FlowDesk: Organizador Inteligente de Projetos para *Freelancers*

Bruno Henrique da Silva Ribeiro[1], Giovane Galvão[1]

[1]Centro Universitário Campo Real

Rua Comendador Norberto, 1299 - Santa Cruz – Guarapuava – PR – Brasil

{engs-brunoribeiro@camporeal.edu.br, prof_giovanegalvao@camporeal.edu.br}

Resumo

Diante do crescimento do trabalho autônomo impulsionado pela digitalização e pela economia sob demanda, observa-se a necessidade de ferramentas que centralizem a comunicação, o acompanhamento e a validação de entregas entre profissionais e clientes. O presente trabalho apresenta o *FlowDesk*, uma plataforma *SaaS (Software as a Service)* voltada à gestão de projetos para *freelancers* criativos, especialmente profissionais da área de *design*. O sistema propõe um ambiente unificado que otimiza a organização e a eficiência operacional de *freelancers*, reunindo em uma única plataforma funcionalidades como gestão de clientes, projetos e tarefas, controle financeiro com divisão de receita entre colaboradores, sistema de *briefings*, editor de propostas comerciais com geração de PDF, rastreamento de tempo, portal exclusivo para o cliente e relatórios de desempenho. O desenvolvimento fundamentou-se nos princípios da Engenharia de Software, nas metodologias ágeis e no *Design* Centrado no Usuário (DCU), garantindo que a experiência do usuário fosse o foco em todas as etapas. Para a construção da aplicação, foram utilizadas tecnologias modernas como *Next.js*, *React*, *TypeScript* e *Supabase*, oferecendo alto desempenho, segurança e integração entre as camadas *front-end* e *back-end*. A avaliação do sistema compreendeu uma análise heurística fundamentada nas dez heurísticas de Nielsen, complementada por testes de usuários reais. Os resultados demonstram que o *FlowDesk* contribui para a melhoria dos processos de gestão e comunicação no mercado *freelancer*, oferecendo uma solução eficiente, acessível e adaptada às demandas da economia criativa.

Palavras-chave: *Saas*. *Freelancers*. Gestão de projetos. *Design*.

Abstract

Faced with the growth of freelance work driven by digitalization and the on-demand economy, there is a clear need for tools that centralize communication, tracking, and delivery validation between professionals and clients. This paper presents FlowDesk, a SaaS (Software as a Service) platform designed for project management tailored to creative freelancers, especially professionals in the design field. The system proposes a unified environment that optimizes the organization and operational efficiency of freelancers, bringing together in a single platform features such as client, project, and task management, financial control with revenue sharing among collaborators, a briefing system, a commercial proposal editor with PDF generation, time tracking, an exclusive client portal, and performance reports. Development was grounded in the principles of Software Engineering, agile methodologies, and User-Centered Design (UCD), ensuring that user experience remained the core focus throughout all stages. For the construction of the application, modern technologies such as Next.js 15, React 18, TypeScript, and Supabase were used, providing high performance, security, and seamless integration between front-end and back-end layers. System evaluation comprised a heuristic analysis grounded in Nielsen's ten usability heuristics, complemented by tests with real users. Results demonstrate that FlowDesk contributes to improving management and communication processes in the freelance market, offering an efficient, accessible, and adaptable solution aligned with the demands of the creative economy.

Keywords: SaaS. Freelancers. Project management. Design.

1. Introdução

O mercado de trabalho passou por uma transformação acelerada nos últimos anos, impulsionada pela digitalização e intensificada pela pandemia de COVID-19, que consolidou modelos flexíveis de atuação e ampliou o volume de profissionais atuando de forma independente. Nesse processo, a economia criativa tornou-se um dos segmentos de maior relevância nacional. De acordo com o Mapeamento da Indústria Criativa 2025 da FIRJAN, o PIB (Produto Interno Bruto) criativo brasileiro atingiu R$393,3 bilhões em 2023, representando 3,59% do PIB do país, o maior valor já registrado na série histórica. O estudo também destaca que os setores de

Consumo, que reúnem *design*, publicidade, moda e arquitetura, concentram quase metade dos empregos criativos formais no país.

O crescimento também se manifesta no mercado global: a Grand View Research (2024) estima que as plataformas de *freelancers* devem alcançar US$ 14,39 bilhões até 2030, enquanto a Upwork (2025) aponta que 28% dos trabalhadores de conhecimento nos EUA já atuam de forma independente. No Brasil, a FIRJAN destaca que a área de *design* ultrapassa 105 mil profissionais formais e um número ainda maior de autônomos, apoiando diretamente a economia digital.

Contudo, esse avanço expõe desafios recorrentes enfrentados pelos *freelancers*: falta de organização, ausência de ferramentas integradas, perda de informações, dificuldade na comunicação com clientes e necessidade constante de revisões e aprovações. A fragmentação de fluxos em planilhas, aplicativos de mensagens, e-mails e *drives* reduz a produtividade e compromete a qualidade das entregas (PRESSMAN, 2016).

Diante desse cenário, este trabalho tem como objetivo geral desenvolver o *FlowDesk*, uma plataforma *SaaS* voltada à gestão integrada de projetos para *freelancers* criativos, especialmente *designers*, centralizando *briefings*, tarefas, arquivos e comunicação em um único ambiente. Para atingir esse propósito, os objetivos específicos incluem: estruturar um fluxo de trabalho visual e intuitivo que facilite o acompanhamento das etapas do projeto; implementar recursos que promovam uma comunicação clara e organizada entre *freelancer* e cliente, reduzindo retrabalhos; e desenvolver mecanismos de armazenamento, controle e visualização de arquivos e entregas, garantindo rastreabilidade e uma experiência coesa durante todo o processo criativo.

Com base nesses objetivos, o *FlowDesk* propõe uma solução alinhada às necessidades reais dos profissionais da economia criativa. O sistema é fundamentado nos princípios da Engenharia de Software, no DCU e nas metodologias ágeis, utilizando tecnologias modernas como *Next.js*, *Node.js* e *Supabase* para oferecer uma aplicação escalável, segura e voltada à experiência do usuário. Nesse contexto, o *FlowDesk* surge como uma alternativa inovadora para otimizar o trabalho dos *freelancers*, reduzindo a complexidade operacional, melhorando a comunicação e promovendo uma gestão mais eficiente e integrada de projetos criativos.

Ao longo deste trabalho, são apresentados o contexto e a justificativa para o desenvolvimento do *FlowDesk*, seguidos do Referencial Teórico, que fundamenta as

escolhas técnicas e metodológicas adotadas. Em seguida, o Estado da Arte mapeia as soluções existentes no mercado e na literatura, identificando lacunas que reforçam a relevância da proposta. A metodologia descreve as etapas de desenvolvimento, a arquitetura do sistema e as estratégias de avaliação adotadas. Por fim, os resultados apresentam o sistema desenvolvido, suas funcionalidades, a avaliação de usabilidade realizada e os testes conduzidos, seguidos das considerações finais com as conclusões do trabalho e sugestões para trabalhos futuros.

2. Referencial Teórico

Nesta seção, são apresentados os principais conceitos, fundamentos e tecnologias que embasam o desenvolvimento do *FlowDesk*, incluindo temas como o mercado de *freelancers*, gestão de projetos e, por fim, as tecnologias empregadas na aplicação.

2.1. Mercado de *Freelancers*

O avanço da digitalização e da conectividade global impulsionou o crescimento do trabalho *freelancer* em diversos setores, especialmente nas áreas criativas e tecnológicas. Relatórios de mercado indicam que plataformas digitais voltadas à intermediação de serviços vêm apresentando crescimento contínuo, refletindo a consolidação de modelos de trabalho baseados em projetos e contratos temporários (GRAND VIEW RESEARCH, 2024). Com a expansão do trabalho remoto e da economia sob demanda, aumentou também a dependência de sistemas digitais capazes de organizar fluxos de tarefas, comunicação e entregas. No entanto, muitos profissionais utilizam ferramentas fragmentadas, como planilhas, aplicativos de mensagens e e-mails, o que compromete a centralização das informações e a rastreabilidade das atividades (PRESSMAN, 2016).

2.2. Gestão de Projetos

A gestão de projetos, no contexto do trabalho *freelancer*, envolve a organização de demandas por cliente, definição de escopo, planejamento de prazos, acompanhamento de entregas e controle de retrabalho. Diferentemente de ambientes

corporativos com equipes dedicadas, *freelancers* frequentemente acumulam funções operacionais e administrativas, o que exige métodos de organização que reduzam perdas de informação e aumentem a previsibilidade. Segundo o Project Management Institute (PMI, 2021), práticas de gerenciamento estruturado contribuem para maior controle sobre escopo, tempo e qualidade, aspectos críticos quando há múltiplos projetos em paralelo. A ausência de processos claros para validação e registro de decisões pode gerar mudanças frequentes no escopo e retrabalho, afetando produtividade e prazos (PRESSMAN, 2016).

Para apoiar esse cenário, ferramentas e práticas de acompanhamento visual e controle de fluxo são amplamente utilizadas, pois facilitam a priorização e a execução de tarefas em etapas. Abordagens como *Kanban* ajudam a representar o status do trabalho e a limitar atividades simultâneas, favorecendo foco e previsibilidade. Mecanismos de registro e rastreabilidade tornam-se essenciais para manter histórico de solicitações, *feedbacks* e versões de entregáveis, especialmente em projetos criativos onde requisitos podem evoluir ao longo do tempo. Conforme Sommerville (2011), métodos iterativos e acompanhamento contínuo contribuem para adaptação às mudanças e melhoria incremental do processo.

2.3. *Design* Centrado no Usuário (DCU)

O Design Centrado no Usuário (DCU) é uma abordagem de desenvolvimento de sistemas interativos que prioriza usuários, necessidades e contexto de uso durante todo o ciclo de vida do produto. A norma ISO 9241-210 estabelece princípios e atividades para o *design* centrado no ser humano, reforçando a importância de compreender usuários e tarefas, definir requisitos, produzir soluções e avaliar continuamente a experiência de uso (ISO, 2019). O DCU caracteriza-se por um processo iterativo, no qual descobertas obtidas em pesquisa com usuários e testes de uso retroalimentam decisões de interface e fluxo, com o objetivo de melhorar usabilidade e acessibilidade (INTERACTION DESIGN FOUNDATION, 2025).

Em aplicações de gestão com múltiplos perfis, como *freelancer* e cliente, o DCU contribui para reduzir atritos de navegação, prevenir erros e tornar as tarefas do sistema mais claras. Para apoiar a avaliação de usabilidade, são amplamente utilizadas as dez heurísticas de Nielsen (NIELSEN, 1994), que funcionam como

diretrizes gerais para identificar problemas recorrentes de interação. São elas: (1) visibilidade do *status* do sistema; (2) correspondência entre o sistema e o mundo real; (3) controle e liberdade do usuário; (4) consistência e padrões; (5) prevenção de erros; (6) reconhecimento em vez de memorização; (7) flexibilidade e eficiência de uso; (8) estética e *design* minimalista; (9) ajuda para reconhecimento, diagnóstico e recuperação de erros; e (10) ajuda e documentação (NIELSEN, 1994). Essa abordagem foi adotada na avaliação do *FlowDesk*, conforme descrito na seção de resultados (NORMAN, 2013).

2.4. Métodos Ágeis e Gestão do Fluxo de Trabalho

Métodos ágeis são amplamente utilizados para lidar com incerteza e mudanças de requisitos em projetos de *software*, priorizando entregas frequentes e aprendizado contínuo. O Manifesto Ágil enfatiza colaboração, *software* funcionando e resposta a mudanças como princípios centrais de desenvolvimento (BECK et al., 2001). Entre os *frameworks* ágeis, o *Scrum* define eventos, responsabilidades e artefatos para orientar a entrega incremental de valor, promovendo inspeção e adaptação ao longo do processo por meio de ciclos curtos e *feedback* contínuo (SCHWABER; SUTHERLAND, 2020).

Complementarmente, o *Kanban* é utilizado para visualizar o fluxo de trabalho e limitar trabalho em andamento (WIP), melhorando previsibilidade e eficiência. Anderson (2010) descreve o *Kanban* como método evolutivo, aplicável sem necessidade de mudanças radicais e orientado a melhorias contínuas no processo. A combinação dessas abordagens foi adotada no desenvolvimento do *FlowDesk*, com organização das entregas em ciclos semanais e controle visual do progresso das funcionalidades ao longo de todo o processo.

2.5. *Next.js* e Renderização Híbrida

O *Next.js* é um *framework* baseado em *React* que permite a construção de aplicações *web* com diferentes estratégias de renderização, como *Server-Side Rendering* (SSR), *Static Site Generation* (SSG) e renderização híbrida (INTUZ, 2025). A renderização no servidor reduz o tempo de carregamento inicial, pois o conteúdo é processado antes de ser enviado ao navegador, enquanto a geração estática permite

que páginas sejam pré-renderizadas em tempo de *build*, reduzindo consumo de recursos e aumentando eficiência. A combinação dessas técnicas caracteriza o modelo híbrido adotado por aplicações modernas (PAGEPRO, 2024).

Além disso, o *Next.js* utiliza arquitetura baseada em componentes reutilizáveis, promovendo padronização estrutural do código e facilitando a manutenção, testes e expansão futura do sistema. O modelo de *Pages Router*, adotado no *FlowDesk*, oferece suporte nativo às rotas de *Application Programming Interface* (API) que compõem o *back-end* complementar da aplicação, permitindo a execução de lógica *server-side* diretamente no mesmo projeto sem necessidade de um servidor separado. Essa característica contribui para maior organização e qualidade do desenvolvimento (PRESSMAN, 2016).

2.6. *Node.js* e Modelo Assíncrono

O *Node.js* é um ambiente de execução *JavaScript* baseado no motor V8 do *Google Chrome*, projetado para construção de aplicações escaláveis do lado do servidor. Sua principal característica é o modelo orientado a eventos e I/O não bloqueante (TILKOV; VINOSKI, 2010), que permite tratar múltiplas requisições simultaneamente sem criar múltiplas *threads* pesadas, tornando-o adequado para aplicações com grande volume de conexões concorrentes.

No *FlowDesk*, o *Node.js* é utilizado por meio das rotas de API do *Next.js* executadas *server-side*, favorecendo a criação de *APIs RESTful* estruturadas e facilitando a integração com bancos de dados, autenticação e serviços externos. Essa abordagem promove maior desacoplamento entre camadas e contribui para uma arquitetura modular e escalável, alinhada às boas práticas de desenvolvimento de aplicações *web* modernas (SOMMERVILLE, 2011).

2.7. Supabase

O *Supabase* é uma plataforma *open source* baseada em *PostgreSQL* que fornece banco de dados relacional, autenticação, armazenamento de arquivos e geração automática de APIs (SUPABASE, 2024). Sua proposta é oferecer uma alternativa moderna e simplificada para construção de aplicações baseadas em dados, com o *PostgreSQL* amplamente reconhecido por sua robustez e conformidade

com propriedades ACID (Atomicidade, Consistência, Isolamento, Durabilidade), garantindo integridade transacional e confiabilidade no armazenamento e recuperação de dados em aplicações multiusuário.

