Revista Perquirere, vol. 23, n. 2: 158-175, 2026
© Centro Universitário de Patos de Minas
https://revistas.unipam.edu.br/index.php/perquirere

Sistema de gerenciamento de clientes para uma empresa de contabilidade: desenvolvimento de uma solução web para otimização do controle organizacional

Client management system for an accounting firm: development of a web-based solution for optimizing organizational control

LUCAS CÂNDIDO CLEMENTE AMARAL
Discente de Sistemas de Informação (UNIPAM)
lucasamarale@unipam.edu.br

JULIANA LILIS DA SILVA
Professora orientadora (UNIPAM)
juliana@unipam.edu.br

Resumo: Este trabalho apresenta o desenvolvimento de um sistema *web* para gerenciamento de clientes em uma empresa de contabilidade. O trabalho justifica-se no contexto organizacional por propor um sistema que otimiza a gestão dos serviços prestados e clientes da empresa. Nesse sentido, o projeto teve como objetivo automatizar processos internos relacionados ao controle de serviços prestados, geração de cobranças e comunicação com clientes, anteriormente realizados de forma manual. A pesquisa é classificada como aplicada, qualitativa e descritiva, e utilizou tecnologias como .NET no *back-end* e React.js com TypeScript no *front-end*, adotando a arquitetura limpa para garantir modularidade e fácil manutenção. O sistema desenvolvido integra processos como cadastro de clientes e serviços prestados, geração de cobranças, controle de usuários e envio de recados, automatizando tarefas e aprimorando a eficiência e o controle da organização. Assim, o sistema pode ser considerado um diferencial no desempenho das atividades de uma empresa de contabilidade.
Palavras-chave: sistema *web*; contabilidade; automação de processos; gerenciamento de clientes; arquitetura limpa.

Abstract: This study presents the development of a web-based system for client management in an accounting firm. The study is justified in the organizational context by proposing a system that optimizes the management of the company's services and clients. In this regard, the project aimed to automate internal processes related to the control of services provided, billing generation, and communication with clients, which were previously performed manually. The research is classified as applied, qualitative, and descriptive, and used technologies such as .NET for the back end and React.js with TypeScript for the front end, adopting clean architecture to ensure modularity and ease of maintenance. The developed system integrates processes such as client and service registration, billing generation, user control, and message sending, automating tasks and improving the organization's efficiency and control. Thus, the system may be considered a differential factor in the performance of activities in an accounting firm.
Keywords: web-based system; accounting; process automation; client management; clean architecture.

1 INTRODUÇÃO

O profissional de contabilidade exerce um papel de extrema importância na gestão empresarial, atuando em todas as fases do ciclo de vida do negócio, desde sua abertura até o seu encerramento, independentemente do tamanho da empresa. Uma das responsabilidades do contador envolve o suporte no controle financeiro das organizações, registrando todas as suas transações. Essa atividade é fundamental para a geração de relatórios e para a condução de análises financeiras, o que proporciona aos gestores uma melhor compreensão da saúde financeira de seu negócio e permite que identifiquem áreas de melhoria com base nesses dados (Unicesumar, 2017).

Alguns dos principais desafios para as empresas de contabilidade estão relacionados à gestão eficiente de seus clientes. Atividades fundamentais como o registro minucioso e o controle dos serviços prestados para cada cliente, assim como a manutenção de um histórico detalhado dessas informações, muitas vezes são feitas de forma manual. Além disso, a comunicação de recados importantes geralmente é feita da mesma maneira. De acordo com Próvio (2023), outro ponto relevante é a dificuldade de alguns profissionais em lidar com a evolução da tecnologia, o que pode impactar negativamente a rotina operacional do negócio, dificultando a integração de ferramentas que poderiam otimizar os processos da empresa.

Diante desses desafios, a tecnologia desempenha um papel fundamental na modernização dos processos contábeis, promovendo transformações significativas na forma como as informações são registradas, processadas e analisadas. Com o aumento da complexidade das exigências fiscais e a demanda por informações em tempo real, cresce a necessidade por *softwares* especializados que automatizam e integram as atividades contábeis. Essas ferramentas não apenas reduzem o risco de erros humanos, mas também otimizam o tempo dos profissionais. A automação contábil garante maior eficiência, precisão e conformidade, consolidando-se como um recurso essencial para a competitividade e sustentabilidade das organizações.

Diante disso, esta pesquisa visou responder à seguinte pergunta: "De que maneira a presença de uma solução automatizada para a gestão de serviços prestados, comunicação com clientes e registro financeiro afeta a eficiência operacional e a administração de uma empresa de contabilidade?"

Nesse sentido, este trabalho se justifica no contexto organizacional ao sugerir o desenvolvimento de um sistema que proporcione benefícios diretos para a empresa de contabilidade, ao possibilitar uma gestão mais eficaz dos serviços oferecidos, facilitar a comunicação com os clientes e aprimorar o controle financeiro.

Portanto, o presente trabalho teve como objetivo geral desenvolver uma aplicação com uma interface amigável e de fácil navegação que possibilite o envio de *e-mails*, como também a manutenção de um histórico dos serviços prestados e valores cobrados para cada um dos clientes de uma empresa de contabilidade. Para alcançar esse objetivo, foram enumerados os seguintes objetivos específicos: gerenciar serviços prestados, gerenciar usuários, gerenciar clientes, enviar *e-mails* para vários clientes, gerenciar histórico de serviços prestados para cada cliente, juntamente com o valor cobrado por esses serviços, e gerenciar histórico de cobranças geradas para cada cliente a cada mês.

2 REFERENCIAL TEÓRICO

Nessa seção, são apresentados conceitos relevantes para o entendimento da arquitetura do sistema e da contextualização do trabalho do profissional da área de contabilidade.

2.1 CONCEITOS E PRINCÍPIOS DA CONTABILIDADE

O contador é o especialista em Ciência Contábil, que oferece os fundamentos necessários para a gestão e integração de uma empresa. Além de elaborar relatórios e demonstrações contábeis, ele também analisa essas informações e sugere possíveis soluções para os problemas empresariais presentes e futuros que identifica no desempenho de suas funções (Capistrano, 2001).

O campo de atuação do profissional contábil nas empresas é bastante extenso. Através da informação contábil, a empresa pode realizar um planejamento tributário eficaz, analisando balanços e demonstrações contábeis. Isso permite avaliar a evolução dos ativos e passivos, das receitas, despesas e custos, além de medir a lucratividade e a rentabilidade (Fonseca *et al*., 2014).

Relatórios contábeis, apenas com dados numéricos e quantitativos, não são suficientes para um gerenciamento eficaz do patrimônio empresarial. É essencial contar com a orientação de um profissional qualificado para interpretar essas informações e ajudar na tomada de decisões pela administração da empresa (Capistrano, 2001).

No passado, o profissional contábil gastava muito tempo com métodos manuais de escrituração, produzindo informações defasadas e relatórios básicos para o fisco, sem contribuir significativamente para as informações gerenciais necessárias à tomada de decisões empresariais. Com o avanço da tecnologia da informação, muitos desses esforços puderam ser substituídos por computadores e bancos de dados, oferecendo diversas facilidades no processamento das informações (Fonseca *et al*., 2014).

2.2 SISTEMAS DE INFORMAÇÃO APLICADOS À CONTABILIDADE

Segundo Possebon e Ribeiro (2020, p. 14), os sistemas de informação gerencial podem ser definidos "como os sistemas que apoiam e fornecem aos gestores as informações necessárias para que possam tomar decisões de forma efetiva, com o objetivo de gerenciar suas organizações com eficácia". Dessa forma é possível concluir que os sistemas de informação gerencial desempenham um papel fundamental no contexto organizacional, pois são responsáveis por transformar dados em informações úteis, permitindo que os gestores tenham uma visão mais clara do desempenho da organização, facilitando o planejamento.

Muitas empresas adotam os sistemas ERP, que são pacotes de *software* padronizados, adquiridos prontos. Eles exigem das empresas a adaptação de seus processos de negócio às funcionalidades disponíveis nesses sistemas. Além disso, no contexto dos sistemas de informação, os ERPs têm assumido um papel central na gestão de médias e grandes organizações, consolidando-se como ferramentas estratégicas para a integração e o controle das operações empresariais (Gomes *et al*, 2022).

