INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA PARAÍBA
CAMPUS CAJAZEIRAS
CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

TASKER: FERRAMENTA PARA GESTÃO DE EQUIPES E PROJETOS

MIKAEL STANLEY TRAJANO LIMA

Cajazeiras
2026

MIKAEL STANLEY TRAJANO LIMA

TASKER: FERRAMENTA PARA GESTÃO DE EQUIPES E PROJETOS

Trabalho de Conclusão de Curso apresentado junto ao Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas do Instituto Federal de Educação, Ciência e Tecnologia da Paraíba - Campus Cajazeiras, como requisito à obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas.

Orientador

Prof. Me. Francisco Paulo de Freitas Neto.

Cajazeiras
2026

IFSertãoPB / Campus Cajazeiras
Coordenação de Biblioteca
Biblioteca Prof. Ribamar da Silva
Catalogação na fonte: Cícero Luciano Félix CRB-15/750

L732t Lima, Mikael Stanley Trajano.
Tasker : ferramenta para gestão de equipes e projetos / Mikael Stanley Trajano Lima.– 2026.

64f. : il.

Trabalho de Conclusão de Curso (Tecnólogo em Análise e Desenvolvimento de Sistemas) - Instituto Federal de Educação, Ciência e Tecnologia do Sertão Paraibano, Cajazeiras, 2026.

Orientador(a): Prof. Me. Francisco Paulo de Freitas Neto.

1. Sistema de gestão. 2. Gestão de projetos. 3. Gestão de desempenho. 4. Sistema de informação. I. Instituto Federal de Educação, Ciência e Tecnologia da Paraíba. II. Título.

IFSertão PB/CZ

CDU: 004.4 (043.2)

MINISTÉRIO DA EDUCAÇÃO
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA PARAÍBA

MIKAEL STANLEY TRAJANO LIMA

TASKER:FERRAMENTA PARA GESTÃO DE EQUIPES E PROJETOS

Trabalho de Conclusão de Curso apresentado junto ao Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas do Instituto Federal de Educação, Ciência e Tecnologia da Paraíba - Campus Cajazeiras, como requisito à obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas.

Orientador

Prof. Me. Francisco Paulo de Freitas Neto

Aprovada em: **13 de agosto de 2026.**

Prof. Me. Francisco Paulo de Freitas Neto - Orientador

Prof. Dra. Eva Maria Campos Pereira - Avaliador

IFPB - Campus Cajazeiras

Prof. Me. Fábio Abrantes Diniz - Avaliador

IFPB - Campus Cajazeiras

Documento assinado eletronicamente por:

- **Fabio Abrantes Diniz**, **PROFESSOR ENS BASICO TECN TECNOLOGICO**, em 14/08/2026 09:26:48.
- **Eva Maria Campos Pereira**, **PROFESSOR ENS BASICO TECN TECNOLOGICO**, em 14/08/2026 09:27:30.
- **Francisco Paulo de Freitas Neto**, **PROFESSOR ENS BASICO TECN TECNOLOGICO**, em 14/08/2026 09:55:27.

Este documento foi emitido pelo SUAP em 14/08/2026. Para comprovar sua autenticidade, faça a leitura do QRCode ao lado ou acesse https://suap.ifpb.edu.br/autenticar-documento/ e forneça os dados abaixo:

Código 923154
Verificador: e52664e621
Código de Autenticação:

Dedico este trabalho a toda minha família, aos presentes e aos que estão em descanso eterno, aqueles que estiveram sempre ao meu lado, auxiliando em todos os momentos da minha vida.

AGRADECIMENTOS

Agradeço primeiramente a Deus por conceder a oportunidade de cursar uma faculdade e abençoar todo este período presente nela. E por esse dom, capaz de ajudar a melhorar vida de muitas pessoas.

A nossa Mãe, Virgem Maria, por sempre interceder em todos os momentos e me guiar nos caminhos do Senhor.

Agradeço a minha família por sempre estar ao meu lado, auxiliando e guiando durante todo este período. A meus pais que, pelos conselhos e puxões de orelhas, me ensinaram a continuar em frente, mesmo com as dificuldades. E me fizeram entender que existe um motivo maior para lutar.