Um dos principais diferenciais do *Supabase* é o suporte a *Row Level Security* (RLS), que permite definir regras de acesso a dados em nível de linha, garantindo isolamento lógico das informações entre usuários. Além disso, o *Supabase Realtime* possibilita a transmissão de atualizações em tempo real ao *front-end* sem necessidade de *polling*, e o *Supabase Storage* gerencia o armazenamento de arquivos com integração nativa ao banco de dados. Essas funcionalidades tornam o *Supabase* a escolha central de *back-end* do *FlowDesk* (SUPABASE, 2024).

2.8. *Tailwind CSS* e Estilização Utilitária

O *Tailwind CSS* é um *framework* de estilização baseado em classes utilitárias que permite a construção de interfaces diretamente no *markup* HTML, reduzindo a necessidade de arquivos CSS extensos e personalizados (TAILWIND LABS, 2024). Diferentemente de *frameworks* tradicionais baseados em componentes prontos, o *Tailwind* adota uma abordagem *utility-first*, na qual pequenas classes representam propriedades específicas de estilo. O sistema de configuração permite definição de *design tokens*, como paletas de cores, tipografia e *breakpoints* responsivos, promovendo alinhamento entre identidade visual e estrutura técnica e facilitando a manutenção evolutiva do projeto.

Outro aspecto relevante é a integração do *Tailwind* com ferramentas modernas de *build*, permitindo remoção automática de classes não utilizadas (tree-shaking), o que reduz significativamente o tamanho final do arquivo CSS e melhora o desempenho da aplicação. No *FlowDesk*, o *Tailwind* é combinado com um sistema de variáveis CSS customizadas que sustenta os dez temas visuais da plataforma, garantindo consistência visual em qualquer tema selecionado sem recarregamento de página (TAILWIND LABS, 2024).

2.9. Controle de Versão e *Deploy* Contínuo

O controle de versão é prática essencial no desenvolvimento de *software*, permitindo rastrear alterações no código e manter histórico detalhado de modificações. Ferramentas como *Git* e plataformas como *GitHub* possibilitam organização de *branches* e controle estruturado de versões, enquanto práticas de integração e *deploy* contínuo (CI/CD) automatizam processos de *build* e publicação do sistema (PRESSMAN, 2016). No *FlowDesk*, o repositório é hospedado no *GitHub* com *deploy* automático configurado na *Vercel* a cada atualização no código-fonte (PAGEPRO, 2024).

A adoção dessas ferramentas contribui para estabilidade do ambiente de produção, redução de erros humanos e maior agilidade na entrega de funcionalidades. Além disso, a *Vercel* oferece suporte nativo a *Cron Jobs*, utilizado no *FlowDesk* para execução diária da rotina de limpeza automática de armazenamento. Dessa forma, o processo de desenvolvimento torna-se mais seguro, rastreável e alinhado às boas práticas de Engenharia de Software (SOMMERVILLE, 2011).

2.10. Considerações Finais do Referencial Teórico

O referencial teórico apresentou os principais conceitos que fundamentam o desenvolvimento do *FlowDesk*, abordando desde o crescimento do mercado de *freelancers* e as práticas de gestão de projetos até as tecnologias modernas empregadas na construção do sistema. Esses fundamentos sustentam a proposta de desenvolvimento de uma plataforma voltada à organização e à colaboração entre profissionais criativos e seus clientes (PMI, 2021).

A análise das tecnologias, como *Next.js*, *Node.js*, *Supabase*, *Tailwind CSS*, *GitHub* e *Vercel*, evidenciou que a adoção de ferramentas consolidadas e de código aberto favorece o desenvolvimento, a integração contínua e a escalabilidade do sistema, reforçando o alinhamento do projeto com as tendências contemporâneas de desenvolvimento *web* (PRESSMAN, 2016). A próxima seção apresenta o Estado da Arte, mapeando lacunas, tendências e oportunidades de inovação para o *FlowDesk* dentro do mercado.

3. Estado da Arte

Nesta seção, realiza-se um mapeamento sistemático de plataformas digitais e ferramentas *SaaS* direcionadas ao gerenciamento de projetos para *freelancers* da economia criativa e do design. Busca-se investigar as alternativas de mercado, analisando seus recursos técnicos, funções e critérios de usabilidade, bem como as lacunas que justificam o desenvolvimento do *FlowDesk*.

A maior contribuição deste mapeamento é oferecer uma visão ampla sobre os avanços e limitações das ferramentas atuais de gestão *freelancer*, analisando como elas abordam colaboração, *feedback* visual, controle de tarefas e integração cliente-profissional. O método de mapeamento sistemático foi adotado por se tratar de uma abordagem estruturada e rigorosa de revisão da literatura, que permite avaliar a abrangência, o escopo e a concentração de estudos em um domínio específico (PETERSON et al., 2008; KITCHENHAM; CHARTERS, 2007). As questões de pesquisa definidas, a *string* de busca utilizada e os critérios de inclusão e exclusão aplicados estão detalhados no Apêndice A.

3.1. Análise dos Resultados

A aplicação da *string* de busca no *Google Scholar*, restrita ao período 2019-2025, retornou 1.720 artigos inicialmente identificados. Ao aplicar o filtro de idioma (português), restaram 40 artigos para análise.

Em seguida foram aplicados os demais critérios de inclusão, artigos completos, alinhamento com gestão de projetos para *freelancers*, foco em *SaaS* ou sistema *web*, e abordagem de colaboração, UX ou DCU, resultando em 5 artigos selecionados.

Por fim, ao aplicar os critérios de exclusão, foram eliminados 2 artigos por duplicidade ou ausência de aplicação prática relacionada ao escopo do trabalho, resultando em 3 artigos para análise final. Embora nenhum deles tratasse especificamente da gestão de projetos para *freelancers* criativos, apresentavam abordagens relacionadas em aspectos pontuais, permitindo responder às questões de pesquisa definidas.

QP1: Quais soluções tecnológicas voltadas à gestão de projetos para *freelancers* criativos foram identificadas nos últimos anos?

Os estudos encontrados abordam ferramentas genéricas de gestão de tarefas e plataformas colaborativas, como *Trello*, *Asana* e *Monday.com*, mas não apresentam soluções direcionadas especificamente a *freelancers* de *design*. Nenhum dos trabalhos descreve o acompanhamento de projetos entre *freelancer* e contratante desde a etapa de *briefing* até a conclusão, com relatórios financeiros e gráficos de desempenho, como a proposta pelo *FlowDesk*. Essa ausência confirma uma lacuna de especialização em soluções voltadas a *freelancers* criativos, cujo fluxo de trabalho é relativamente diferente do ambiente corporativo tradicional.

QP2: Quais dessas soluções abordam colaboração direta e *feedback* entre cliente e profissional dentro da plataforma?

Foi identificado que, nos sistemas analisados, a comunicação cliente-profissional e acompanhamento de projetos é geralmente tratada por canais externos (e-mail, *WhatsApp*), sem integração direta à plataforma de gestão. Assim, conclui-se que a colaboração estruturada, como a proposta pelo *FlowDesk*, com comentários visuais, notificações automáticas e histórico de interações, ainda é um ponto negligenciado nos artigos nacionais.

QP3: Quais tecnologias e arquiteturas são mais utilizadas na implementação dessas plataformas?

Embora os artigos não apresentem detalhamento técnico completo, foi possível identificar tendência de uso de *frameworks* modernos como *React*, *Node.js* e *Firebase*, além de arquiteturas baseadas em *APIs REST* e serviços em nuvem. Nenhum estudo, contudo, menciona o *Supabase* ou integrações específicas com ferramentas de automação e armazenamento inteligente.

QP4: Quais lacunas ainda existem no mercado e na literatura em relação a sistemas voltados para *freelancers* criativos?

O mapeamento evidenciou três lacunas principais no cenário analisado. A primeira é uma lacuna temática: os estudos encontrados não abordam a realidade específica dos *freelancers* criativos no contexto de gestão de projetos, tratando o tema de forma genérica ou voltada a ambientes corporativos tradicionais.

A segunda é uma lacuna metodológica: não foram identificadas pesquisas com avaliações práticas de usabilidade ou estudos sobre experiência de uso em plataformas *SaaS* voltadas a profissionais autônomos do setor criativo.

A terceira é uma lacuna tecnológica: as soluções existentes não integram, em um único ambiente, gestão de projetos, controle financeiro, *feedback* contextual e automação de armazenamento, funcionalidades que compõem o núcleo do *FlowDesk*. Essa análise confirmou a originalidade e a relevância da proposta desenvolvida neste trabalho.

3.2. Considerações Finais do Mapeamento Sistemático

O mapeamento sistemático realizado demonstra que a produção científica brasileira ainda é principiante no campo da gestão de projetos voltada a *freelancers* criativos. A ausência de publicações diretamente relacionadas, somada à pouca presença de estudos sobre usabilidade e experiência do usuário em sistemas desse tipo, reforça a pertinência e originalidade do desenvolvimento do *FlowDesk*. A análise indicou que as soluções disponíveis são, em sua maioria, ferramentas genéricas de gestão de tarefas, sem especialização no contexto criativo, destacando-se a ausência de soluções que integrem simultaneamente gestão de projetos, finanças, *feedback* contextual e automação de armazenamento.

Dessa forma, o *FlowDesk* se posiciona como uma resposta direta às lacunas identificadas, oferecendo um modelo *SaaS* específico para *freelancers* criativos com foco em *design* e comunicação visual, um fluxo de trabalho que une colaboração estruturada e transparência entre cliente e profissional, e a aplicação de métodos de usabilidade e UX visando contribuir para a literatura científica nacional sobre o tema. O mapeamento confirma que o *FlowDesk* é não apenas uma proposta de valor tecnológico, mas também uma contribuição acadêmica relevante, capaz de preencher uma lacuna na pesquisa sobre plataformas digitais para gestão de projetos para *freelancers* no Brasil.

4. Metodologia

A metodologia adotada neste trabalho tem como finalidade descrever de forma estruturada o processo de desenvolvimento da solução proposta, detalhando as etapas executadas, as tecnologias empregadas e os procedimentos utilizados para implementação e validação do sistema. A organização metodológica foi construída com base nos princípios da Engenharia de Software e do DCU, buscando assegurar

clareza, reprodutibilidade e rigor técnico, com foco contínuo nas necessidades dos perfis envolvidos (*freelancer* e cliente).

Por se tratar do desenvolvimento de um sistema computacional aplicado a um problema real, o trabalho exigiu a definição clara de requisitos, arquitetura, modelagem de dados e estratégias de testes. Cada decisão técnica foi fundamentada na literatura e alinhada às boas práticas de desenvolvimento de aplicações *web* modernas, garantindo escalabilidade, segurança e desempenho.

O desenvolvimento foi conduzido de forma incremental, inspirado nos princípios do *Scrum* e do *Kanban*, com organização das entregas em ciclos semanais que permitiram a implementação progressiva das funcionalidades, validação contínua e ajustes estruturais ao longo do processo. O *backlog* do produto foi definido previamente, listando todas as funcionalidades planejadas, e a cada ciclo um conjunto de funcionalidades era priorizado e desenvolvido até sua conclusão antes de avançar para o próximo incremento.

O desenvolvimento foi organizado em quatro grandes incrementos. O primeiro incremento contemplou a base estrutural do sistema: configuração da arquitetura *Next.js* com *Pages Router*, integração com o *Supabase*, implementação do fluxo completo de autenticação com JWT, *onboarding* inicial e a estrutura do banco de dados com as primeiras políticas de *Row Level Security*. O segundo incremento abrangeu os módulos centrais de gestão: clientes, projetos com *kanban* visual, tarefas e subtarefas, arquivos, *links* e o *feed* de atividades, além da *dashboard* principal com os *cards* de métricas. O terceiro incremento incorporou os módulos de maior complexidade: *briefings* com templates e resposta pública pelo cliente, editor de propostas comerciais com geração de PDF via *Puppeteer*, controle financeiro com registro de pagamentos e cobrança ao cliente, *time tracker* com cronômetro global e por tarefa, e o sistema de *co-working* com convites, *splits* financeiros e *dashboard* do colaborador. O quarto e último incremento foi dedicado ao portal do cliente, ao sistema de planos com integração ao *Stripe*, ao gerenciamento de armazenamento com *Cron Job* de limpeza automática, ao módulo de relatórios com cinco painéis analíticos e ao sistema de dez temas visuais.

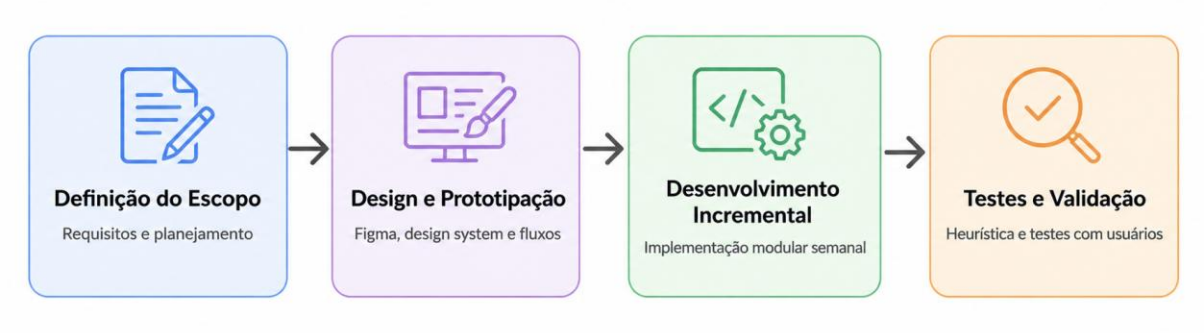
Do ponto de vista ágil, foram adotadas práticas do *Scrum* como a definição de um *backlog* priorizado, a organização do trabalho em ciclos com escopo delimitado e a revisão contínua das entregas ao final de cada incremento. O *Kanban* foi utilizado

como método de visualização e controle do fluxo de trabalho, limitando as funcionalidades em desenvolvimento simultâneo e tornando o progresso visível ao longo de todo o processo. Essa combinação permitiu que o sistema fosse construído de forma organizada, com foco na entrega de valor funcional a cada ciclo, adaptando-se às necessidades identificadas ao longo do desenvolvimento (SCHWABER; SUTHERLAND, 2020; ANDERSON, 2010).

4.1. Etapas do Desenvolvimento

A Figura 1 ilustra, em alto nível, as quatro etapas que orientaram o desenvolvimento do *FlowDesk*. A primeira etapa compreendeu a definição do escopo do sistema, com o levantamento dos requisitos funcionais e não funcionais e a delimitação das funcionalidades a serem implementadas. A segunda etapa envolveu a construção do *design system* no *Figma*, incluindo a definição de variáveis de cor, tipografia, componentes visuais, logotipos e prototipação dos principais fluxos de telas. A terceira etapa correspondeu ao desenvolvimento incremental da aplicação, conduzido de forma modular e organizado em ciclos semanais, permitindo a implementação progressiva das funcionalidades com validação contínua. A quarta etapa abrangeu os testes e a validação do sistema, por meio de avaliação heurística com base nas dez heurísticas de Nielsen e testes práticos com usuário real.