A contabilidade é um dos setores contemplados pelos sistemas ERP. As demonstrações contábeis como o balanço patrimonial, o demonstrativo de resultados e a demonstração do fluxo de caixa, exigidas por obrigações legais ou utilizadas para fins de análise gerencial, desempenham um papel fundamental no suporte à tomada de decisões e na condução estratégica das organizações (Gomes *et al*, 2022).

Outras empresas adotam sistemas de informação que atendam a demandas específicas de acordo com suas necessidades. Alguns desses sistemas são desenvolvidos especialmente para uma dada empresa, de acordo com os requisitos que são identificados pela equipe de desenvolvimento do *software* (Silveira, 2023).

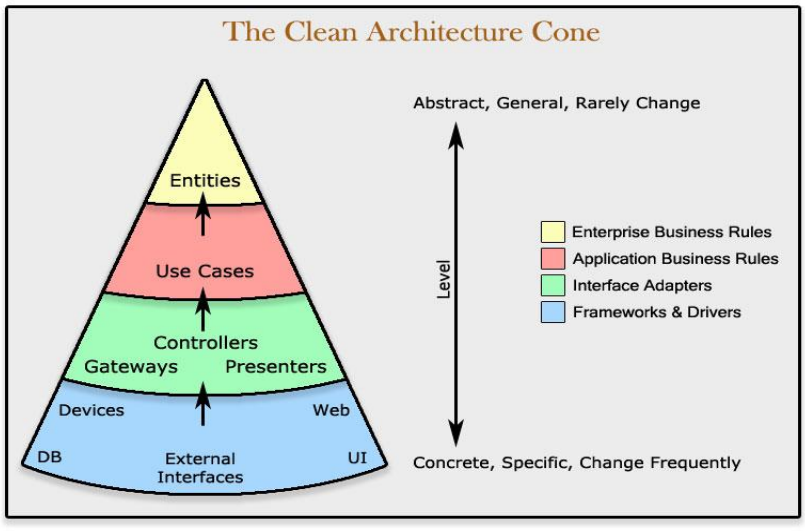
2.3 ARQUITETURA EM CAMADAS E ARQUITETURA LIMPA

A arquitetura em camadas é um dos padrões mais comuns desde a construção dos primeiros sistemas de *software* de grande porte nas décadas de 60 e 70. Nesse modelo, as classes são agrupadas em módulos maiores, conhecidos como camadas, que são organizadas hierarquicamente. Dessa forma, uma camada só pode utilizar serviços como chamar métodos, criar objetos, estender classes, definir parâmetros, lançar exceções, entre outros da camada diretamente abaixo dela (Valente, 2020).

A arquitetura limpa, proposta por Robert C. Martin, é um modelo de *design* de *software* voltado à criação de sistemas modulares, escaláveis e de fácil manutenção. Seu principal objetivo é garantir que o código seja flexível e sustentável ao longo do tempo, por meio da clara separação de responsabilidades e da organização em camadas bem definidas, cada uma com funções específicas (Boussiengui, 2023).

Essa abordagem propõe uma estrutura de código clara e coerente, baseada na comunicação entre camadas dispostas do exterior para o interior como mostrado na Figura 1. As camadas mais internas concentram a lógica de negócio e permanecem isoladas das externas, o que assegura que o sistema não dependa diretamente de *frameworks* ou bibliotecas externas. Dessa forma, a aplicação mantém uma estrutura organizada e um fluxo lógico bem delineado (Boff *et al*, 2023).

Figura 1: Cone da arquitetura limpa



Fonte: Outlaw, 2018.

Com essa separação, minimiza-se a influência de tecnologias externas sobre o núcleo do sistema, permitindo que a lógica de negócio continue estável mesmo diante de mudanças. A arquitetura limpa busca construir soluções que vão além da funcionalidade imediata, priorizando simplicidade, clareza e longevidade, ao mesmo tempo em que alinham a implementação técnica aos objetivos estratégicos da organização (Boussiengui, 2023).

2.4 API REST

Web services são servidores *web* desenvolvidos para atender a necessidades específicas de sites ou outras aplicações. Programas clientes se comunicam com esses servidores usando *Application Programming Interfaces* (APIs). Em termos gerais, uma API disponibiliza um conjunto de dados e funcionalidades para facilitar a interação entre programas de computador e possibilitar a troca de informações (Masse, 2011).

O estilo arquitetônico REST (*Representational State Transfer*) é amplamente utilizado no desenvolvimento de APIs para *web services* modernos. Uma API *web* que adota o estilo REST é chamada de REST API. Quando um *web service* utiliza uma REST API, ele é considerado "RESTful". Uma REST API é formada por uma coleção de recursos interconectados, e esse conjunto é conhecido como o modelo de recursos da REST API (Masse, 2011).

A arquitetura REST é fundamentada em quatro princípios principais. Primeiramente, a identificação de recursos é feita por meio de URIs (*Uniform Resource Identifier*), que oferecem um endereço único para interagir com os recursos de um serviço *web* RESTful, facilitando a descoberta e acesso aos recursos. Em segundo lugar, a interface uniforme é implementada através de quatro operações básicas: *PUT*, *GET*, *POST* e *DELETE*. O método *POST* é usado para criar novos recursos, que podem ser removidos com o *DELETE*, enquanto o *GET* recupera a representação atual de um recurso e o *PUT* atualiza o estado de um recurso existente (Pautasso *et al*, 2008).

Além disso, a arquitetura RESTful adota mensagens autodescritivas, que permitem a dissociação entre recursos e suas representações, possibilitando acesso em diferentes formatos. Os metadados associados são essenciais para controle de cache, detecção de erros e autenticação. Por fim, as interações *stateful* são gerenciadas por meio de hiperlinks, com mensagens de solicitação autossuficientes. Técnicas como URI reescrita, *cookies* e campos de formulário ocultos são utilizadas para gerenciar o estado entre as interações, com o estado sendo incorporado nas mensagens de resposta para indicar os futuros estados válidos (Pautasso *et al*, 2008).

3 METODOLOGIA

Este trabalho caracteriza-se como uma pesquisa do tipo aplicada, com abordagem qualitativa e descritiva. Aplicada, pois visa gerar uma solução para auxiliar os trabalhos desenvolvidos por escritórios de contabilidade, por meio do desenvolvimento de um software de gestão. Adota-se uma abordagem qualitativa, tendo em vista a ênfase na compreensão das necessidades dos usuários, seus processos e desafios operacionais. A pesquisa é descritiva, uma vez que se propõe a observar,

registrar e analisar os fenômenos relacionados à gestão contábil, sem interferir diretamente em seu funcionamento, mas busca retratá-los para fundamentar o desenvolvimento da solução tecnológica proposta.

A primeira etapa do trabalho foi dedicada à pesquisa detalhada em diversas fontes, incluindo livros, artigos acadêmicos e material disponível na *web*, com o objetivo de aprofundar conhecimentos nos assuntos necessários ao desenvolvimento da aplicação proposta. Essa fase inicial permitiu o levantamento de dados e informações relevantes, que serviram de base para o desenvolvimento das próximas etapas.

Posteriormente, foi realizado um levantamento de requisitos por meio de entrevistas e reuniões com os membros da empresa de contabilidade, com o objetivo de definir o escopo do sistema, garantindo um planejamento alinhado aos objetivos da empresa. A partir da definição dos requisitos foi gerado o protótipo da aplicação e o modelo do banco de dados utilizando as ferramentas Figma e SQL Power Architect.

O *back-end* foi desenvolvido com base nos princípios da arquitetura limpa, promovendo a separação de responsabilidades entre os diferentes níveis da aplicação, mantendo a lógica de negócio isolada de frameworks e bibliotecas externas. Como sistema gerenciador de banco de dados, foi utilizado o SQL Server, que é um banco de dados relacional.

Para o desenvolvimento foram utilizadas as seguintes tecnologias:

- para o *back-end*, foi empregado o *framework* .NET, com a linguagem de programação C#.
- no *front-end*, foi utilizado o *framework* React.js, em conjunto com a linguagem TypeScript e a biblioteca Next.js.

Para o gerenciamento do trabalho, foi adotado o método ágil Kanban, em conjunto com a ferramenta Trello, visando a uma organização eficiente das tarefas e entregas.

