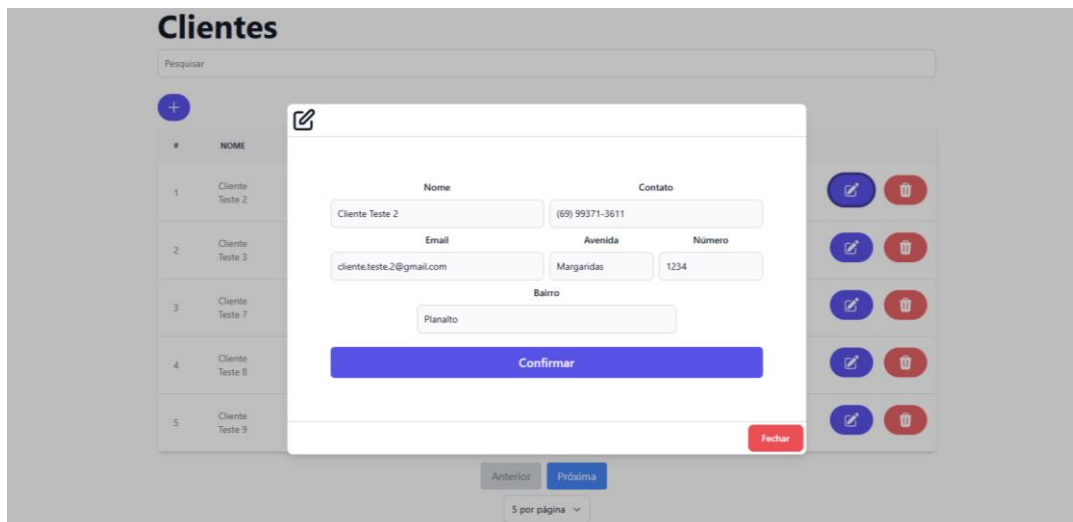


Figura 11. Janela pop-up para editar um cliente.

Ao clicar no botão de exclusão, uma janela pop-up é exibida para o usuário confirmar a exclusão do registro selecionado. Nessa janela há uma mensagem que alerta que essa ação é irreversível (Figura 12), possuindo também dois botões: um para cancelar e outro para confirmar. Caso o usuário negue a remoção do registro, a janela desaparece e retorna o usuário para a tabela de clientes. Após a confirmação da exclusão, o sistema faz a remoção do registro e atualiza a tabela de clientes de forma dinâmica. Além disso, o usuário recebe uma notificação visual informando que o cliente foi excluído com sucesso.

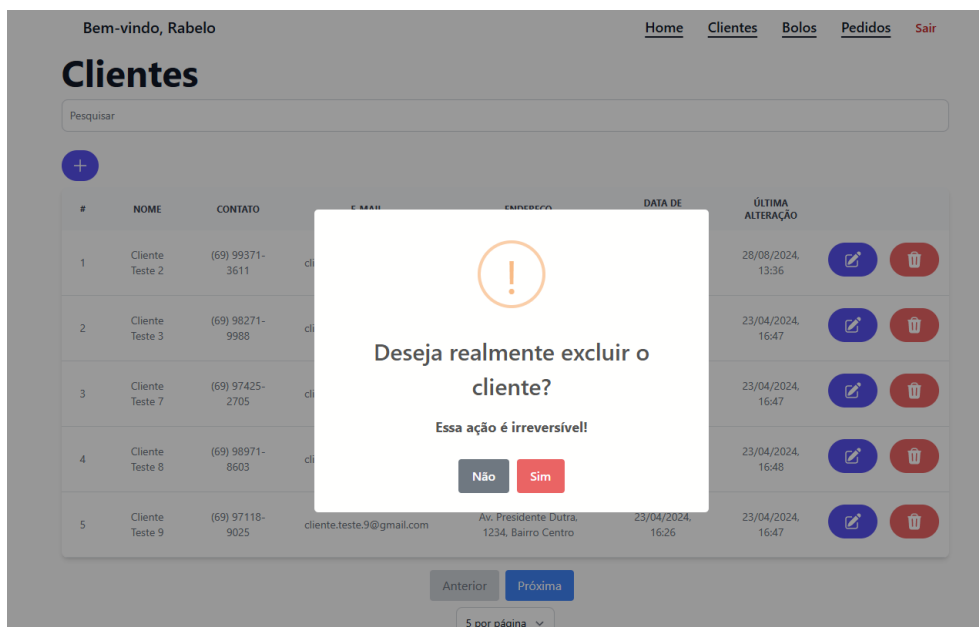


Figura 12 - Janela pop-up para confirmação de exclusão.