Figura 1- Etapas do Desenvolvimento do *FlowDesk.*



Fonte: O autor (2026).

4.2. Arquitetura e Modelagem

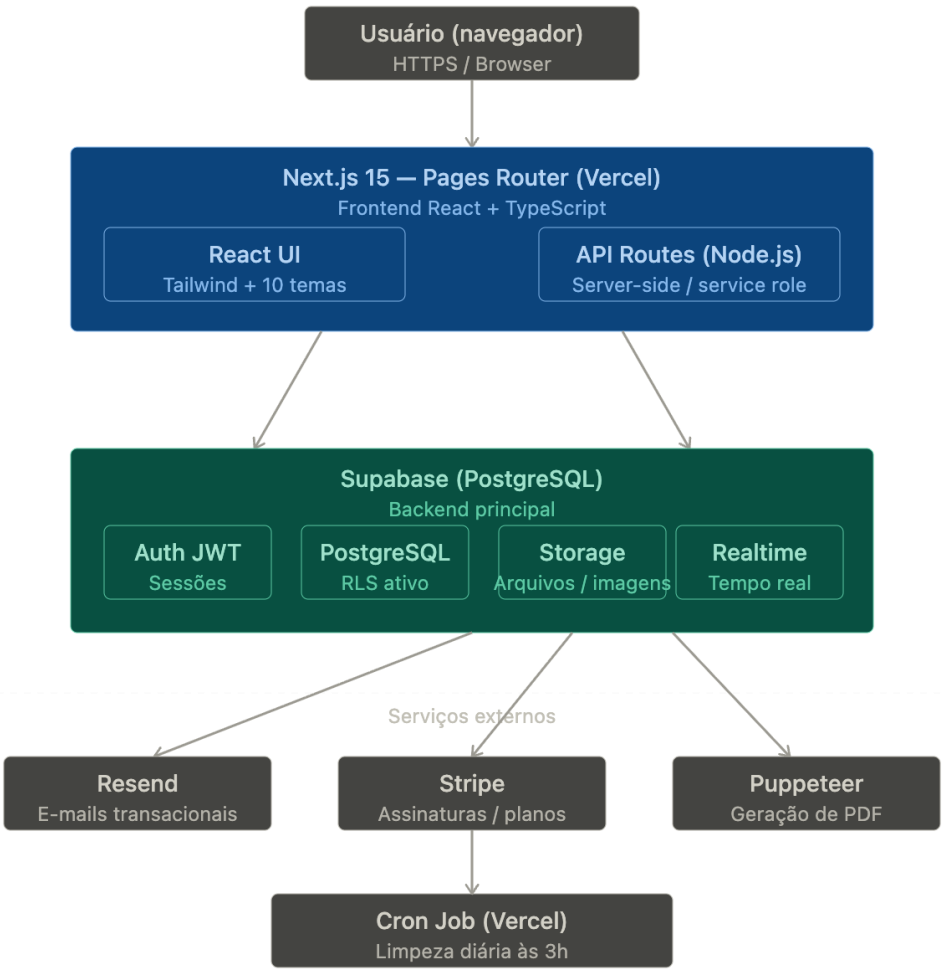
A arquitetura adotada baseia-se na separação entre camada de apresentação e camada de persistência de dados, integradas por meio de requisições estruturadas. O *front-end* foi desenvolvido utilizando *Next.js*, *framework* baseado em *React*, permitindo renderização híbrida e organização modular por componentes reutilizáveis. A camada de *back-end* e persistência foi estruturada utilizando *Node.js* em conjunto com *Supabase*, plataforma baseada em *PostgreSQL*, possibilitando autenticação integrada, gerenciamento relacional de dados e aplicação de políticas de segurança em nível de registro por meio de *Row Level Security* (RLS).

A modelagem do banco de dados foi realizada com base no modelo relacional, estruturando entidades principais como usuários, projetos e tarefas, com relacionamentos definidos por meio de chaves estrangeiras, garantindo integridade referencial e consistência estrutural entre as tabelas. Cada projeto está vinculado a um usuário autenticado, assegurando isolamento das informações por meio de políticas de controle de acesso. O *deploy* foi realizado na *Vercel* com integração contínua via *GitHub*, permitindo publicação automatizada e garantindo disponibilidade contínua da aplicação.

5. Resultados e Discussões

O desenvolvimento do *FlowDesk* resultou em uma plataforma *SaaS* completa, estruturada em módulos interdependentes que cobrem todas as etapas do fluxo de trabalho de um *freelancer* criativo: da prospecção de clientes ao recebimento financeiro. O sistema encontra-se em produção e acessível pelo domínio app.oflowdesk.com. A Figura 2 apresenta a arquitetura geral do sistema, ilustrando a separação entre camadas e a integração entre os serviços utilizados. Os demais diagramas do sistema, casos de uso, sequência e classes, estão detalhados no Apêndice C.

Figura 2 - Diagrama de componentes do *FlowDesk.*



Fonte: O autor (2026).

Do ponto de vista técnico, o *FlowDesk* é uma aplicação *web* desenvolvida com *Next.js* no modelo *Pages Router*, *React* e *TypeScript* na camada de apresentação, e *Supabase* como plataforma principal de *back-end*, fornecendo banco de dados *PostgreSQL*, autenticação JWT, controle de acesso por *Row Level Security* (RLS), armazenamento de arquivos e comunicação em tempo real via *Supabase Realtime*. O *back-end* complementar é composto por rotas de API executadas *server-side* no ambiente *Node.js* da *Vercel*, responsáveis por operações que exigem permissões administrativas ou lógica de negócio mais complexa, como o sistema de convites, a sincronização de *splits* financeiros e a geração de PDFs via *Puppeteer*, garantindo que dados sensíveis nunca sejam expostos ao cliente. O banco de dados é composto por vinte e uma tabelas relacionais, todas com RLS ativo e funções

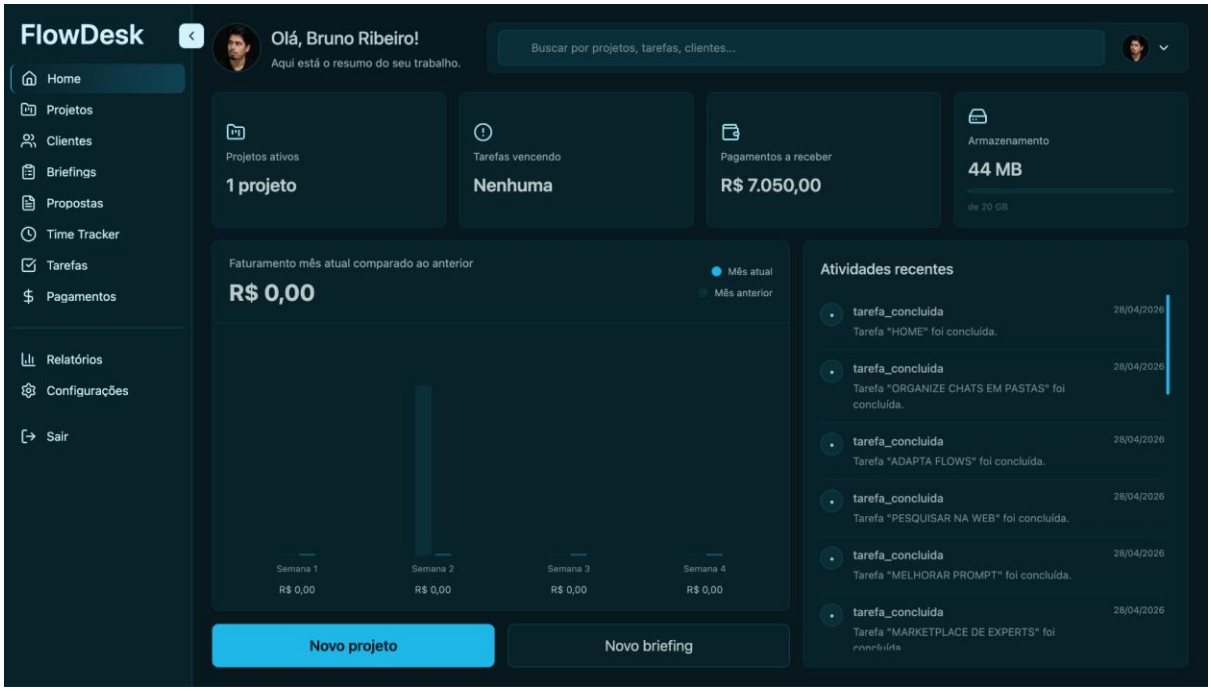
auxiliares em *PostgreSQL*, *is_project_owner()*, *is_project_member()* e *can_accept_project_invite()*, que encapsulam as verificações de permissão. A estilização da interface foi construída com *Tailwind CSS* e um sistema de variáveis CSS customizadas, suportando dez temas visuais completos, alternáveis dinamicamente sem recarregamento de página.

5.1. Funcionalidades Implementadas

5.1.1. *Dashboard* Principal

A *dashboard* é a tela central do sistema após o *login*, oferecendo ao *freelancer* uma visão consolidada do negócio em tempo real, apresentada na Figura 3.

Figura 3 - *Dashboard* principal do sistema.



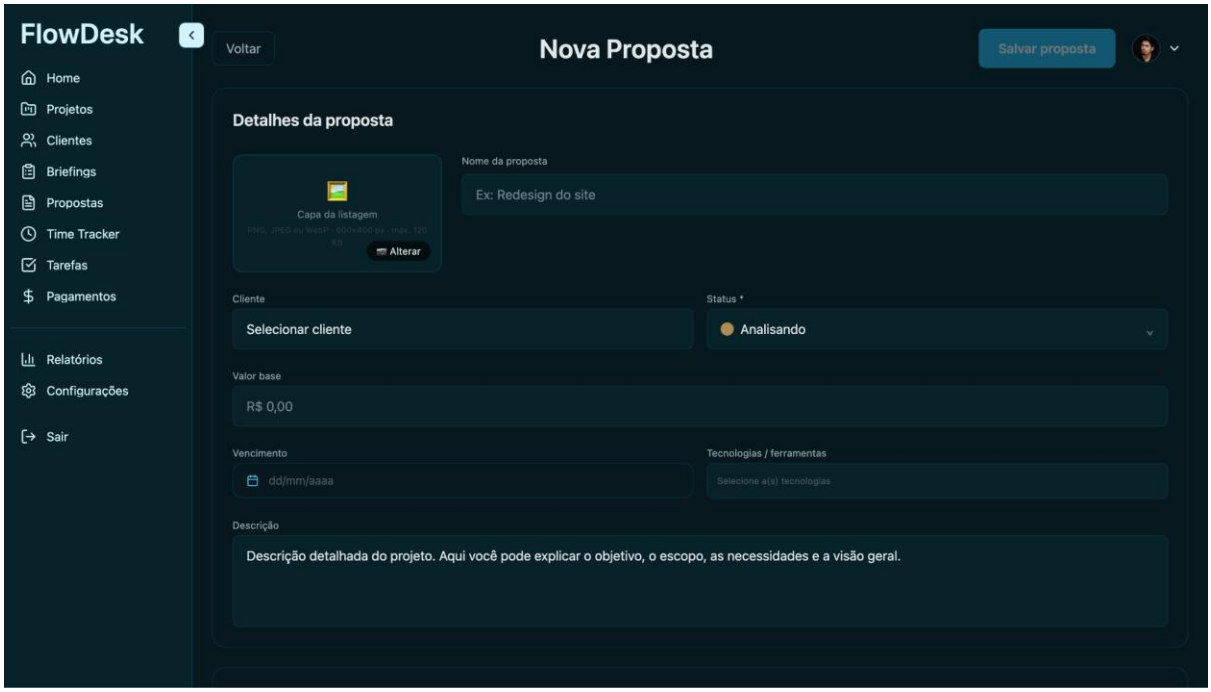
Fonte: O autor (2026).

A tela é composta por quatro *cards* de métricas: projetos ativos, tarefas vencendo, pagamentos a receber e uso de armazenamento. O *card* de armazenamento exibe uma barra de progresso colorida que varia do verde ao vermelho conforme o consumo, com *cache* local de cinco minutos para carregamento instantâneo. Abaixo dos *cards*, a *dashboard* apresenta um gráfico de barras dividido pelas semanas do mês, comparando o faturamento do mês atual com o faturamento do mês anterior e um *feed* de atividades com o histórico de ações realizadas no sistema, como criação de tarefas, envio de *briefings* e registro de pagamentos.

5.1.2. Propostas Comerciais

O *FlowDesk* oferece um editor completo de propostas comerciais com geração de PDF. O *freelancer* pode partir de três templates pré-configurados, adicionando apenas os detalhes da proposta, que é preenchida de forma automática no *template* escolhido. Após inserir os detalhes da proposta e personalizar cores, logos e capa, o *freelancer* pode editar os textos da proposta para melhor se adequar ao projeto em questão, na Figura 4 é apresentada a tela de criação de proposta comercial.

Figura 4 - Tela de criação de proposta comercial.



Fonte: O autor (2026).

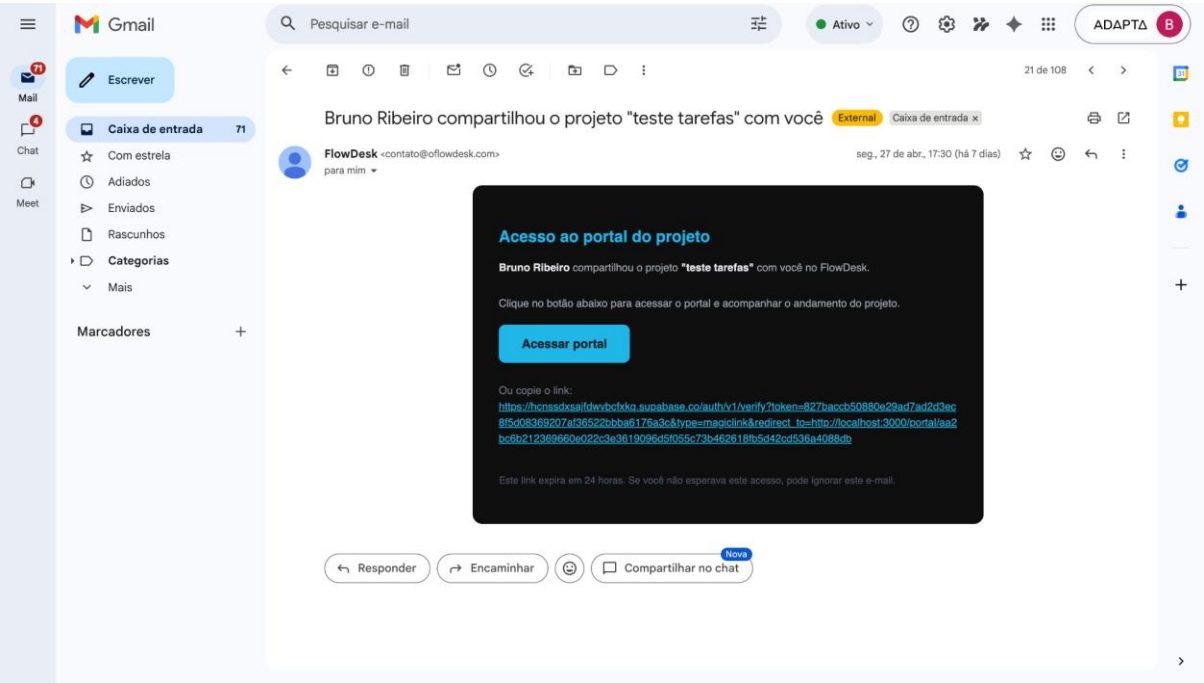
A edição de texto é realizada pelo editor *rich text TipTap*, com suporte a negrito, itálico, sublinhado, títulos, listas, alinhamento, *links* e destaques. A geração do PDF é realizada *server-side* via *Puppeteer*, garantindo fidelidade visual entre o editor e o documento exportado. A proposta pode ser enviada ao cliente diretamente pelo sistema ou exportada como arquivo PDF para envio externo. As demais funcionalidades do sistema, incluindo autenticação, gestão de clientes e projetos, *briefings*, controle financeiro, *co-working*, tarefas, *time tracker*, relatórios e o sistema de planos e armazenamento, estão detalhadas no Apêndice B.