Por fim, após a finalização do desenvolvimento das funcionalidades do sistema, foi realizada a implementação de testes unitários para assegurar a qualidade do código. Além disso, todos os dados sensíveis que foram utilizados na aplicação foram devidamente ocultados, garantindo assim a proteção e a preservação dessas informações. Essa abordagem visa validar as funcionalidades, proteger a privacidade dos funcionários e clientes da empresa e garantir a integridade dos dados.

4 ANÁLISE E DISCUSSÃO DOS RESULTADOS/DESENVOLVIMENTO

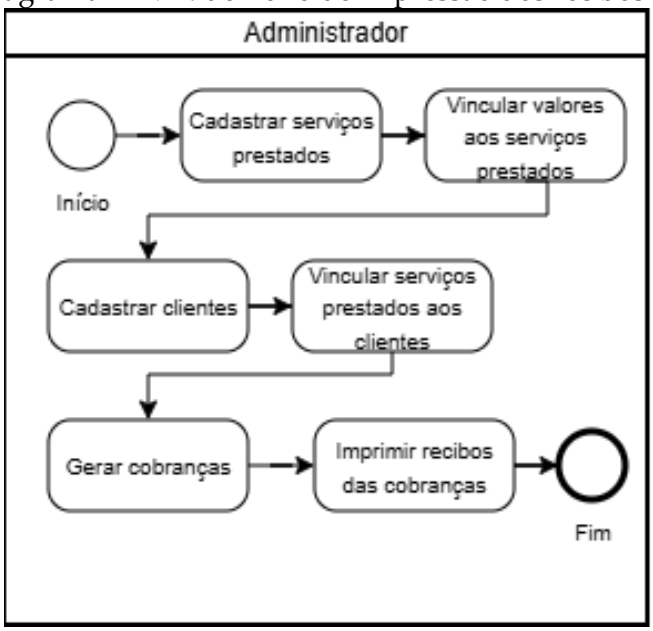
Nesta seção, são apresentados o planejamento, o desenvolvimento e as funcionalidades do sistema desenvolvido.

4.1 PLANEJAMENTO DO SISTEMA

Como resultado das etapas de levantamento, análise e especificação de requisitos, foram elaborados o modelo do banco de dados e os diagramas BPMN (*Business Process Model and Notation*) do sistema. O diagrama BPMN apresentado na

Figura 2 evidencia o fluxo do usuário administrador para geração das cobranças. Inicialmente esse usuário deve realizar o cadastro dos serviços prestados definindo valores a eles, em seguida cadastrar os clientes vinculando serviços prestados a eles. Por fim, o usuário poderá gerar as cobranças e imprimir os recibos, finalizando o processo.

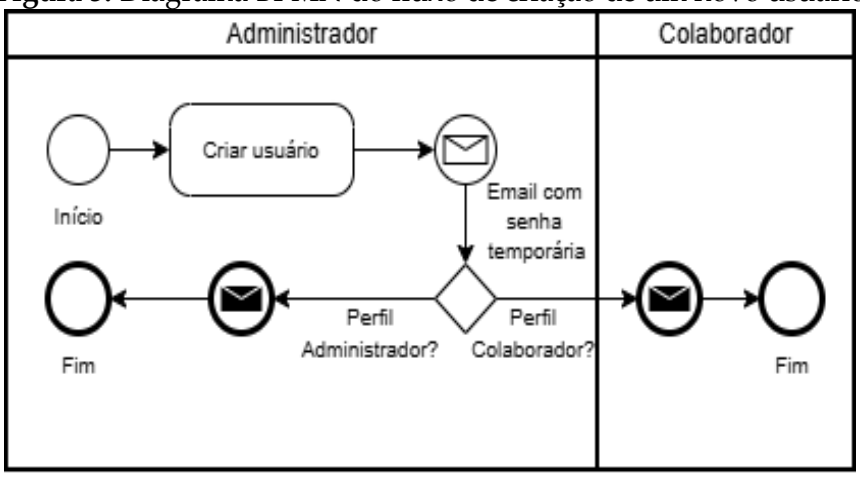
Figura 2: Diagrama BPMN do fluxo de impressão dos recibos de cobrança



Fonte: elaborado pelos autores, 2025.

Também se fez necessário compreender o fluxo representado na Figura 3, que trata do processo de criação de um novo usuário. Esse processo é iniciado por um administrador; após concluir o cadastro básico do novo usuário, é enviado automaticamente um *e-mail* contendo uma senha temporária para que o novo colaborador ou administrador possa acessar o sistema.

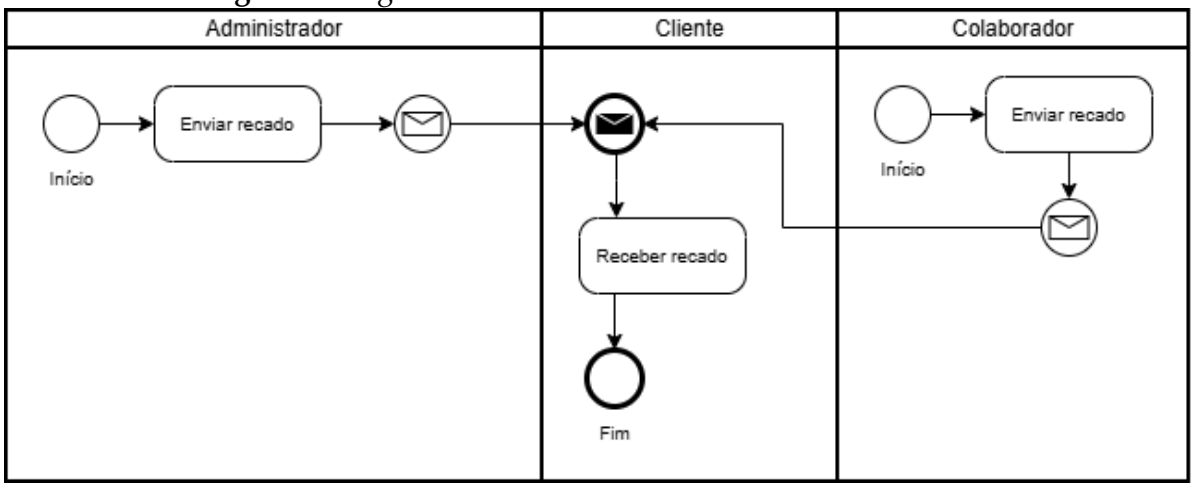
Figura 3: Diagrama BPMN do fluxo de criação de um novo usuário



Fonte: elaborado pelos autores, 2025.

Outro processo analisado foi o fluxo de envio de recados. Como apresentado na Figura 4, essa ação pode ser realizada por um administrador ou por um colaborador. Os usuários poderão iniciar o processo preenchendo um formulário com as informações da mensagem e um filtro de tipo de pessoa (física ou jurídica) que ao ser submetido enviará o recado para todos os clientes que se encaixarem no filtro informado.

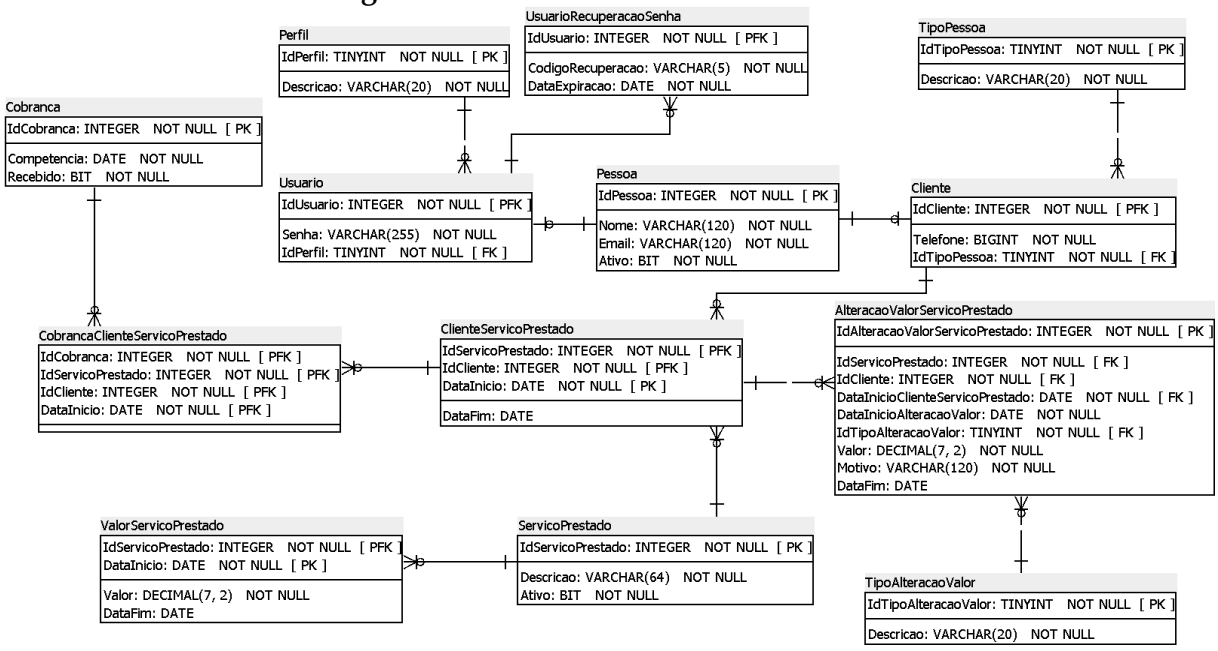
Figura 4: Diagrama BPMN do fluxo de envio de recados