Apesar de pequenas diferenças pontuais como diferentes campos de registro e tamanho da tabela, as páginas de Bolos e Pedidos seguem no mesmo padrão da página de Clientes, mantendo uma familiaridade e autenticidade no leiaute do sistema, tornando-o fácil de aprender.

A última execução do usuário no fluxo padrão do site, é o registro de pedidos, que é feito com base numa lista de Clientes e Bolos, fazendo uma tabela única que referencia dados de ambas as tabelas.

Bem-vindo, Rabelo

Home Clientes Bolos Pedidos Sair

Pedidos

Pesquisar

#	BOLO					CLIENTE		PREÇO	DATA DE ENTREGA		
	Massa	Recheio	Tamanho	Nome no topo	Idade	Nome	Contato				
1	Chocolate	Morango	15	João	22	Cliente Teste 7	(69) 97425-2705	R\$ 200	30/04/2024, 18:40		
2	Chocolate	Morango	15	João	22	Cliente Teste 2	(69) 99371-3611	R\$ 200	27/04/2024, 19:13		
3	Chocolate	Morango	15	João	22	Cliente Teste 8	(69) 98971-8603	R\$ 2	19/06/2024, 12:18		

Anterior Próxima

5 por página

Política de Privacidade Contatos

© 2024 CAKETTER. All Rights Reserved.

Figura 13 - Tabela de pedidos.

Por ser construída apenas com o valor, data de entrega e chaves estrangeiras das tabelas de Clientes e Bolos, a tabela de Pedidos possui uma leitura simples e fácil, com dados que podem ser unidos de formas variadas. Como um lembrete ao usuário, quando um pedido estiver próximo da data de entrega, um ícone calendário é exibido acima da data, na cor vermelha. E, ao passar o mouse por cima do ícone, uma mensagem é exibida informando o tempo necessário até a entrega do pedido.

Pedidos

Pesquisar

#	BOLO					CLIENTE		PREÇO	DATA DE ENTREGA		
	Massa	Recheio	Tamanho	Nome no topo	Idade	Nome	Contato				
1	Chocolate	Morango	15	João	22	Cliente Teste 7	(69) 97425-2705	R\$ 200	30/04/2024, 18:40		
2	Chocolate	Morango	15	João	22	Cliente Teste 2	(69) 99371-3611	R\$ 200	27/04/2024, 19:13		
3	Chocolate	Morango	15	João	22	Cliente Teste 8	(69) 98971-8603	R\$ 2	19/06/2024, 12:18		
4	Chocolate	Morango	15	João	22	João Meireles	(12) 45125-1255	R\$ 22	31/08/2024, 13:41		

Faltam 3 dias para entregar este pedido

Figura 14 - Mensagem de aviso informando proximidade da data de entrega.

4. Conclusão

Destaca-se que o sistema Caketter, desenvolvido com uma interface simples e intuitiva, demonstra potencial para atender às necessidades de empresários individuais e pequenas

empresas. Utilizando tecnologias de código aberto, o Caketter proporciona uma solução acessível e eficiente, especialmente para aqueles que ainda não dispõem de recursos para sistemas complexos e caros.

Com a implementação do Docker, o sistema é facilmente escalável em ambientes de produção. Vale também ressaltar que a biblioteca React é facilmente portada para dispositivos móveis, aumentando ainda mais o alcance do sistema, seja ele lançado em lojas de aplicativos ou mantendo-se web, comportando-se como um aplicativo web progressivo (PWA) que se adapta a todas as plataformas que é utilizado.

A futura distribuição do sistema Caketter de neste formato poderá contribuir com muitos empresários individuais e autônomos, bem como pequenas empresas que no início do seu negócio ainda não dispõem de muitos recursos para aquisição de sistemas mais complexo e por muitas vezes com elevados custos financeiro e de infraestrutura. Ressaltamos ainda os possíveis ganhos em agilidade e organização dos pedidos, diminuindo falhas ocorridas do uso do sistema.

Embora o Caketter já ofereça uma plataforma funcional e eficaz para a gestão de pedidos na área de confeitaria, há oportunidades para melhorias futuras, como a expansão do banco de dados para incluir métricas financeiras para a produção de *dashboards* (painéis) interativos e integração de funcionalidades como upload de imagens e notificações por e-mail são aprimoramentos que podem enriquecer a experiência do usuário. Essas atualizações visam não apenas aumentar a interatividade, mas também fornecer um controle detalhado e personalizado para os usuários.