5.2. Portal do Cliente

O portal do cliente é um ambiente exclusivo e separado da interface principal do *FlowDesk*, desenvolvido para oferecer ao cliente do *freelancer* uma visão organizada e transparente do andamento do projeto, sem que seja necessário criar uma conta na plataforma. O acesso é realizado por meio de um *link* com *token* de acesso único, gerado pelo *freelancer* a partir da página do projeto.

O *freelancer* convida o cliente diretamente pela plataforma. O sistema gera um *token* criptográfico único e envia um e-mail ao cliente via *Resend* com o *link* de acesso ao portal, conforme apresentado na Figura 5. O cliente acessa o *link*, realiza um *login* simplificado informando apenas seu e-mail e o *token* recebido, e passa a navegar livremente pelo portal sem necessidade de cadastro ou senha. Esse modelo garante segurança no acesso sem impor barreiras de uso ao cliente, que muitas vezes não possui familiaridade com plataformas digitais de gestão.

Figura 5 - Convite de acesso ao portal do cliente.

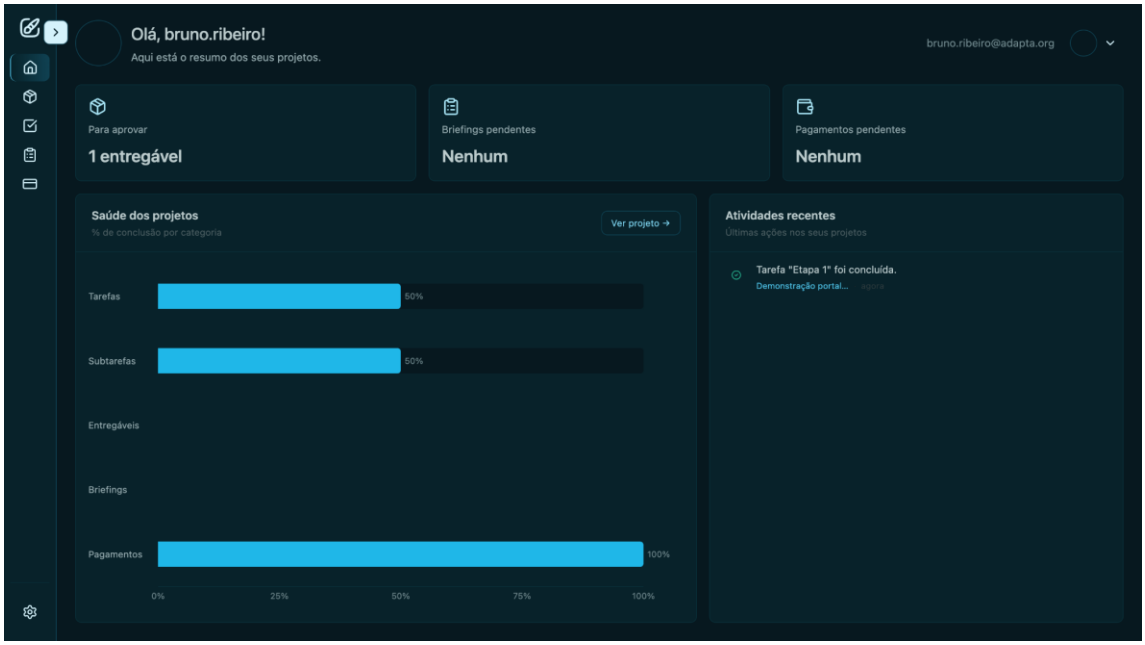


Fonte: O autor (2026).

Após o *login*, o cliente é direcionado a uma *dashboard* que apresenta uma visão geral de todos os seus projetos vinculados, com o *status* atualizado de cada um e notificações de cobranças pendentes, conforme apresentado na Figura 6. A partir da *dashboard*, o cliente pode acessar a lista completa de seus projetos e os arquivos

e *links* disponibilizados pelo *freelancer*, responder *briefings* diretamente pelo portal, visualizar e aprovar ou reprovar entregáveis com comentário escrito, e acompanhar cobranças pendentes com todos os detalhes necessários para o pagamento, incluindo valor, data prevista e chave PIX cadastrada pelo *freelancer*.

Figura 6 - *Dashboard* do cliente.



Fonte: O autor (2026).

O portal do cliente representa um diferencial importante do *FlowDesk* em relação às ferramentas genéricas de gestão de projetos disponíveis no mercado, que raramente oferecem um ambiente dedicado e acessível ao cliente final sem exigir cadastro ou licença adicional.

5.3. Avaliação Heurística

Como parte da avaliação de usabilidade do *FlowDesk*, foi realizada uma avaliação heurística com base nas dez heurísticas de usabilidade de Nielsen (NIELSEN, 1994). A técnica consiste na inspeção sistemática da *interface* por um

avaliador especialista, com o objetivo de identificar problemas de interação sem depender exclusivamente de testes com usuários finais. Cada problema encontrado foi associado à heurística correspondente e classificado por nível de severidade, seguindo a escala proposta pelo próprio Nielsen: baixa, média, alta e crítica.

A inspeção foi conduzida pelo próprio desenvolvedor do sistema, percorrendo os principais fluxos da aplicação: autenticação, criação e gestão de projetos, tarefas, *briefings*, propostas, pagamentos, *co-working* e portal do cliente. Os problemas identificados estão consolidados no Quadro 3.

Quadro 3 - Avaliação Heurística.

Problema Encontrado	**Heurística Violada**	**Severidade**	**Descrição**
Ausência de mecanismo de desfazer ações em determinados fluxos	3 - Controle e liberdade do usuário	Média	O sistema não oferece de forma generalizada a possibilidade de desfazer ações recentes, como edições em tarefas ou alterações em propostas, por meio de atalhos como Ctrl+Z ou botão de desfazer.
Terminologia do sistema de *splits* financeiros pode ser pouco familiar a alguns perfis de usuário	2 - Correspondência entre o sistema e o mundo real	Baixa	Termos como "*split*", "percentual" e "valor fixo" no contexto do *co-working*, embora corretos e amplamente utilizados no mercado criativo, podem demandar uma breve adaptação de freelancers menos familiarizados com gestão financeira.
Falta de página de ajuda e documentação centralizada	10 - Ajuda e documentação	Média	O sistema não possui uma seção de ajuda, perguntas frequentes ou documentação de uso acessível dentro da própria plataforma. Usuários que encontrem dificuldades em funcionalidades mais avançadas, não dispõem de um canal de suporte integrado para consulta.

Ausência de confirmação visual mais destacada após ações de exclusão	5 - Prevenção de erros	Média	Embora o sistema exiba modais de confirmação antes de exclusões, o *feedback* visual após a conclusão da ação poderia ser mais expressivo, com mensagens de sucesso mais destacadas que reforcem ao usuário que a operação foi realizada com êxito e que a ação é irreversível.
Onboarding inicial sem tutorial guiado	6 - Reconhecimento em vez de memorização	Baixa	Após o primeiro *login*, o usuário é direcionado diretamente à *dashboard* sem um tutorial introdutório ou tour guiado pelas principais funcionalidades.

Fonte: O autor (2026).

De forma geral, os problemas identificados na avaliação heurística apresentam severidade baixa e média, sem ocorrências de severidade alta ou crítica. Esse resultado é condizente com o cuidado dedicado à experiência do usuário ao longo do desenvolvimento, com adoção de padrões visuais consistentes, linguagem adaptada ao contexto criativo e interface organizada. Os pontos identificados indicam oportunidades de melhoria para versões futuras do sistema, especialmente no que diz respeito à documentação integrada, ao mecanismo de desfazer ações e ao *onboarding* inicial.

5.4. Testes e Validação com Usuário Real

Como etapa complementar à avaliação heurística, foi conduzido um teste de usabilidade com um usuário real, com o objetivo de validar o funcionamento do sistema em um contexto de uso cotidiano e identificar possíveis dificuldades não previstas durante o desenvolvimento. O participante é profissional da área criativa e

possui experiência com ferramentas digitais de gestão, perfil compatível com o público-alvo do *FlowDesk*.

O teste foi conduzido de forma observacional, com o participante utilizando o sistema livremente enquanto realizava um conjunto de tarefas representativas do fluxo de trabalho de um *freelancer*. As tarefas propostas foram:

- Tarefa 1: Cadastrar um novo cliente e criar um projeto vinculado a ele;
- Tarefa 2: Criar tarefas e subtarefas dentro do projeto;
- Tarefa 3: Criar e enviar um *briefing* ao cliente;
- Tarefa 4: Criar uma proposta comercial a partir de um *template* e exportar em PDF;
- Tarefa 5: Registrar um pagamento e enviar uma cobrança ao cliente;
- Tarefa 6: Iniciar e encerrar uma sessão no *time tracker* vinculada ao projeto;
- Tarefa 7: Convidar um colaborador para o projeto e configurar o *split* financeiro;
- Tarefa 8: Acessar o painel de relatórios e verificar a receita do período.

Durante a execução, foram observados o tempo médio de conclusão de cada tarefa, o número de erros cometidos, a necessidade de auxílio externo e o número aproximado de interações até a conclusão. Ao final, o participante foi convidado a compartilhar comentários, impressões gerais e sugestões de melhoria. O Quadro 4 apresenta os resultados obtidos durante o teste.

Quadro 4 - Resultados de testes com usuários.

Tarefa	**Tempo Médio**	**Erros**	**Conclusão**	**Observações**
1 - Cadastrar cliente e criar projeto	1min 45s	0	Sem auxílio	Fluxo concluído com facilidade
2 - Criar tarefas e subtarefas	1min 10s	0	Sem auxílio	Navegação intuitiva
3 - Criar e enviar *briefing*	2min 20s	1	Sem auxílio	Breve hesitação na seleção do template
4 - Criar proposta	3min 05s	0	Sem auxílio	Elogiou a

e exportar PDF				qualidade visual do PDF gerado
5 - Registrar pagamento e cobrar cliente	1min 30s	0	Sem auxílio	Fluxo claro e direto
6 - Iniciar e encerrar sessão no *time tracker*	0min 45s	0	Sem auxílio	Considerou a funcionalidade prática
7 - Convidar colaborador e configurar *split*	2min 50s	1	Sem auxílio	Pequena dúvida inicial sobre o tipo de *split*
8 - Acessar relatórios e verificar receita	1min 00s	0	Sem auxílio	Gostou da visualização por gráficos

Fonte: O autor (2026).

Os resultados indicam boa aceitação do sistema pelo usuário. Todas as tarefas foram concluídas sem necessidade de auxílio externo, com baixo número de erros e tempo de execução dentro do esperado para um primeiro contato com a plataforma. Os dois erros registrados, um na seleção do template de *briefing* e outro na configuração do tipo de *split*, são pontos já mapeados na avaliação heurística e representam oportunidades de melhoria na documentação e nos textos de apoio dessas funcionalidades.

Entre os comentários coletados ao final do teste, o participante destacou positivamente a qualidade visual da plataforma, a consistência da interface entre os diferentes módulos, a praticidade do *time tracker* integrado ao projeto e a qualidade do PDF gerado pelo editor de propostas. Como sugestão, indicou a inclusão de um tutorial inicial para novos usuários e a adição de *tooltips* explicativos nos campos de configuração do *split* financeiro, pontos que se alinham diretamente aos achados da avaliação heurística.

5.5. Limitações

Apesar do amplo conjunto de funcionalidades implementadas e dos resultados positivos obtidos nos testes, o *FlowDesk* apresenta algumas limitações que representam oportunidades de evolução para versões futuras. A primeira diz respeito à ausência de um mecanismo generalizado de desfazer ações, em fluxos de

edição de conteúdo, como alterações em propostas e tarefas, o sistema não oferece suporte a atalhos como Ctrl+Z ou botão de desfazer, o que pode gerar retrabalho em casos de alterações indesejadas. A segunda refere-se à ausência de documentação e ajuda integrada à plataforma, o que pode dificultar o uso de funcionalidades mais avançadas por usuários com menor familiaridade com ferramentas de gestão.

A terceira limitação diz respeito à ausência de recursos de inteligência artificial, como sugestão automática de tarefas, análise preditiva de prazos ou geração assistida de conteúdo para propostas e *briefings*, representando uma oportunidade de diferenciação relevante para versões futuras. Por fim, a quarta limitação refere-se à base de validação com usuários, os testes conduzidos até o momento envolveram um único usuário real de forma individual, com uma validação ampliada em andamento junto a dez profissionais recrutados. Essas limitações não comprometem o funcionamento ou a proposta central do sistema, mas indicam direções claras para o desenvolvimento contínuo do *FlowDesk* como produto.

6. Considerações finais

Este trabalho teve como objetivo desenvolver o *FlowDesk*, um *SaaS* voltado à *freelancers* da área de design para gerenciamento de projetos, tarefas e clientes. O sistema foi concebido como resposta direta a um problema recorrente e bem documentado no mercado de trabalho autônomo: a fragmentação de ferramentas e a ausência de uma solução centralizada que atendesse simultaneamente às necessidades de organização, comunicação, controle financeiro e colaboração de profissionais independentes.

A partir do referencial teórico construído, foi possível fundamentar as escolhas técnicas e metodológicas do projeto em bases consolidadas da Engenharia de Software, dos métodos ágeis e do *Design* Centrado no Usuário. O mapeamento sistemático realizado no estado da arte confirmou a escassez de soluções especializadas para o contexto de *freelancers* criativos na literatura científica brasileira, reforçando a pertinência e a originalidade da proposta desenvolvida.

O desenvolvimento foi conduzido de forma incremental, com implementação progressiva dos módulos e validação contínua das decisões técnicas ao longo do processo. O resultado é uma plataforma *web* completa, em produção no

domínio app.oflowdesk.com, que centraliza gestão de clientes, projetos e tarefas, controle financeiro com divisão de receita entre colaboradores, sistema de *briefings*, editor de propostas comerciais com geração de PDF, rastreamento de tempo, portal exclusivo para o cliente e relatórios de desempenho. O sistema opera com arquitetura segura, baseada em *Row Level Security* no banco de dados *PostgreSQL*, autenticação JWT, comunicação em tempo real e modelo de assinatura integrado ao *Stripe*.