Fonte: elaborado pelos autores, 2025.

Com base nas informações coletadas junto à empresa de contabilidade e de acordo com o entendimento dos fluxos dos processos analisados, foi desenvolvido o modelo de banco de dados apresentado na Figura 5, tendo como objetivo representar de maneira organizada as entidades, relacionamentos e regras de negócio identificadas durante o planejamento do sistema.

Figura 5: Modelo do banco de dados



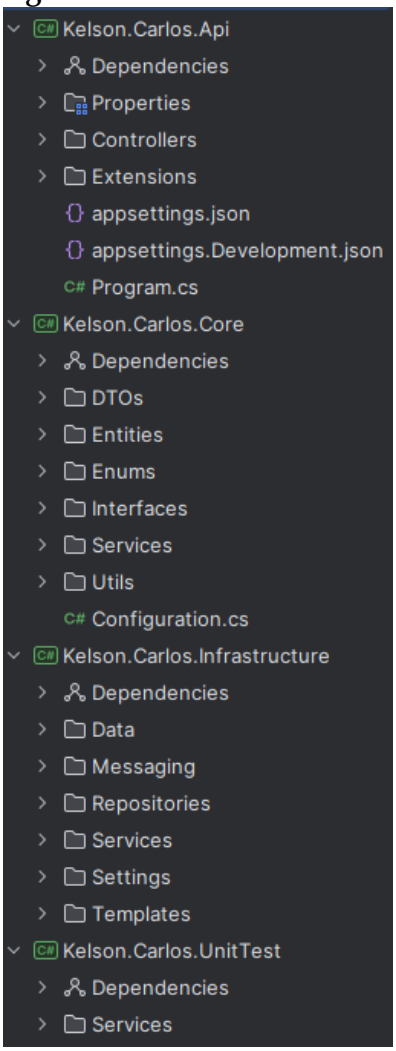
Fonte: elaborado pelos autores, 2025.

Como pode ser observado na Figura 5, as tabelas “ServicoPrestado” e “ValorServicoPrestado” armazenam, respectivamente, os serviços oferecidos pela empresa e o histórico de seus valores. Já as tabelas “TipoPessoa”, “Cliente” e “ClienteServicoPrestado” registram os dados dos clientes e seus vínculos com os serviços prestados. A tabela “Cobranca” guarda as cobranças geradas para cada cliente, enquanto “CobrancaClienteServicoPrestado” relaciona essas cobranças com os serviços correspondentes. As tabelas “TipoAlteracaoValor” e “AlteracaoValorServicoPrestado” mantêm o histórico de acréscimos e descontos aplicados aos serviços. Por fim, as tabelas de autenticação e autorização controlam o acesso dos usuários, sendo que tanto clientes quanto usuários são representados como “Pessoa” no sistema.

4.2 ESTRUTURA DO *BACK-END*

Para o desenvolvimento do *back-end*, o sistema foi estruturado em quatro camadas: Api, *Core*, *Infrastructure* e *UnitTest*, com base na arquitetura limpa, como pode ser visualizado na Figura 6.

Figura 6: Estrutura do *back-end*



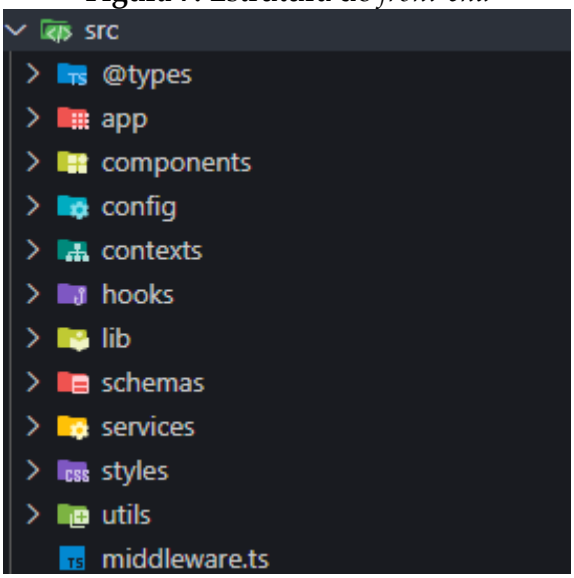
Fonte: elaborado pelos autores, 2025.

A camada Api é responsável pela exposição dos *endpoints* e pelas configurações gerais da aplicação. A camada *Core* contém a lógica de negócio do sistema, mantendo-se independente de tecnologias externas. Já a camada *Infrastructure* implementa o acesso ao banco de dados, integrações com serviços externos e utilização de *frameworks* e bibliotecas de terceiros, que, quando necessários na camada *Core*, são acessados por meio de interfaces, sendo suas implementações injetadas na camada Api através do mecanismo de injeção de dependência. Por fim, o projeto *UnitTest* abriga os testes unitários dos serviços, garantindo a validação das regras de negócio.

4.3 ESTRUTURA DO *front-end*

Para o desenvolvimento do *front-end*, a estrutura de pastas foi organizada de forma a separar claramente as responsabilidades, como demonstrado na Figura 7, visando facilitar a manutenção do código e reutilização de componentes.

Figura 7: Estrutura do *front-end*



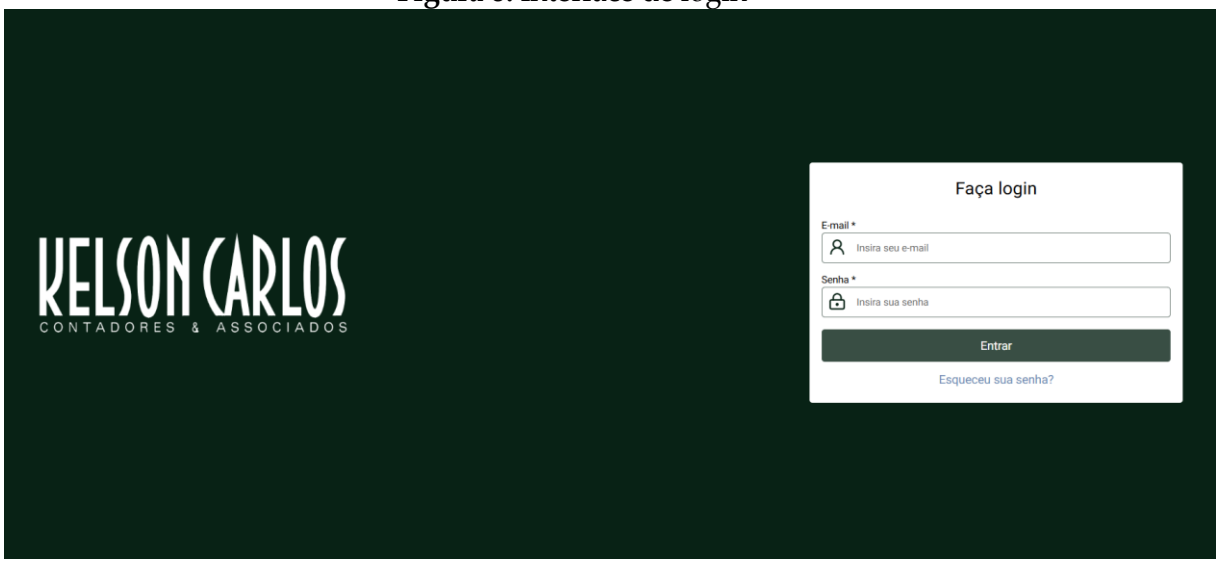
Fonte: elaborado pelos autores, 2025.