Os resultados obtidos demonstraram que o sistema “Caketter” é uma ferramenta simplificada, eficaz e intuitiva para auxiliar profissionais da área de confeitaria na gestão de seus pedidos, contribuindo para a otimização de processos e melhoria da qualidade dos serviços oferecidos, solidificando sua posição como uma solução eficaz para a gestão de pedidos.

5. Referências

Henrique, Pablo. Introdução a Typescript: o que é e como começar. Geekhunter. 2021. Disponível em: <https://blog.geekhunter.com.br/introducao-a-typescript/>. Acesso em: 01 de abr. 2024.

IPEA. A ciência e a tecnologia como estratégia de desenvolvimento. jul, 2019. Disponível em: <https://www.ipea.gov.br/cts/pt/central-de-conteudo/artigos/artigos/116-a-ciencia-e-a-tecnologia-como-estrategia-de-desenvolvimento>. Acesso em 25 de abr. 2024.

Lima, Marcio Luis Rodrigues de. PostgreSQL — um pouco deste SGBD muito robusto e Open Source. Medium. Disponível em: <https://medium.com/permalink-univesp/popostgresql-um-pouco-deste-sgbd-muito-robusto-e-open-source-44e96fe2a90b>. Acesso em: 21 abr. 2024.

Lopes, Michele. React: o que é e como funciona. EBAC. 2023. Disponível em: <https://ebaonline.com.br/blog/react-o-que-e-como-funciona>. Acesso em: 24 abr. 2024.

Maia, Maria; Barbosa, Ricardo; Williams, Peter. Usabilidade e experiência do usuário de sistemas de informação: em busca de limites e relações. Ciência da Informação em Revista, p. 36, 2020.

Merkel, Dirk. Docker: lightweight Linux containers for consistent development and

- deployment. Linux Journal, vol. 2014, no 239, março de 2014, p. 2:2.
- Nascimento, Wellington. Entendendo tokens JWT (Json Web Token). Medium. 2018. Disponível em: <https://medium.com/tableless/entendendo-tokens-jwt-json-web-token-413c6d1397f6>. Acesso em: 24 abr. 2024.
- Oliveira, Francisco; Oliveira, Fernando. Interação Humano Computador. 2. ed. Fortaleza, CE: EdUECE, 2015.
- SEBRAE. O que é software livre e quais as vantagens em usá-lo na sua empresa. Disponível em: <https://sebrae.com.br/sites/PortalSebrae/artigos/o-que-e-software-livre-e-quais-as-vantagens-em-usa-lo-na-sua-empresa,2928d53342603410VgnVCM100000b272010aRCRD>. Acesso em 13 de mai. 2024.
- Silvestre, Gabriel. ORM - Interface com o Banco de Dados. DEV Community. 2022. Disponível em: <https://dev.to/gabrielhsilvestre/orm-interface-com-o-banco-de-dados-3am1>. Acesso em 24 abr. 2024.
- Rodrigues, Kauê. ORM (Object Relational Mapper): Saiba o Que é. Disponível em: <https://dev.to/gabrielhsilvestre/orm-interface-com-o-banco-de-dados-3am1>. Acesso em: 1 set. 2024.
- Baltieri, André. O que é um Framework, Disponível em: <https://blog.balta.io/o-que-e-um-framework/>. Acesso em: 1 set. 2024.
- Torta, Mania. Confeitaria cresce 200% em meio à Pandemia. G1. 2021. Disponível em: <https://g1.globo.com/mg/triangulo-mineiro/especial-publicitario/mania-torta/noticia/2021/05/10/confeitaria-cresce-200percent-em-meio-a-pandemia.ghtml>. Acesso em: 22 de mar. 2023.
- Delage. O que é Order Management System? 2024. Disponível em: <https://delage.com.br/blog/oms-order-managment-system-o-que-e-e-porque-e-imprescindivel-para-atender-ao-novo-padrao-de-consumo/>. Acesso em: 1 set. 2024
- Vitalino, Jeferson Fernando Noronha; Castro, Marcus André Nunes. Descomplicando o Docker. Brasport, 2018.
- Paixão, João. O que é UUID? Porque usá-lo? Disponível em: <https://medium.com/trainingcenter/o-que-é-uuid-porque-usá-lo-ad7a66644a2b>. Acesso em 29 ago. 2024.