A avaliação do sistema indicou resultados positivos tanto na perspectiva técnica quanto na perspectiva da experiência do usuário. A avaliação heurística com base nas dez heurísticas de Nielsen não identificou problemas de severidade alta ou crítica, e o teste com usuário real demonstrou que todas as tarefas propostas foram concluídas com sucesso, sem necessidade de auxílio externo e com baixo número de erros. Os pontos de melhoria identificados, ausência de mecanismo de desfazer ações e falta de documentação integrada, representam oportunidades de evolução para versões futuras, sem comprometer o funcionamento central da plataforma.

O *FlowDesk* posiciona-se como uma contribuição relevante tanto no âmbito acadêmico quanto no mercado de ferramentas digitais para a economia criativa. No âmbito acadêmico, preenche uma lacuna identificada no mapeamento sistemático, oferecendo um estudo aplicado sobre o desenvolvimento de plataformas *SaaS* voltadas especificamente ao contexto de *freelancers* criativos no Brasil. No âmbito de mercado, apresenta uma alternativa integrada e acessível às ferramentas genéricas atualmente disponíveis, com diferenciais como o portal do cliente sem necessidade de cadastro, o sistema de *co-working* com *split* financeiro automatizado e o modelo de assinatura escalável com controle inteligente de armazenamento.

Como trabalhos futuros, sugere-se a implementação de um mecanismo generalizado de desfazer ações nos módulos de edição de conteúdo, o desenvolvimento de uma base de conhecimento e documentação integrada à plataforma e a incorporação de recursos de inteligência artificial para assistência na criação de propostas, *briefings* e análise preditiva de prazos e receitas. Sugere-se também a ampliação da base de testes com usuários de diferentes perfis e regiões, a fim de coletar dados mais representativos sobre padrões de uso e necessidades específicas do público-alvo.

Em síntese, o desenvolvimento do *FlowDesk* demonstrou que é possível construir uma solução *SaaS* completa, segura e centrada no usuário para o mercado *freelancer* criativo, utilizando tecnologias modernas de desenvolvimento *web* e boas práticas de Engenharia de *Software*. A plataforma está em produção, validada funcionalmente e com potencial concreto de crescimento e adoção, consolidando-se como uma ferramenta capaz de transformar a forma como *freelancers* criativos organizam, executam e gerenciam seu trabalho.

Referências

ANDERSON, David J. Kanban: *Successful Evolutionary Change for Your Technology Business*. Sequim: Blue Hole Press, 2010.

BECK, Kent et al. *Manifesto for Agile Software Development*. 2001. Disponível em: https://agilemanifesto.org/.

FIRJAN. *Mapeamento da Indústria Criativa 2025: painel de dados*. Disponível em: https://observatorio.firjan.com.br/inteligencia-de-dados/mapeamento-da-industria-criativa-painel-de-dados.

FIRJAN. *Mapeamento da Indústria Criativa 2025: notícia de lançamento*. Disponível em: https://www.firjan.com.br/noticias/mapeamento-da-industria-criativa-2025-8AE4828D96AAF437019783E8CFE02F31-00.htm.

GRAND VIEW RESEARCH. *Freelance Platforms Market Size, Share & Trends Analysis Report, 2024–2030*. São Francisco, 2024. Disponível em: https://www.grandviewresearch.com/industry-analysis/freelance-platforms-market-report.

INTERACTION DESIGN FOUNDATION (IxDF). *User-Centered Design: An Overview.* Aarhus, 2025. Disponível em: https://www.interaction-design.org/literature/topics/user-centered-design.

INTUZ. *Benefits of Next.js: Why Use Next.js For Front-End Development 2025*. Disponível em: https://www.intuz.com/blog/5-reasons-why-you-should-use-next.js-for-your-front-end-development.

ISO. ISO 9241-210: *Ergonomics of human-system interaction - Part 210: Human-centred design for interactive systems*. Geneva: International Organization for Standardization, 2019.

KITCHENHAM, Barbara; CHARTERS, Stuart. *Guidelines for Performing Systematic Literature Reviews in Software Engineering.* Technical Report EBSE 2007-001, Keele University, 2007.

NORMAN, Donald A. *The Design of Everyday Things*. Revised and Expanded Edition. New York: Basic Books, 2013.

NIELSEN, Jakob. *Usability Engineering*. Boston: Academic Press, 1994.

PAGEPRO. *What's New in Next.js 15: Faster Development with Turbopack.* 2024. Disponível em: https://pagepro.co/blog/next-js-15-new-features/.

PRESSMAN, Roger S. *Engenharia de Software: uma abordagem profissional*. 8. ed. Porto Alegre: AMGH, 2016.

PROJECT MANAGEMENT INSTITUTE (PMI). *Guia PMBOK – Um Guia do Conhecimento em Gerenciamento de Projetos*. 7. ed. Newtown Square: PMI, 2021.

SCHWABER, Ken; SUTHERLAND, Jeff. *The Scrum Guide: The Definitive Guide to Scrum: The Rules of the Game*. Scrum.org, 2020. Disponível em: https://scrumguides.org. Acesso em: maio 2026.

SOMMERVILLE, Ian. *Engenharia de Software*. 9. ed. São Paulo: Pearson, 2011.

SUPABASE. *Documentation – Introduction to Supabase*. 2024. Disponível em: https://supabase.com/docs.

TAILWIND LABS. *Tailwind CSS Documentation*. 2024. Disponível em: https://tailwindcss.com/docs.

TILKOV, Stefan; VINOSKI, Steve. *Node.js: Using JavaScript to Build High-Performance Network Programs*. *IEEE Internet Computing*, v. 14, n. 6, p. 80–83, 2010.

UPWORK. *Upwork Study Finds 1 in 4 U.S. Skilled Knowledge Workers Now Work Independently, Generating US $1.5 Trillion in Earnings*. 2025. Disponível em: https://investors.upwork.com/news-releases/news-release-details/upwork-study-finds-1-4-us-skilled-knowledge-workers-now-work.

Apêndice

Apêndice A - Mapeamento Sistemático: Questões de Pesquisa, *String* de Busca e Critérios de Inclusão e Exclusão

Este apêndice detalha os procedimentos metodológicos adotados no mapeamento sistemático descrito na Seção 3, incluindo as questões de pesquisa definidas, a *string* de busca utilizada e os critérios de inclusão e exclusão aplicados na seleção dos artigos.

1. Questões de Pesquisa (QPs)

No Quadro A.1 é mostrado as questões de pesquisa (QP) utilizadas:

Quadro A.1 Questões de Pesquisa.

QP1	Quais soluções tecnológicas voltadas à gestão de projetos para *freelancers* criativos foram identificadas nos últimos anos?
QP2	Quais dessas soluções abordam colaboração direta e *feedback* entre cliente e profissional dentro da plataforma?
QP3	Quais tecnologias e arquiteturas são mais utilizadas na implementação dessas plataformas?
QP4	Quais lacunas ainda existem no mercado e na literatura em relação a sistemas voltados para *freelancers* criativos?

Fonte: O autor (2026).

2. *String* de busca

Para mapear os trabalhos acadêmicos, foi elaborada uma *string* de busca com palavras-chave relacionadas ao tema. A pesquisa foi realizada na plataforma *Google Scholar*. O Quadro A.2 apresenta a *string* utilizada.

Quadro A.2 *String* de busca.

String	Expressão
1	*("freelancer" OR "creative professional" OR "designer") AND ("project*

	management" OR "workflow platform" OR "collaboration tool") AND ("SaaS" OR "web platform")

Fonte: O autor (2026).

3. Critérios de Inclusão e Exclusão

Para a seleção dos artigos, foram definidos os seguintes Critérios de Inclusão (CI):

- CI1: Artigos e trabalhos completos publicados entre 2019 e 2025.
- CI2: Artigos escritos em português.
- CI3: Estudos que abordem plataformas *SaaS*, sistemas *web* ou aplicativos para gestão de projetos *freelancers*.
- CI4: Trabalhos que envolvam colaboração, experiência do usuário ou automação de processos digitais.

Após a aplicação dos critérios de inclusão (CI) feita, os artigos selecionados foram analisados com os critérios de exclusão (CE), resultando na eliminação daqueles que não atenderam a esses critérios:

- CE1: Artigos duplicados.
- CE2: Artigos que não apresentem contexto de *freelancers* ou gestão de projetos.
- CE3: Estudos puramente teóricos, sem aplicação prática.
- CE4: Artigos que não puderam ser acessados.

Apêndice B - Funcionalidades Complementares do Sistema

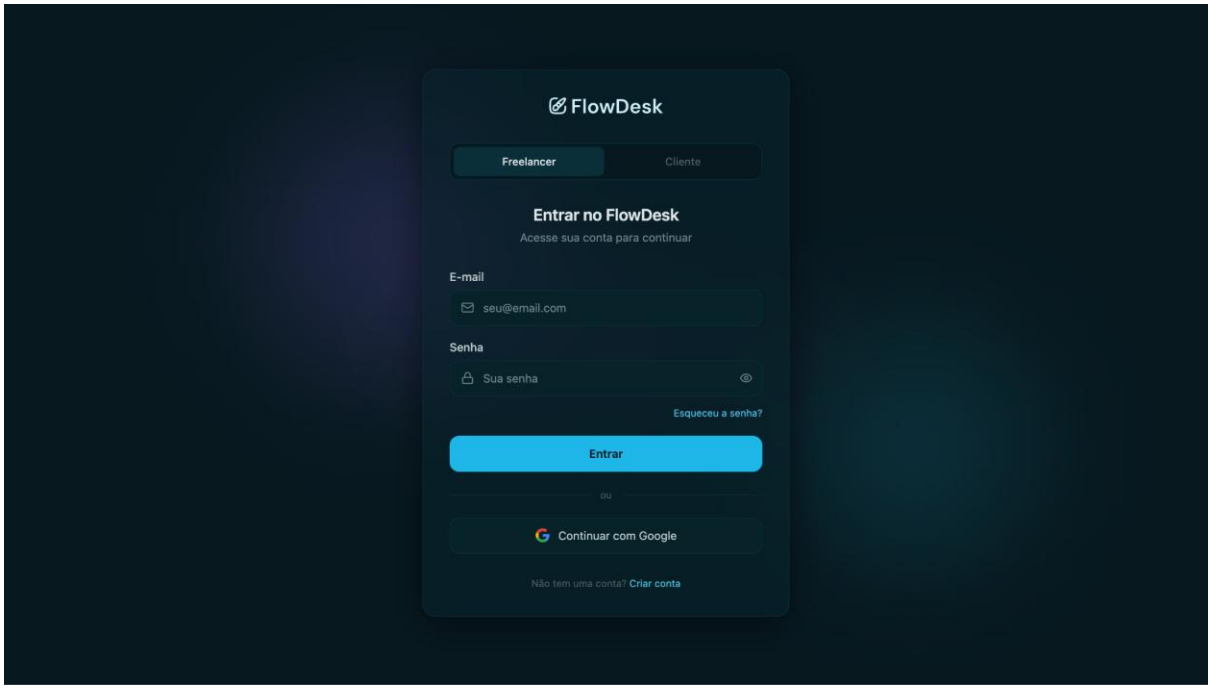
Este apêndice descreve as funcionalidades complementares do *FlowDesk*, que compõem o sistema de forma integrada às funcionalidades principais descritas na Seção 5.2.

1. Autenticação e *Onboarding*

O *FlowDesk* possui um fluxo completo de autenticação gerenciado pelo *Supabase Auth* com *tokens* JWT. O sistema oferece cadastro com

validação de e-mail em tempo real, *login* com e-mail e senha, recuperação de senha via e-mail e redefinição por página dedicada com *token* de recuperação. Após o primeiro acesso, o usuário é direcionado a um fluxo de *onboarding* para configuração inicial do perfil, incluindo nome e foto de avatar, comprimida automaticamente para o formato *WebP* com dimensões de 256×256 *pixels* e tamanho máximo de 48KB, conforme apresentado na Figura B.1. Todas as rotas internas da aplicação verificam a existência de sessão ativa antes de renderizar o conteúdo, redirecionando automaticamente usuários não autenticados para a tela de *login*.

Figura B.1 - Tela de *login.*



Fonte: O autor (2026).

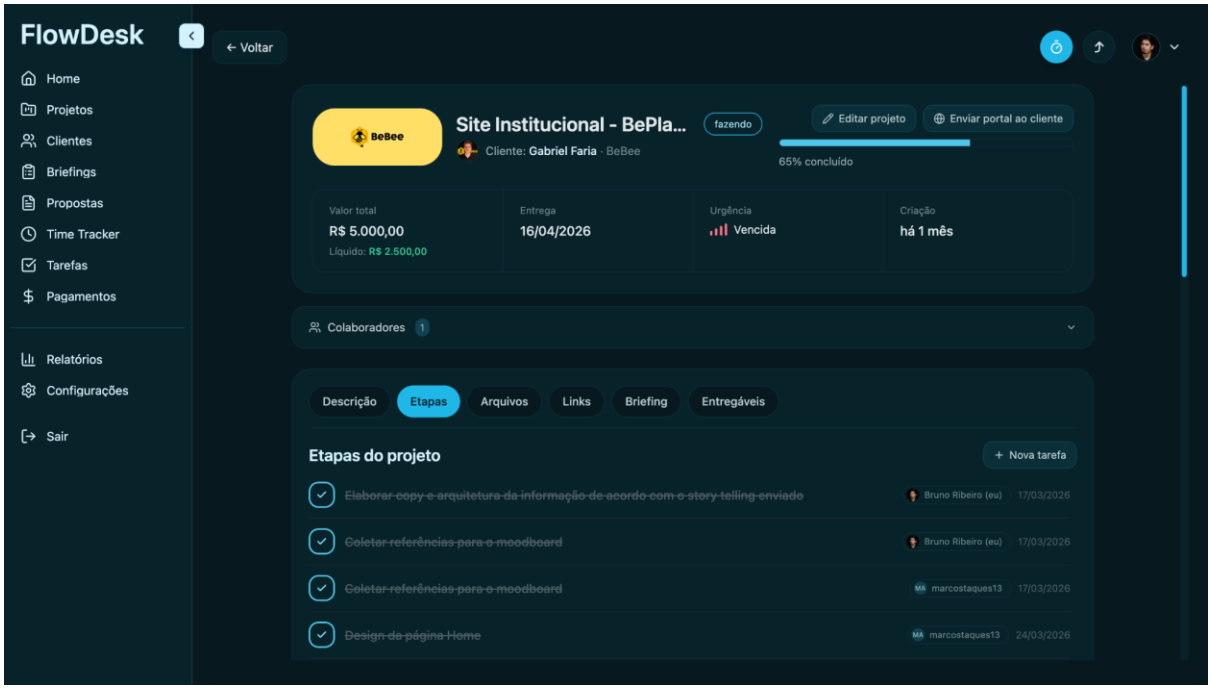
2. Gestão de Clientes e Projetos

O módulo de clientes oferece *CRUD* completo para gestão da carteira de contatos do *freelancer*, com cadastro de nome, e-mail, telefone, empresa e foto. O *upload* da foto do cliente passa por um *pipeline* de otimização automática, sendo convertida para *WebP* e comprimida para no máximo 48KB antes do armazenamento no *Supabase Storage*.