Na Figura 7 foram apresentadas as pastas criadas. Dentro de "*@types*" podem ser encontradas as tipagens e *enums* utilizados na integração com o *back-end*, enquanto "services" abriga as requisições e configurações da API. As telas estão localizadas em "*app*", e os componentes reutilizáveis em "*components*". As configurações gerais do projeto encontram-se em "*config*", e a pasta "*lib*" contém a configuração necessária para o funcionamento da biblioteca *Styled Components* (utilizada para estilização) com o Next.js. Já "*styles*" reúne estilos compartilhados entre diferentes telas. O contexto de autenticação do usuário está implementado em "*contexts*", enquanto "*hooks*" armazena os *hooks* customizados criados para o projeto. As validações de formulários ficam em "*schemas*", e as funções utilitárias em "*utils*". Por fim, o arquivo "*middleware*", localizado na raiz, segue as recomendações do Next.js para definir as rotas privadas da aplicação.

4.4 FUNCIONAMENTO E INTERFACE DA APLICAÇÃO

Para acessar o sistema, o usuário deve realizar o login através da interface apresentada na Figura 8, utilizando as credenciais de uma conta previamente criada por um administrador. Caso o usuário esqueça a senha, é possível solicitar a recuperação informando o e-mail cadastrado. Nessa situação, o sistema gera um código de recuperação com validade de 30 minutos, que é enviado para o e-mail informado dentro de um link que redireciona o usuário para uma rota específica de redefinição de senha.

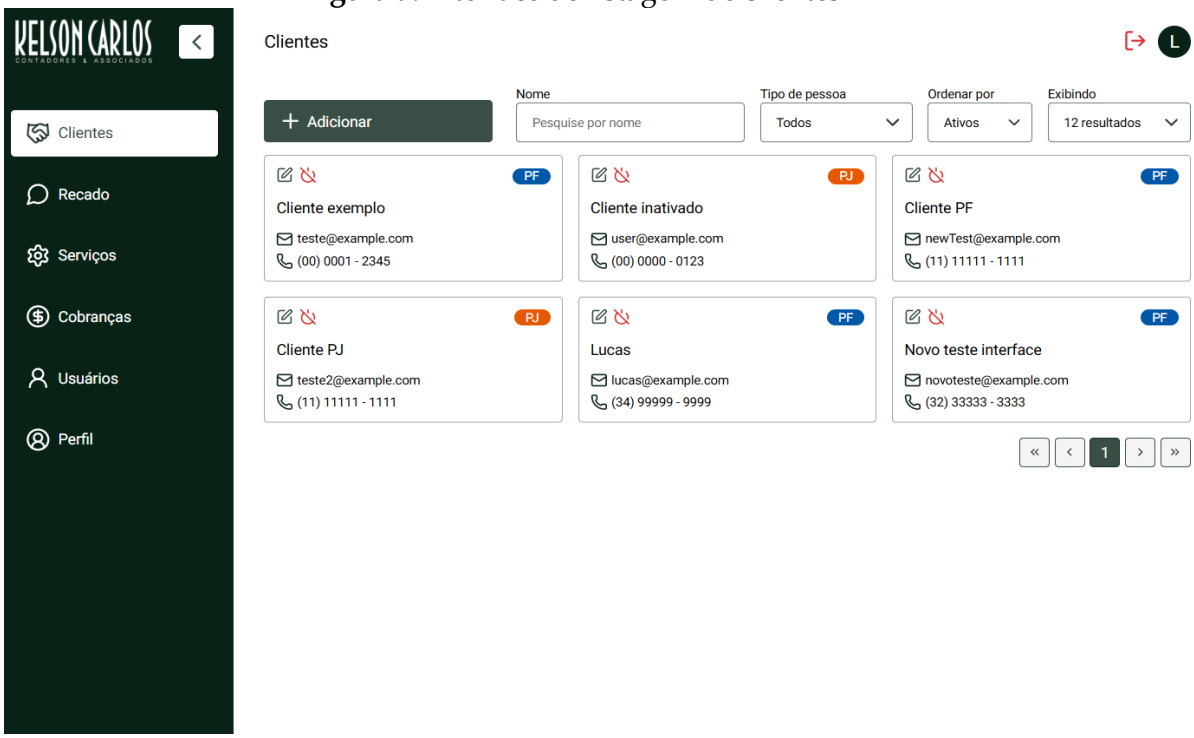
Figura 8: Interface de login



Fonte: elaborado pelos autores, 2025.

Após realizar o login, um usuário com permissão de administrador será redirecionado para a listagem de clientes, como mostrado na Figura 9. Após o redirecionamento, o usuário poderá filtrar os clientes cadastrados, cadastrar um novo ou se preferir editar um cadastro existente. Além disso, também é possível inativar os clientes já cadastrados.

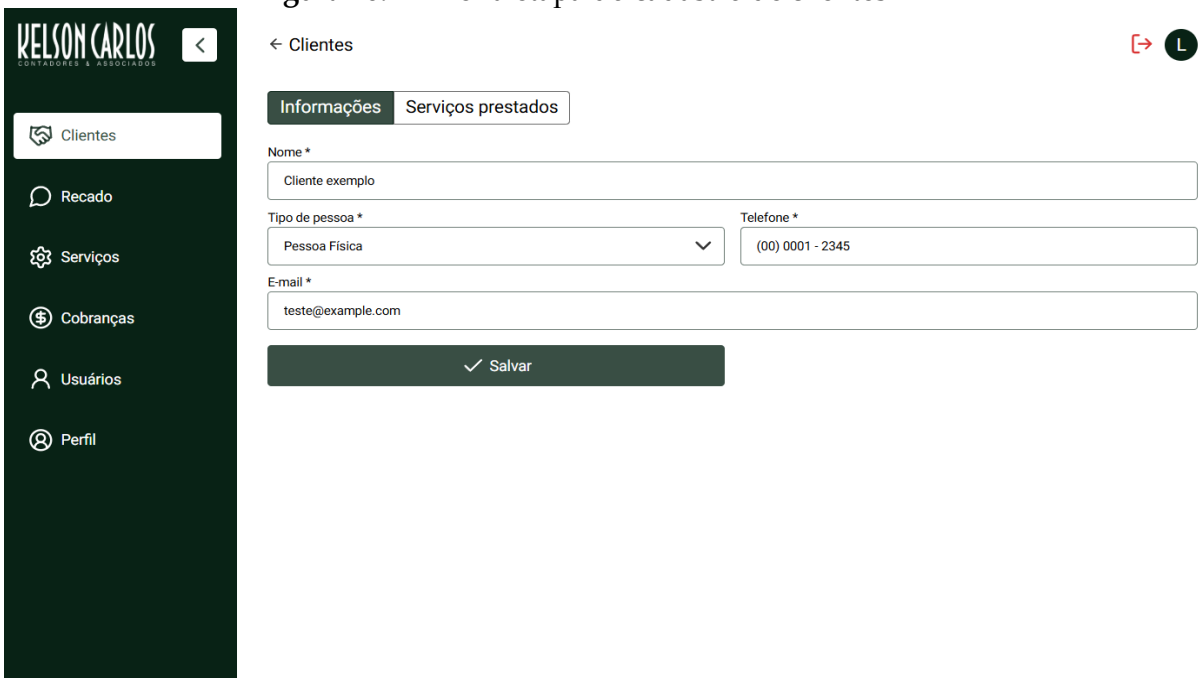
Figura 9: Interface de listagem de clientes



Fonte: elaborado pelos autores, 2025.

Ao acessar a funcionalidade de cadastro de clientes, o processo é dividido em duas etapas. A primeira, apresentada na Figura 10, é um formulário com as informações básicas do cliente, que deve ser submetido antes de prosseguir para a etapa de definição dos serviços prestados à esse cliente.