O módulo de projetos é o núcleo do sistema. Cada projeto reúne todas as informações e sub-módulos relacionados a um trabalho específico:

tarefas, arquivos, *links*, *briefings*, propostas, pagamentos e colaboradores. Na criação, o *freelancer* define título, descrição, cliente vinculado, valor, prazo de entrega, data de início, forma de pagamento, colaboradores, tarefas a serem realizadas e *links* do projeto. É possível também fazer *upload* de uma capa para o projeto, redimensionada automaticamente para 640×360 *pixels* no formato *WebP*, conforme ilustrado na Figura B.2.

Figura B.2 - Tela de detalhes do projeto.



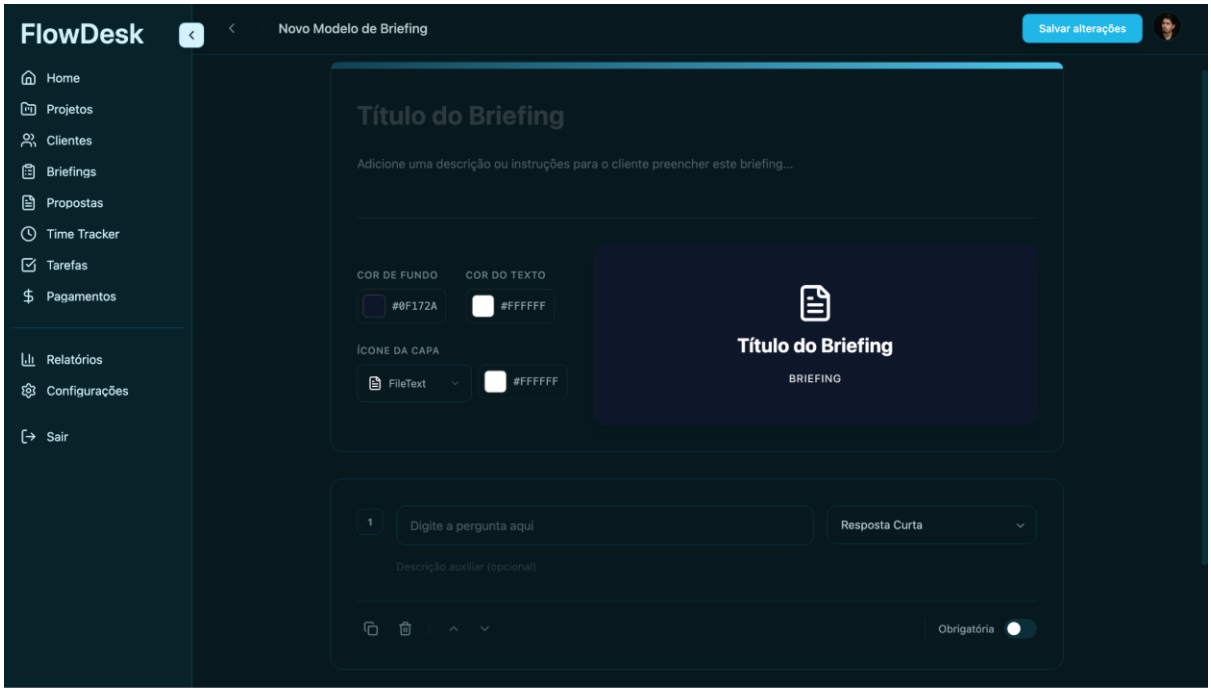
Fonte: O autor (2026).

Os projetos são organizados em um *kanban* visual com quatro colunas de *status*: Para Fazer, Em Andamento, Pagamento Pendente e Finalizado. O sistema implementa um mecanismo de bloqueio de conclusão que verifica, antes de permitir a finalização do projeto, se há tarefas incompletas, *briefings* sem resposta, entregáveis pendentes de aprovação e pagamentos em aberto, garantindo que nenhum ciclo seja encerrado com pendências.

3. *Briefings*

O módulo de *briefings* permite ao *freelancer* criar *templates* reutilizáveis de coleta de informações e enviá-los ao cliente antes do início do projeto. Os *templates* suportam os seguintes tipos de campo: texto curto, texto longo, seleção única e caixa de seleção, permitindo a construção de formulários adaptados a diferentes tipos de projeto, a Figura B.3 apresenta a tela de *briefings*.

Figura B.3 - Tela de *Briefings.*



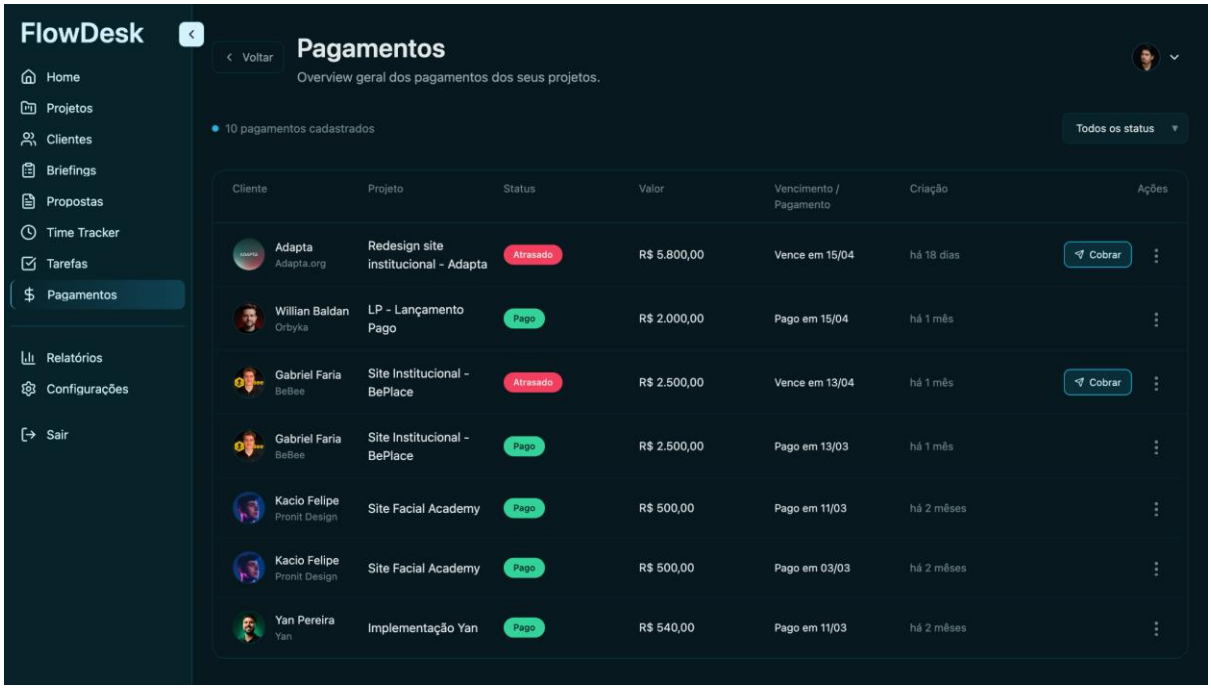
Fonte: O autor (2026).

Após o envio, o cliente recebe um *link* público para responder o *briefing* sem necessidade de criar uma conta na plataforma. O *freelancer* acompanha o *status* de cada envio, pendente, respondido ou expirado, e visualiza as respostas organizadas campo a campo em uma interface dedicada. O sistema registra a data de resposta e permite definir prazo de validade para cada envio.

4. Controle Financeiro

O módulo financeiro permite o registro e acompanhamento completo dos pagamentos de cada projeto. O *freelancer* cadastra pagamentos com valor, forma de pagamento, PIX, PIX em duas vezes ou cartão, e *status*. Quando há colaboradores no projeto, pode ser definido se o colaborador receberá um valor fixo ou porcentagem derivado do pagamento a receber. Após a configuração e aceite do colaborador, os cálculos são feitos de forma automática e organizados no painel financeiro de cada um, permitindo ao *freelancer* visualizar o valor bruto e o valor líquido após o repasse, conforme apresentado na Figura B.4.

Figura B.4 - Tela de pagamentos.



Fonte: O autor (2026).

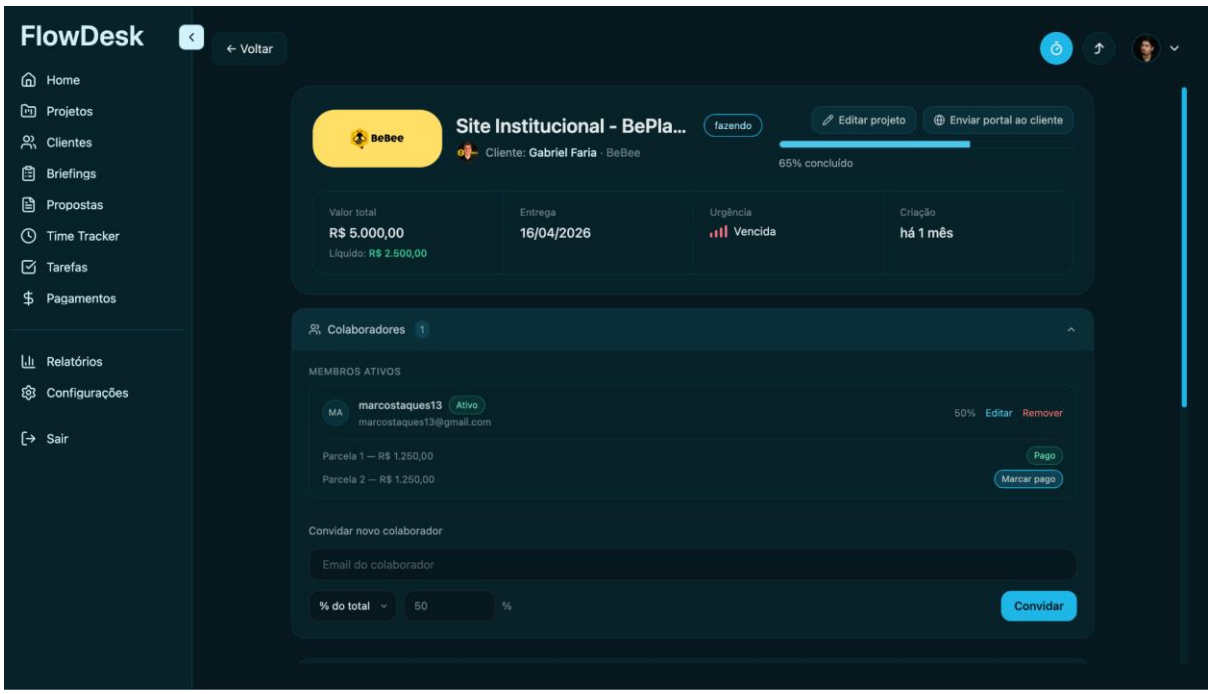
O sistema permite o envio de cobrança ao cliente diretamente por e-mail, incluindo a chave PIX cadastrada e os detalhes do pagamento. Um registro de data do último envio de cobrança evita notificações duplicadas. A visão consolidada de pagamentos reúne todos os pagamentos de todos os projetos com filtros por *status* e período, incluindo um *card* de pagamentos a

receber na *dashboard* principal que considera projetos de qualquer *status* com valores pendentes.

5. *Co-working* e *Split* Financeiro

O sistema de *co-working* permite ao *freelancer* convidar outros profissionais para colaborar em seus projetos, com controle granular sobre permissões e divisão financeira. O *owner* do projeto envia um convite por e-mail, gerado com *token* criptográfico e validade de sete dias. O colaborador recebe o e-mail com o *link* de aceitação, faz *login* ou cria uma conta e passa a integrar o projeto com o papel de colaborador, na Figura B.5 é apresentada a tela de configuração do *co-working*.

Figura B.5 - Tela de configuração do *co-working.*



Fonte: O autor (2026).

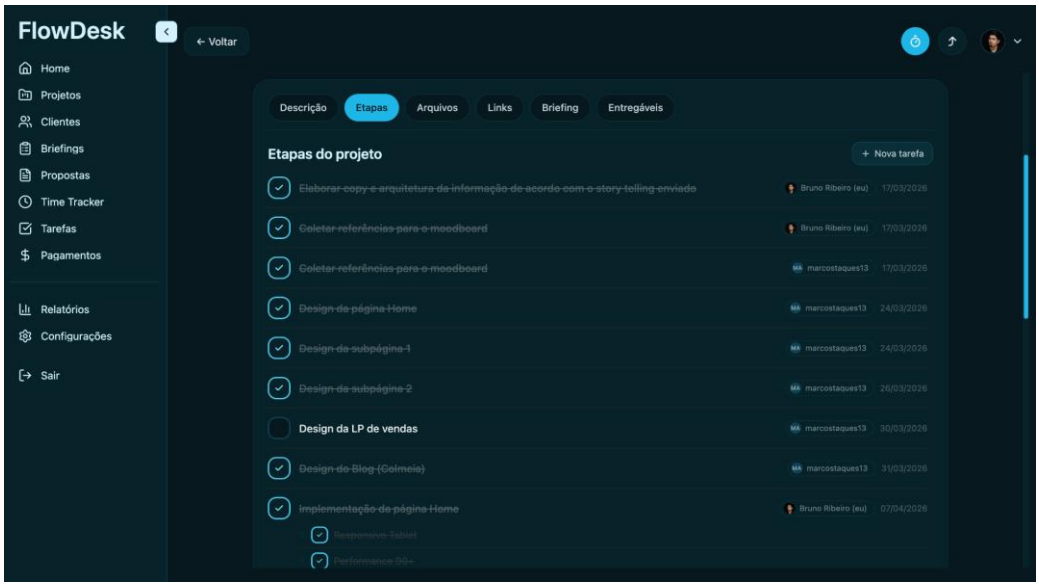
A divisão financeira pode ser configurada de duas formas: percentual, em que o colaborador recebe uma porcentagem de cada pagamento do projeto, ou valor fixo, independente dos pagamentos recebidos. O sistema calcula e registra automaticamente o valor devido ao colaborador a cada novo pagamento cadastrado. O colaborador pode solicitar pagamento ao

owner diretamente pela plataforma, e o *owner* registra o repasse quando realizado. É importante destacar que o pagamento do cliente e o repasse ao colaborador são processos independentes, um projeto com todos os pagamentos recebidos não implica necessariamente que o colaborador já foi pago.

6. Tarefas e Subtarefas

O sistema de tarefas permite ao *freelancer* organizar o trabalho interno de cada projeto em unidades menores e rastreáveis. Cada tarefa possui título, descrição, nível de prioridade, responsável e prazo de entrega. Os níveis de prioridade disponíveis são Nenhuma, Baixa, Média, Alta e Urgente, permitindo acompanhamento visual do progresso. Cada tarefa pode conter subtarefas vinculadas, com marcação individual de conclusão, como ilustrado na Figura B.6.

Figura B.6 - Tela de tarefas e subtarefas do projeto.



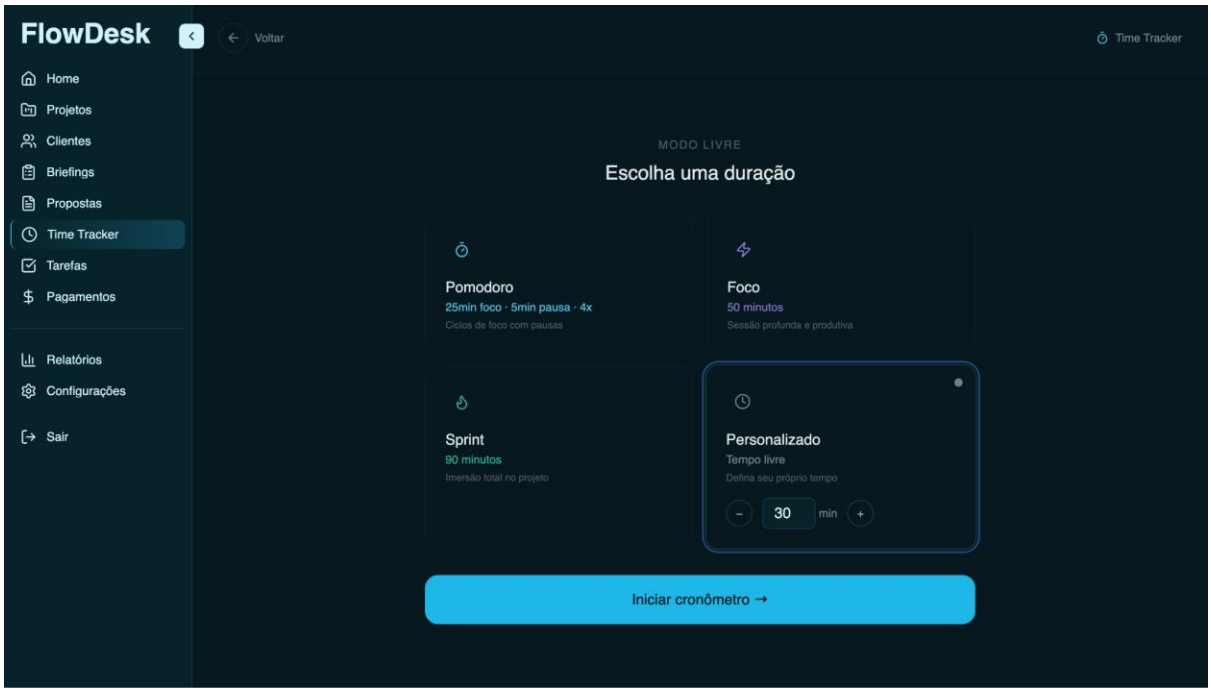
Fonte: O autor (2026).

Além da visualização dentro do projeto, o sistema oferece uma página global de tarefas em que o *freelancer* pode visualizar e filtrar todas as tarefas de todos os projetos em um único lugar, facilitando a priorização do trabalho diário.

7. Time Tracker

O rastreamento de tempo permite ao *freelancer* registrar com precisão as horas dedicadas a cada projeto e tarefa. O sistema oferece três pontos de acesso ao cronômetro: uma página global que pode ser acessada pela *sidebar*, um cronômetro integrado na página do projeto e um cronômetro vinculado diretamente à tarefa. Cada sessão registra os horários de início e fim e a duração em segundos, conforme demonstrado na Figura B.7.

Figura B.7 - Tela inicial do *Time Tracker.*



Fonte: O autor (2026).

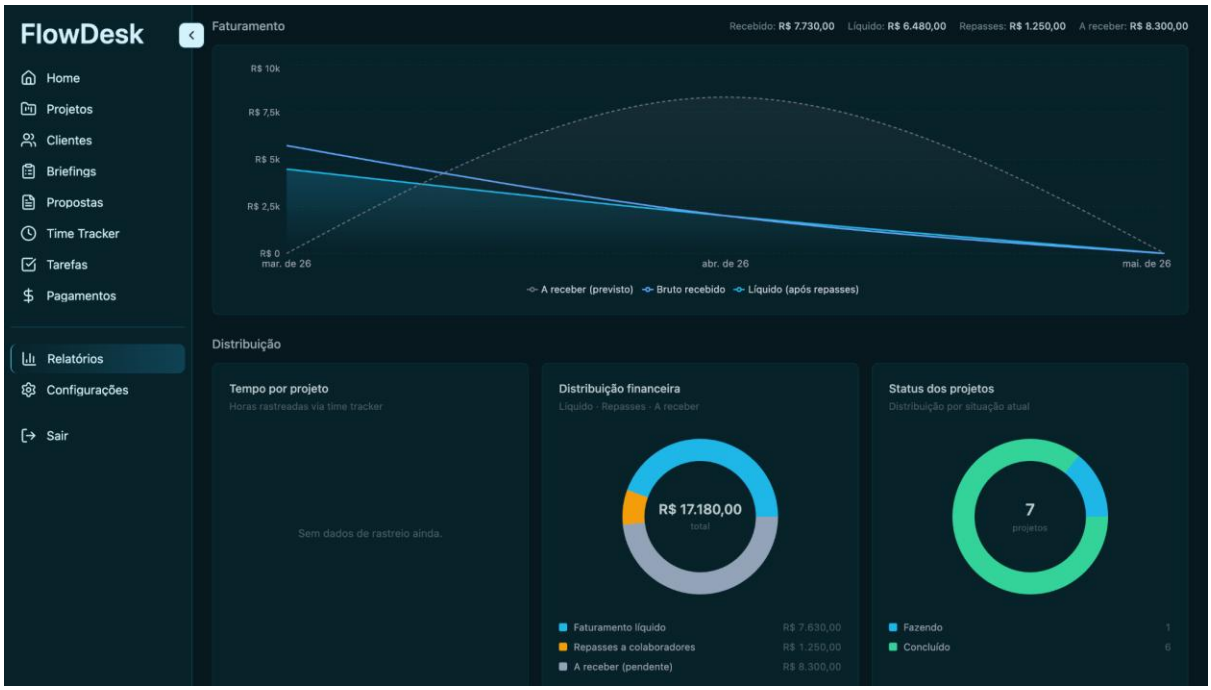
O histórico de sessões fica disponível por projeto, e o tempo total trabalhado é utilizado nos relatórios de produtividade e desempenho, permitindo ao *freelancer* analisar quanto tempo cada tipo de projeto ou cliente demanda na prática.

8. Relatórios

O módulo de relatórios oferece cinco painéis analíticos para acompanhamento do desempenho do negócio. O relatório de *Overview* apresenta receita total, projetos concluídos, horas trabalhadas e *ticket* médio, incluindo ganhos obtidos como colaborador em projetos de terceiros. O relatório de Performance compara metas definidas com resultados realizados por período.

O relatório de Produtividade analisa tarefas concluídas e tempo médio por tarefa. O relatório de Comparação de Projetos permite análise lado a lado de múltiplos projetos. Por fim, o relatório de Receita apresenta o *breakdown* financeiro por projeto, cliente e período, considerando também ganhos como colaborador. A Figura B.8 ilustra um dos painéis disponíveis no módulo.

Figura B.8 - Tela de relatórios.



Fonte: O autor (2026).

Todos os relatórios possuem filtros por período e são renderizados com gráficos interativos pela biblioteca *Recharts*, permitindo visualização clara de tendências e padrões ao longo do tempo.

9. Sistema de Planos e Armazenamento

O *FlowDesk* opera com um modelo de assinatura *SaaS* estruturado em três planos: *Trial*, Essencial e Profissional. O período de *trial* tem duração de sete dias e oferece acesso completo à plataforma sem restrições, permitindo que o *freelancer* conheça todas as funcionalidades antes de escolher um plano pago. Após o encerramento do *trial*, o usuário é direcionado para a seleção de um dos planos disponíveis. O Quadro B.1 apresenta os limites e recursos de cada plano.

Quadro B.1 - Comparativo de Planos.

Essencial	**Profissional**
Até 10 projetos	Projetos ilimitados
Até 10 clientes	Clientes ilimitados
1 colaborador por projeto	Até 5 colaboradores por projeto
Cliente pode apenas responder briefing e acompanhar status do projeto	Portal completo do cliente. Pode responder briefings, aprovar ou desaprovar demandas, enviar feedbacks, confirmar pagamentos.
5GB de armazenamento	20GB de armazenamento

Fonte: O autor (2026).

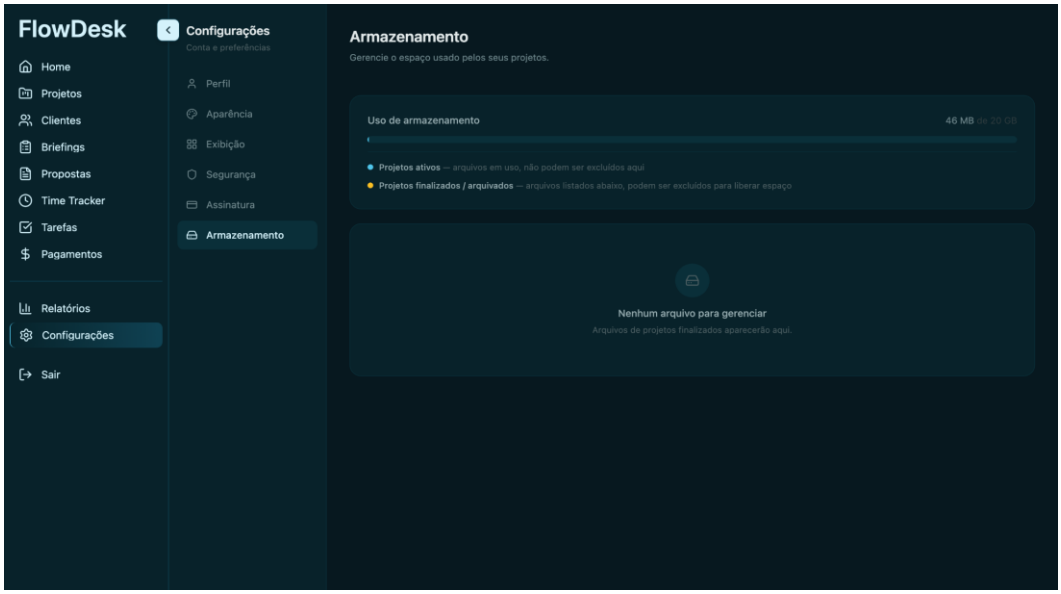
A gestão das assinaturas é realizada pela integração com o *Stripe*, plataforma de pagamentos responsável pelo processamento de cobranças recorrentes, troca de plano, cancelamento e histórico de faturas. O *freelancer* acessa o painel de gerenciamento da assinatura diretamente pelo *FlowDesk*, que abre o *Stripe Customer Portal* sem necessidade de sair da plataforma. O sistema também suporta a contratação de *add-ons* de armazenamento extra em incrementos de 10GB, para usuários que necessitam de capacidade adicional sem precisar migrar de plano.

O sistema verifica os limites do plano ativo antes de cada ação relevante. Ao tentar criar um novo projeto ou cliente além do limite permitido, convidar um colaborador acima do número máximo por projeto ou fazer *upload* de arquivos que ultrapassem o espaço disponível, o sistema bloqueia a ação e exibe um *modal* de

upgrade centralizado com informações específicas sobre o limite atingido. Esse modal é disparado por um evento customizado propagado por qualquer parte da aplicação, e o componente responsável por exibi-lo está montado globalmente, garantindo consistência na experiência independentemente de qual módulo o usuário esteja utilizando no momento.

O uso de armazenamento é exibido na *dashboard* principal por meio de um *card* com barra de progresso colorida, que varia do verde ao amarelo e ao vermelho conforme o consumo se aproxima do limite do plano. A página de configurações oferece uma aba dedicada ao gerenciamento detalhado do armazenamento, onde o *freelancer* visualiza os arquivos que podem ser excluídos manualmente, restritos a projetos com *status* finalizado, concluído ou arquivado, agrupados por projeto com indicação de dias restantes para exclusão automática. A Figura B.9 ilustra essa interface de gerenciamento.

Figura B.9 - Tela de gerenciamento do armazenamento



Fonte: O autor (2026).

O sistema implementa um mecanismo de limpeza automática executado diariamente por um *Cron Job* configurado via *Vercel*, acionado às três horas da manhã. O *job* identifica projetos com *status* não ativo cujo campo *completed_at* registra data superior a trinta dias, remove os arquivos correspondentes do *bucket* no *Supabase Storage*, exclui as capas dos projetos do *bucket project-covers* e limpa os registros da tabela arquivos_projeto no banco de dados. A rota responsável pela

execução é protegida por um segredo de autenticação injetado automaticamente pela *Vercel*, impedindo acionamentos externos não autorizados.

Todo *upload* de imagem no sistema passa por um *pipeline* de otimização antes do armazenamento. O processo envolve validação de formato e tamanho, conversão para *WebP* via *Canvas API* e compressão por busca binária para atingir o tamanho máximo permitido sem perda desnecessária de qualidade. As especificações variam conforme o contexto: avatares e fotos de perfil são processados em 256×256 *pixels* com limite de 48KB, capas de projeto em 640×360 *pixels* com limite de 120KB, e *banners* em 1200×600 *pixels* com limite de 400KB. Esse *pipeline* garante que o armazenamento seja utilizado de forma eficiente, reduzindo o consumo do espaço disponível no plano do usuário.

Apêndice C - Diagramas do Sistema

Este apêndice apresenta os diagramas desenvolvidos para o *FlowDesk*, oferecendo diferentes perspectivas sobre a arquitetura, os fluxos de interação e a estrutura de dados do sistema.

1. Diagrama de Casos de Uso

O diagrama de casos de uso da Figura C.1 apresenta todas as funcionalidades do sistema e os atores envolvidos, evidenciando a separação entre as ações disponíveis ao *freelancer* e ao cliente dentro da plataforma.

Figura C.1 - Diagrama de Casos de Uso.



Fonte: O autor (2026).