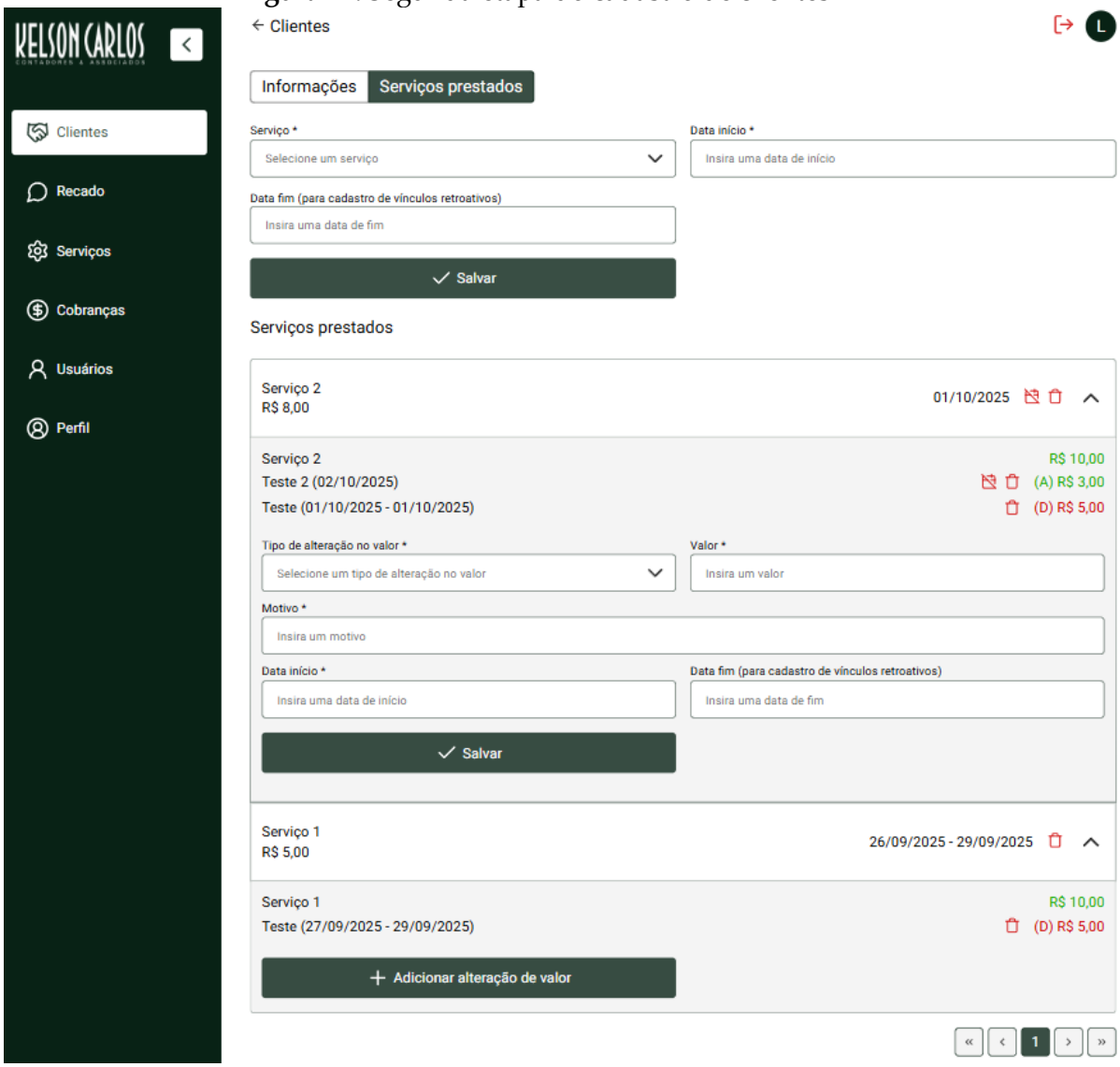
Figura 10: Primeira etapa do cadastro de clientes



Fonte: elaborado pelos autores, 2025.

Já a segunda etapa, representada na Figura 11, é onde o usuário poderá vincular os serviços prestados para aquele cliente, detalhar o valor cobrado para cada vínculo, acompanhar o histórico dos relacionamentos criados, excluir e encerrar esses vínculos, como também adicionar, encerrar, excluir e visualizar alterações de valor (acréscimo ou desconto) na prestação dos serviços.

Figura 11: Segunda etapa do cadastro de clientes



Fonte: elaborado pelos autores, 2025.

Semelhante à funcionalidade de listagem de clientes, existe uma interface com as funções de listagem de serviços prestados pelo escritório de contabilidade. Nessa listagem, é possível filtrar os resultados, criar um novo serviço ou editar um existente, e inativar um registro. Diferentemente da funcionalidade de listagem de clientes, é possível excluir um serviço prestado caso não esteja vinculado a nenhum cliente.

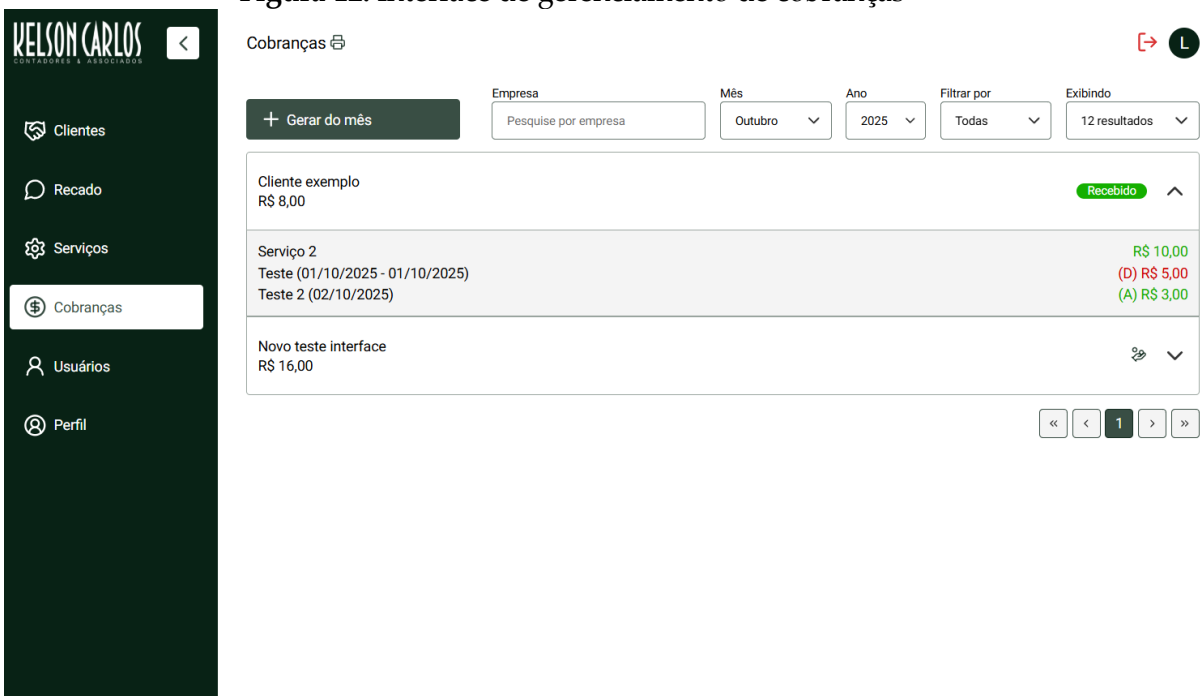
No cadastro de serviços prestados, o processo também é dividido em duas etapas. Na primeira etapa, é apresentado um formulário no qual o usuário deve informar

o nome do serviço prestado; já na segunda etapa é possível cadastrar o valor a ser cobrado por aquele serviço ao ser vinculado com um cliente, acompanhar o histórico de valores, encerrar o valor ativo e excluir algum registro, semelhante às etapas do cadastro de clientes.

Após concluir o cadastro dos serviços prestados, dos clientes e dos respectivos vínculos entre eles, o usuário pode acessar a interface de cobranças, onde é possível gerar as cobranças referentes a um mês e ano específicos. Ao realizar essa geração, o sistema exibe uma listagem detalhada das cobranças correspondentes ao período selecionado, incluindo os valores e informações associadas.

A interface, conforme mostrado na Figura 12, oferece filtros para a consulta das cobranças e apresenta um ícone de ação que permite confirmar o pagamento de cada cobrança individualmente. Além disso, há um outro ícone na parte superior da tela que possibilita imprimir os recibos das cobranças filtradas pelo período escolhido.

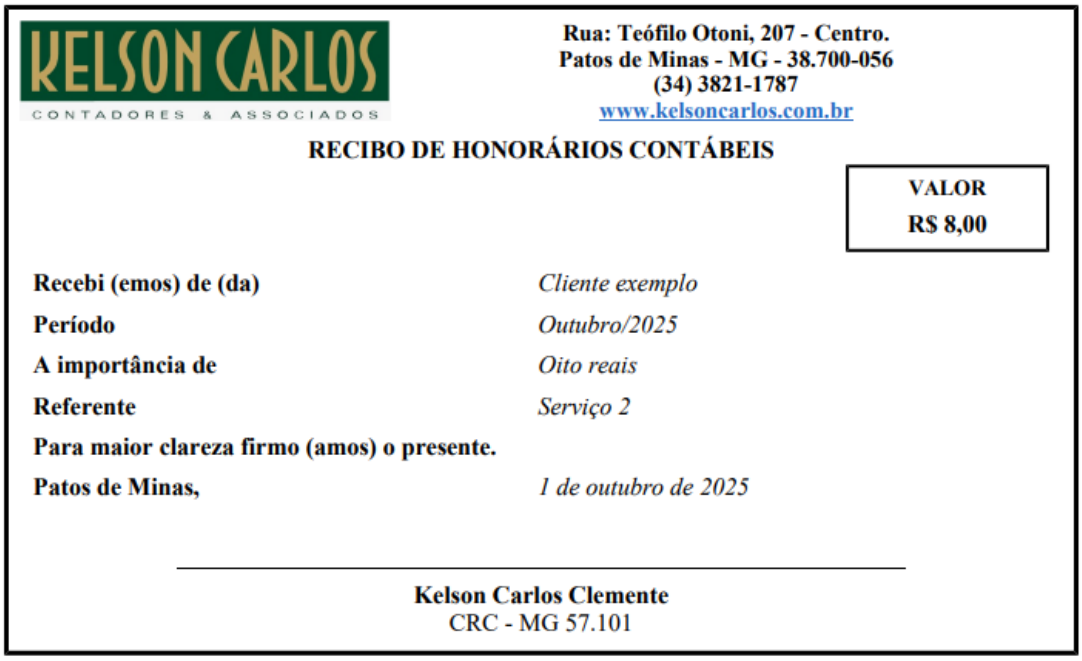
Figura 12: Interface de gerenciamento de cobranças



Fonte: elaborado pelos autores, 2025.

O modelo do recibo foi disponibilizado pelo escritório de contabilidade no formato de planilha, que foi posteriormente convertido em *template* HTML. A partir da conversão, a funcionalidade de impressão dos recibos foi desenvolvida utilizando a biblioteca *iText*, uma ferramenta disponível para diversas tecnologias que disponibiliza métodos para a criação e manipulação de documentos PDF de forma programática. A Figura 13 demonstra um exemplo de um recibo gerado para uma cobrança utilizando essa biblioteca para converter o *template* HTML em um documento PDF.

Figura 13: Exemplo de PDF com recibo de cobrança


KELSON CARLOS
CONTADORES & ASSOCIADOS

Rua: Teófilo Otoni, 207 - Centro.
Patos de Minas - MG - 38.700-056
(34) 3821-1787
www.kelsoncarlos.com.br

RECIBO DE HONORÁRIOS CONTÁBEIS

VALOR
R$ 8,00

Recebi (emos) de (da)	*Cliente exemplo*
Período	*Outubro/2025*
A importância de	*Oito reais*
Referente	*Serviço 2*

Para maior clareza firmo (amos) o presente.

Patos de Minas, *1 de outubro de 2025*

Kelson Carlos Clemente
CRC - MG 57.101

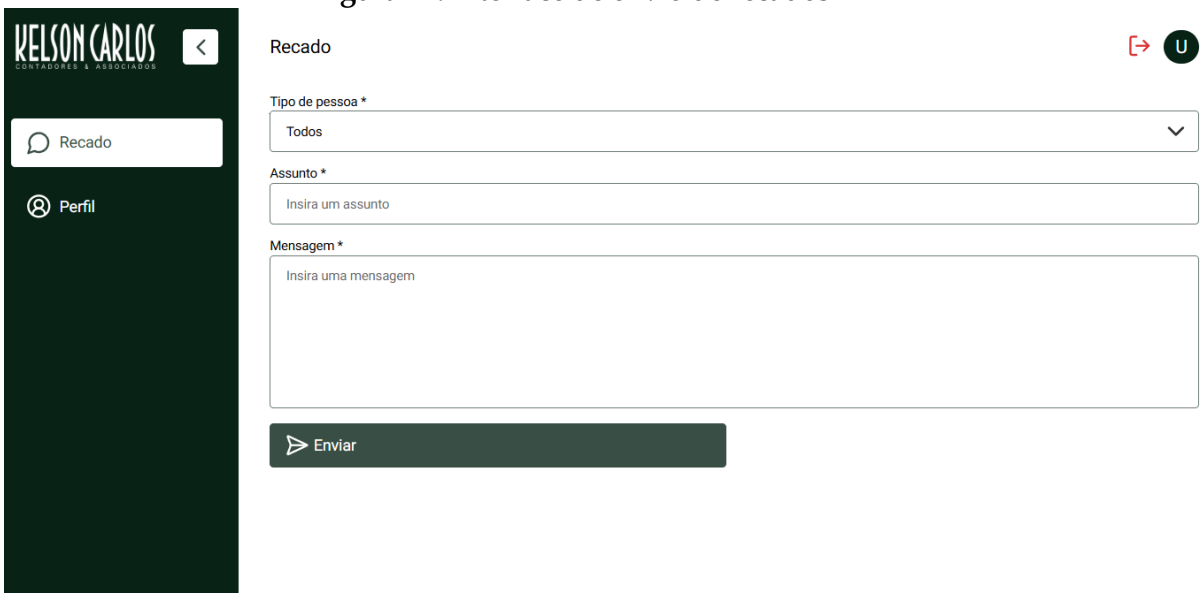
Fonte: elaborado pelos autores, 2025.

Ao usuário administrador também são disponibilizadas funcionalidades para o gerenciamento dos usuários do sistema. Essas funcionalidades são semelhantes ao gerenciamento de clientes e serviços prestados, de forma que, ao acessar a opção "Usuários" na barra de navegação, o administrador pode listar os usuários cadastrados (com exceção do usuário autenticado), aplicar filtros para facilitar a busca, bem como editar, inativar ou criar novos usuários.

Na funcionalidade de criação ou edição de um usuário, o processo dessa vez possui apenas uma etapa. Nessa etapa, o administrador poderá preencher o formulário com as informações básicas do usuário, como nome, perfil (administrador ou colaborador) e *e-mail*. No caso de um novo cadastro, será enviado um *e-mail* para o novo usuário informando a criação do novo acesso com uma senha aleatória.

Com isso, foram apresentadas todas as funcionalidades exclusivas do usuário administrador. O usuário colaborador, por sua vez, possui apenas duas funcionalidades, compartilhadas com o administrador. A primeira, ilustrada na Figura 14, refere-se ao envio de recados por e-mail aos clientes, permitindo ainda aplicar filtros por tipo de pessoa (física ou jurídica) conforme a necessidade.