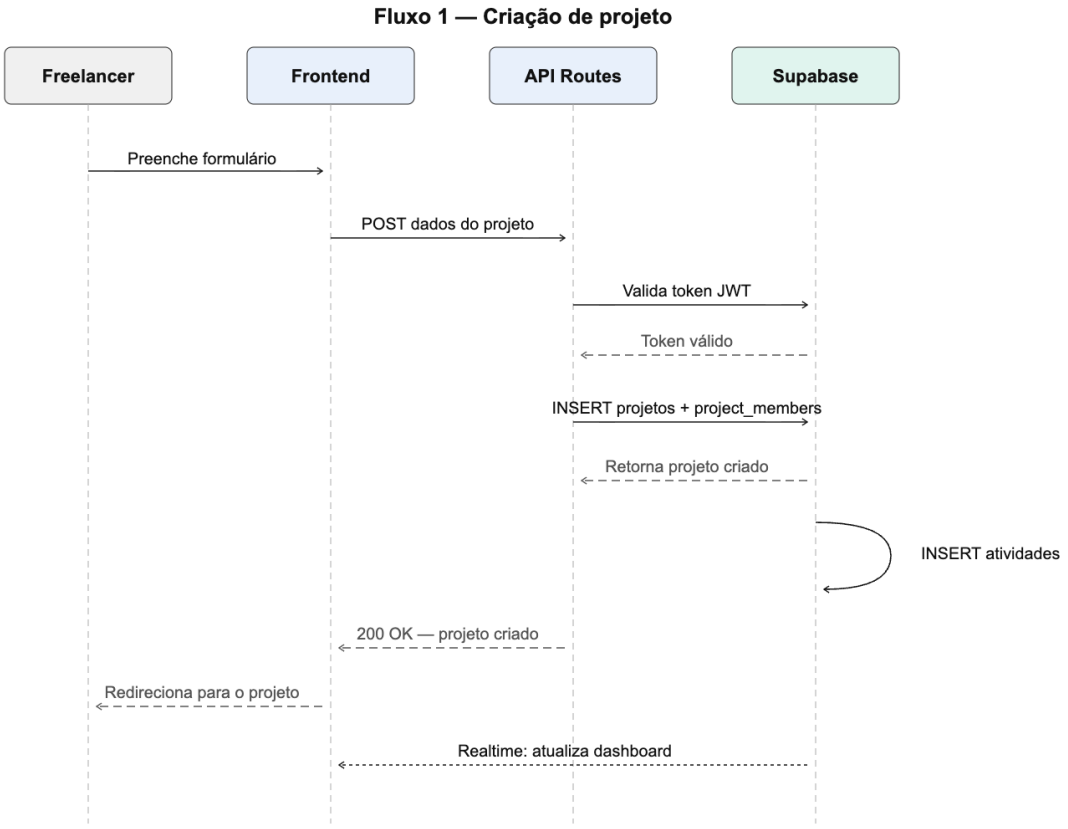
2. Diagramas de Sequência

Os diagramas de sequência ilustram em nível técnico os principais fluxos de interação entre os componentes do sistema.

No fluxo de criação de projeto, ilustrado na Figura C.2, o *freelancer* preenche o formulário de criação no *front-end*, que envia os dados via requisição para as rotas de API *server-side*. A API valida o *token* JWT junto ao *Supabase* e, confirmada a autenticidade, realiza a inserção do projeto na

tabela projetos e o registro do *owner* na tabela *project_members*. Após a conclusão, o *Supabase Realtime* notifica o *front-end* automaticamente, atualizando a *dashboard* sem necessidade de recarregamento da página.

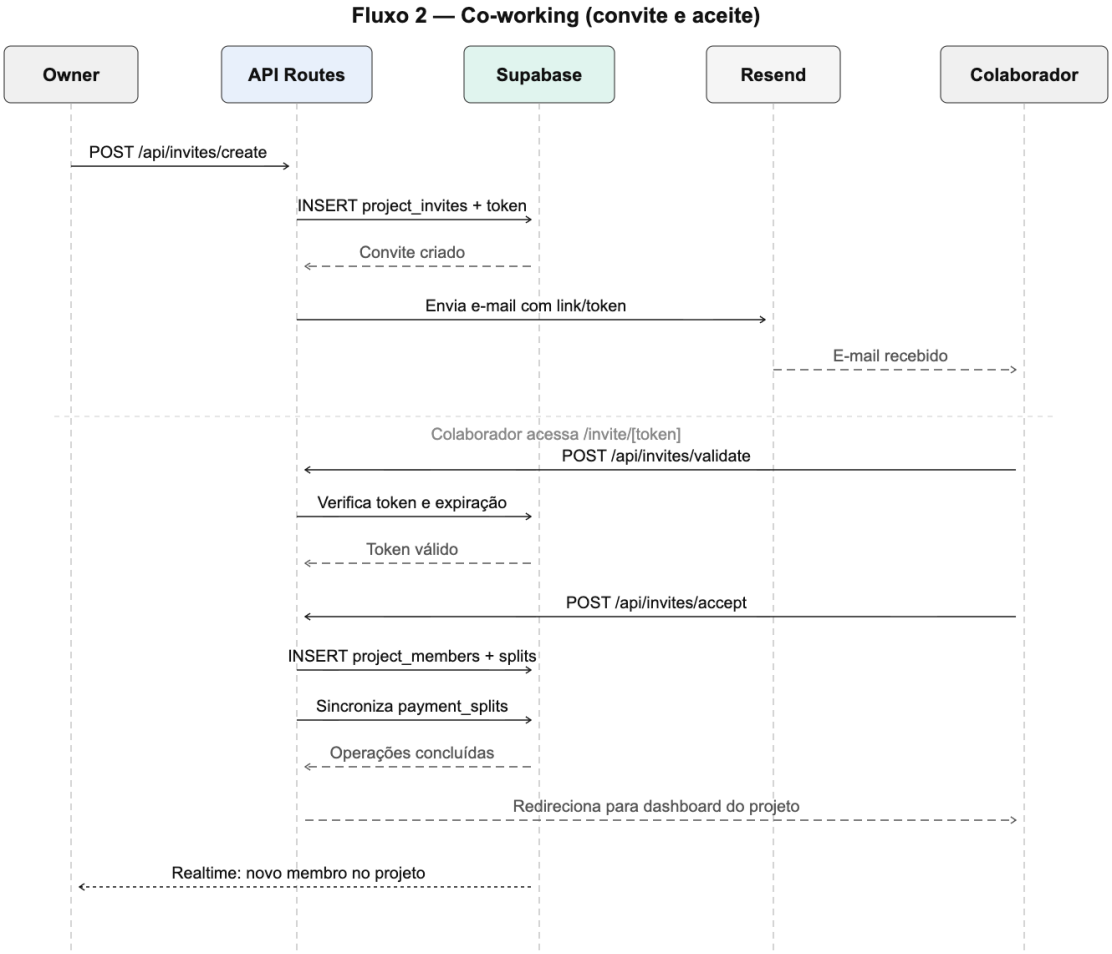
Figura C.2 - Diagrama de sequência: criação de projeto.



Fonte: O autor (2026).

No fluxo de *co-working*, ilustrado na Figura C.3, o *owner* cria o convite pela plataforma, que aciona a rota POST */api/invites/create*. A API gera um *token* criptográfico, insere o convite na tabela *project_invites* com validade de sete dias e aciona o *Resend* para envio do e-mail ao colaborador. O colaborador acessa o *link* recebido, e o sistema valida o *token* e sua expiração via POST */api/invites/validate*. Ao confirmar o aceite, a rota POST */api/invites/accept* cria os registros em *project_members* e *project_member_splits* e sincroniza os *splits* financeiros na tabela *collaborator_payment_splits*. O *Supabase Realtime* notifica o *owner* em tempo real sobre a entrada do novo colaborador no projeto.

Figura C.3 - Diagrama de sequência: fluxo de *co-working.*

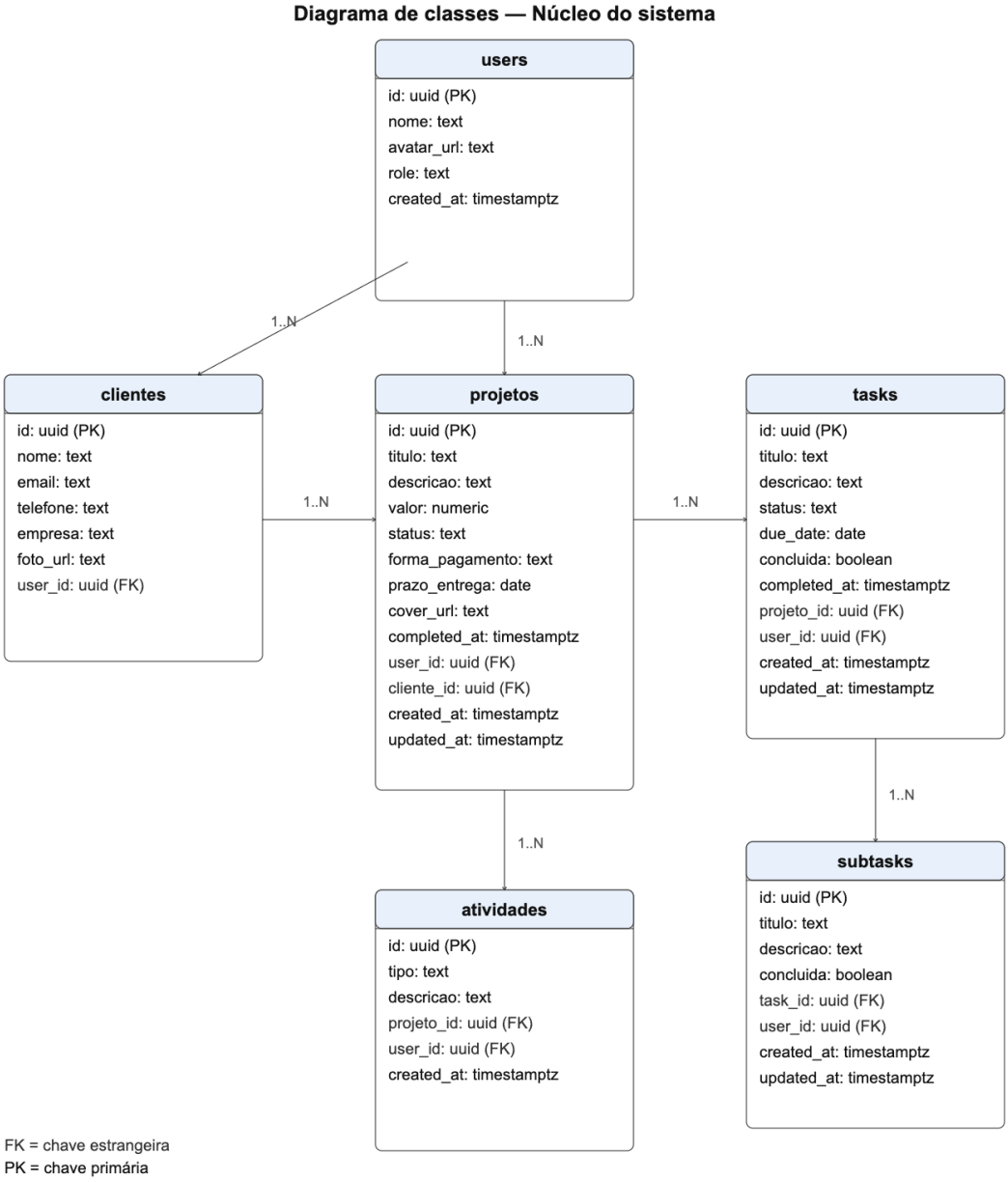


Fonte: O autor (2026).

3. Diagramas de Classes

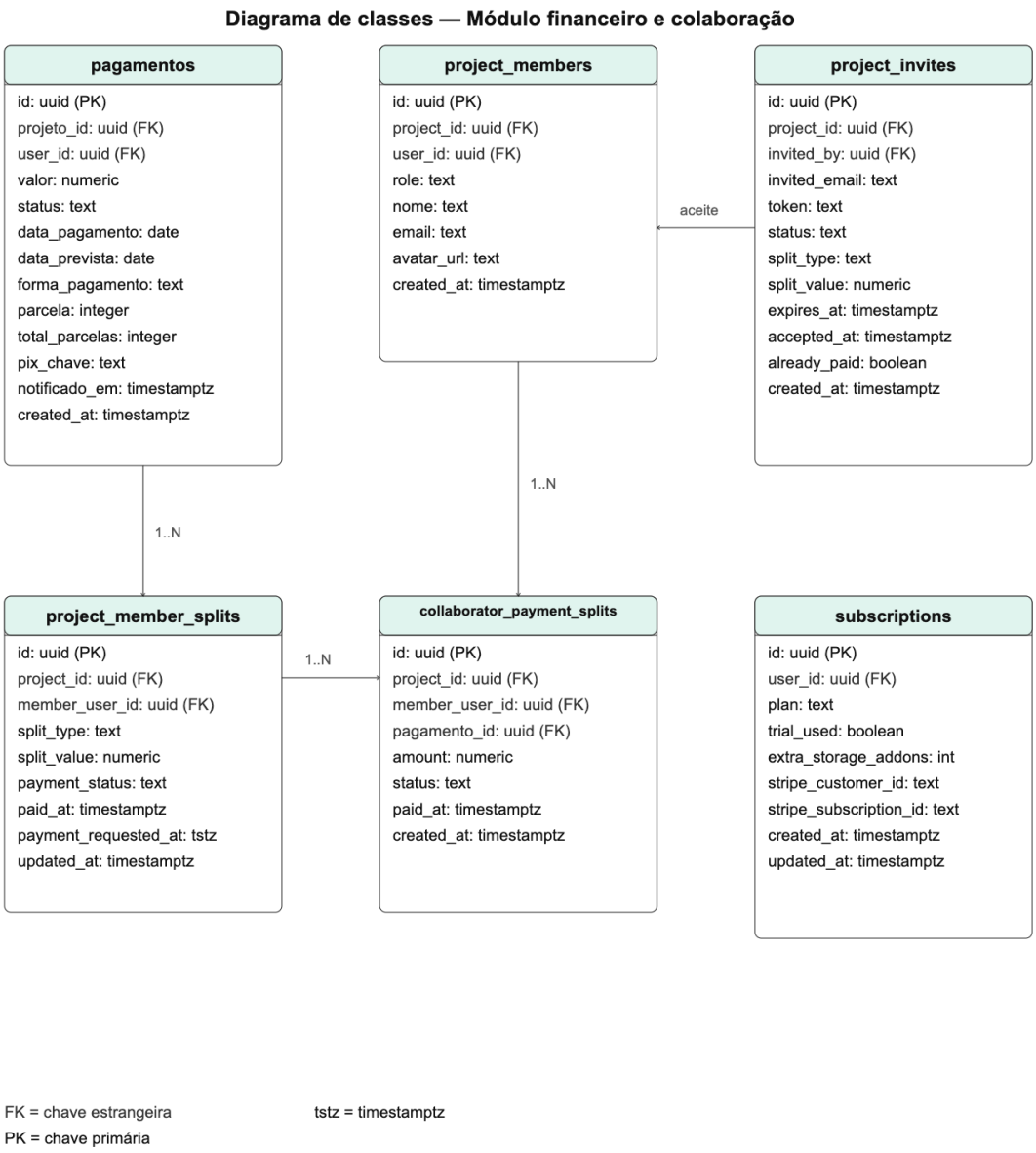
Os diagramas de classes apresentam a estrutura interna do banco de dados do *FlowDesk*, organizados em dois grupos. O primeiro diagrama, ilustrado na Figura C.4, apresenta o núcleo do sistema, composto pelas entidades de usuários, clientes, projetos, tarefas, subtarefas e atividades, com seus respectivos atributos e relacionamentos. O segundo diagrama, ilustrado na Figura C.5, apresenta o módulo financeiro e de colaboração, abrangendo as entidades de pagamentos, membros do projeto, convites, *splits* financeiros e assinaturas.

Figura C.4 - Diagrama de classes: núcleo do sistema.



Fonte: O autor (2026).

Figura C.5 - Diagrama de classes: módulo financeiro e colaboração.



Fonte: O autor (2026).