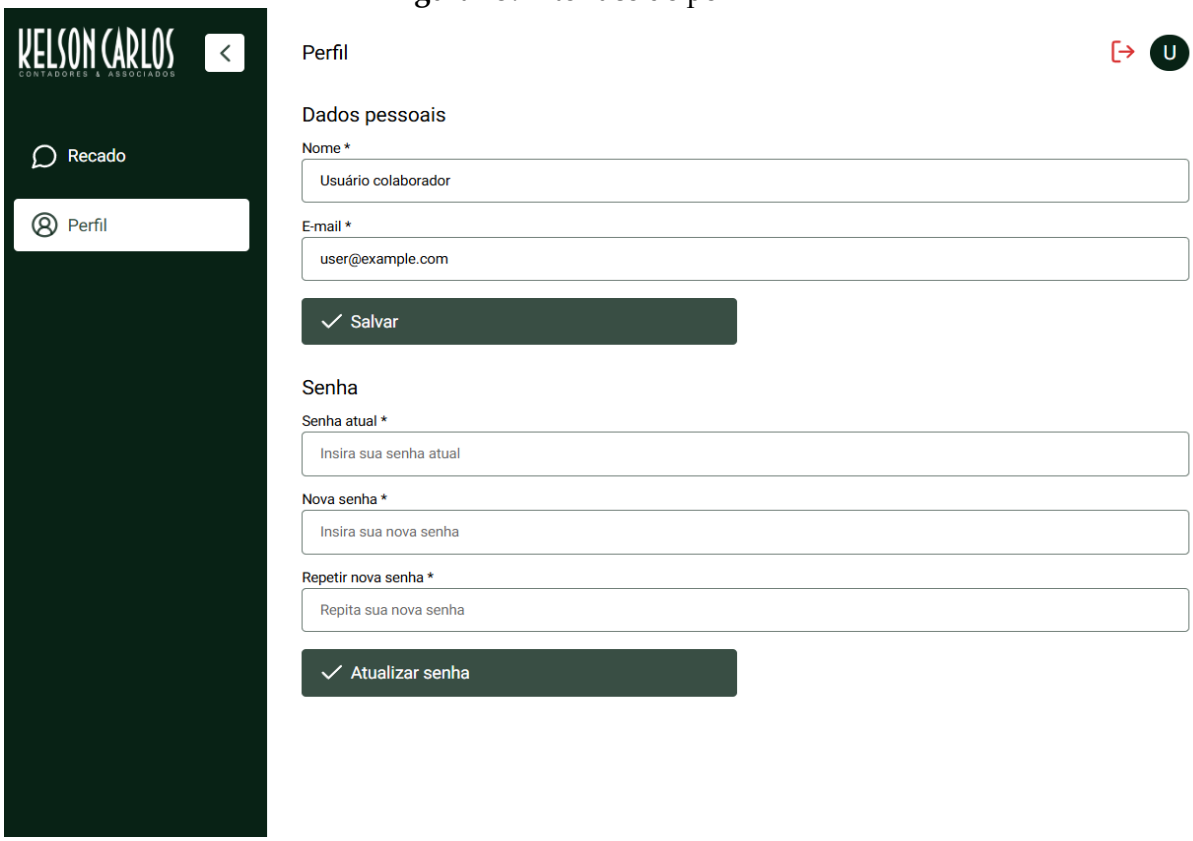
Figura 14: Interface de envio de recados



Fonte: elaborado pelos autores, 2025.

Por fim, a segunda funcionalidade compartilhada entre os usuários administrador e colaborador no sistema é a funcionalidade de perfil, apresentada na Figura 15, na qual os usuários poderão alterar as suas próprias informações.

Figura 15: Interface de perfil



Fonte: elaborado pelos autores, 2025.

Na interface são apresentados dois formulários para atualização do cadastro. Um deles possui a responsabilidade de atualizar os dados pessoais do usuário como nome e *e-mail*; já o outro formulário tem a função de atualização da senha utilizada para acessar o sistema.

5 CONCLUSÃO

Com base em todas as funcionalidades apresentadas e descritas ao longo deste artigo, é possível afirmar que todos os objetivos propostos foram plenamente alcançados. O sistema desenvolvido contempla todas as operações previstas, como o gerenciamento completo de clientes, serviços prestados, cobranças e usuários. Além disso, foram implementadas as funcionalidades específicas destinadas aos diferentes perfis de acesso (administrador e colaborador), garantindo o correto controle de permissões e a usabilidade esperada no contexto do escritório de contabilidade.

Para o escritório de contabilidade, o sistema representará um diferencial significativo no desempenho das atividades diárias, pois automatiza processos que são realizados manualmente. Essa automação contribui para uma gestão mais eficiente dos clientes e proporciona aos colaboradores mais tempo para se dedicarem às atividades contábeis em si. Outro destaque é a funcionalidade de envio de recados, que aprimora a comunicação com os clientes, tornando-a mais ágil e organizada.

Como proposta de evolução do sistema, sugerem-se dois novos módulos complementares. O primeiro seria um módulo de relatórios, voltado à geração de informações gerenciais que permitam ao administrador acompanhar o desempenho da empresa, como valores cobrados, serviços mais contratados e número de clientes ativos. Esse recurso proporcionaria uma visão analítica dos dados, auxiliando na tomada de decisões estratégicas. O segundo seria um portal do cliente, que ofereceria um ambiente de autoatendimento para que os clientes possam acessar suas cobranças, recibos e serviços contratados, além de visualizar os comunicados recebidos. Juntos, esses módulos ampliariam a funcionalidade e o valor da aplicação, fortalecendo a eficiência operacional e o relacionamento com os clientes.

Em suma, o sistema desenvolvido não apenas atingiu todos os objetivos propostos, mas também serviu como base para identificar oportunidades de evolução que podem elevar ainda mais a qualidade e a usabilidade da aplicação em futuras versões.

REFERÊNCIAS

BOFF, Giovani de Souza *et al*. **Serviço para gerenciamento de identidades e acessos baseado em Arquitetura Limpa e Microsserviços**. Trabalho de Conclusão de Curso (Bacharel em Tecnologia em Sistemas Para Internet) - Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul, Porto Alegre, 2023.

BOUSSIENGUI, Pedro Henrique Wilfride de Lima. **Proposta de arquitetura e desenvolvimento de API do sistema Bot Educacional utilizando os princípios de Domain-Driven Design e Arquitetura Limpa**. Trabalho de Conclusão de Curso (Bacharel em Engenharia de Computação) - Universidade Federal do Rio Grande do Norte, Natal, 2023.

CAPISTRANO, Lucimara Maranhão. **O papel do contador**. Trabalho de Conclusão de Curso (Bacharel em Ciências Contábeis) - Universidade Federal de Santa Catarina, Florianópolis, 2001.

FONSECA, Reinaldo Aparecida *et al*. A importância do Contador nas organizações. In: XI SIMPÓSIO DE EXCELÊNCIA EM GESTÃO E TECNOLOGIA (SEGeT), 2014, Resende. **Anais** [...]. Resende: AEDB, 2014. Disponível em: https://www.aedb.br/seget/arquivos/artigos14/32720337.pdf.

GOMES, Jéssica Araldi *et al*. Sistema (Enterprise Resource Planning) ERP na contabilidade. **eHumanit@s**, Araçatuba, São Paulo, p. 228-242, 2022.

MASSE, Mark. **REST API *design rulebook***. O'Reilly Media, Inc., 2011.

OUTLAW, Michael. **Clean Architecture:** Make Your Architecture Scream. 2018. Disponível em: https://www.codingblocks.net/podcast/clean-architecture-make-your-architecture-scream/.

PAUTASSO, Cesare; ZIMMERMANN, Olaf; LEYMANN, Frank. **Restful web services vs. "big" web services**: making the right architectural decision. Proceedings of the 17th international conference on World Wide Web, 2008. p. 805-814.

POSSEBON, Cláudia; RIBEIRO, Wagner de Souza. **Sistemas de informação gerencial**. São Paulo: Editora Senac, 2020.

PRÓVIO. **Os 10 maiores desafios dos profissionais da área de contabilidade**. LinkedIn, 24 jul. 2023. Disponível em: https://shre.ink/Mdef.

SILVEIRA, Luis André da. **Um estudo sobre o desenvolvimento de produtos digitais acessíveis dos pontos de vista gerencial e estratégico das organizações**. 2023. Dissertação (Mestrado) - Universidade de São Paulo, São Paulo, 2023. Disponível em: https://teses.usp.br/teses/disponiveis/100/100131/tde-04122023-235404/.

UNICESUMAR. **Qual a atuação do contador e suas responsabilidades em uma empresa?** 22 ago. 2017. Disponível em: https://shre.ink/MdeW.

VALENTE, Marco Tulio. **Engenharia de software moderna**: princípios e práticas para desenvolvimento de software com produtividade. Editora: Independente, 2020.