Aos professores, em especial ao meu orientador Francisco Paulo de Freitas Neto, que dedicaram parte de suas vidas a transmitirem seus conhecimentos, contribuindo efetivamente ao meu desenvolvimento intelectual e profissional.

A meus amigos, Fernando Santos e Jeyce Luiza, que através dos treinos e as partidas de sinucas, dos momentos descontraídos e as conversas aleatórias. Me fizeram buscar cada vez mais melhorar como pessoa.

Aos meus avós, em especial a Maria Ana de Jesus, vovó Caiquinha, que me fizeram entender que a vida é para ser vivida com leveza e levar essa leveza para vida de outras pessoas.

”A preparação é o que separa um homem comum de alguém capaz de mudar o mundo.”

Bruce Wayne, em Batman Begins (2005)

RESUMO

Este Trabalho de Conclusão de Curso tem como objetivo o desenvolvimento do Tasker, uma aplicação voltada ao gerenciamento de tarefas e projetos, com foco na melhoria do monitoramento de prazos e na geração automatizada de relatórios de desempenho. A motivação para o desenvolvimento do sistema surge da dificuldade enfrentada por pequenas empresas e equipes de desenvolvimento ao controlar prazos e consolidar dados de produtividade utilizando métodos informais ou ferramentas excessivamente complexas. A problemática central abordada consiste na alta carga cognitiva imposta aos usuários e no tempo excessivo gasto em atividades de controle e gestão, em detrimento da execução das tarefas.

O projeto mostra-se relevante para a área de Análise e Desenvolvimento de Sistemas ao propor soluções centradas na experiência do usuário, aliando automação de processos, clareza visual e apoio à tomada de decisão gerencial. O sistema foi projetado para oferecer sinalização visual passiva de prazos críticos, por meio de *badges* dinâmicos e mudanças automáticas de estado das tarefas, além da geração de relatórios gerenciais com apenas um clique.

No desenvolvimento da aplicação, foram utilizadas tecnologias modernas voltadas ao desenvolvimento web, incluindo linguagens de programação, *frameworks* e bibliotecas que possibilitaram a criação de uma interface responsiva, modular e escalável, além de um *backend* capaz de processar dados de tempo e desempenho de forma automatizada. Destacam-se como determinantes para o sucesso do projeto a adoção de uma arquitetura bem definida e o uso de ferramentas que facilitaram a integração entre *frontend*, *backend* e banco de dados.

Como considerações finais, conclui-se que o Tasker contribui de forma relevante para a otimização da gestão de tarefas e projetos, oferecendo uma solução prática, eficiente e alinhada às necessidades reais de equipes modernas, reforçando sua importância para a área de Análise e Desenvolvimento de Sistemas.

Palavras-chave: Gerenciamento de tarefas; Monitoramento de prazos; Relatórios de desempenho; Usabilidade; Sistemas de informação.

ABSTRACT

This Final Course Project aims to develop Tasker, an application designed for task and project management, focusing on improving deadline monitoring and the automated generation of performance reports. The motivation for developing the system arises from the difficulties faced by small businesses and development teams in controlling deadlines and consolidating productivity data through informal methods or overly complex tools. The central issue addressed is the high cognitive load imposed on users and the excessive amount of time spent on monitoring and management activities, to the detriment of task execution.

The project is relevant to the field of Systems Analysis and Development as it proposes user-centered solutions that combine process automation, visual clarity, and support for managerial decision-making. The system was designed to provide passive visual signaling of critical deadlines through dynamic badges and automatic task status updates, as well as the generation of management reports with a single click.

The development of the application employed modern web development technologies, including programming languages, frameworks, and libraries that enabled the creation of a responsive, modular, and scalable interface, in addition to a backend capable of automatically processing time and performance data. Key factors contributing to the project's success include the adoption of a well-defined architecture and the use of tools that facilitated integration between the frontend, backend, and database.

As a final consideration, it is concluded that Tasker makes a significant contribution to optimizing task and project management by providing a practical, efficient solution aligned with the real needs of modern teams, reinforcing its importance within the field of Systems Analysis and Development.

Keywords: Task management; Deadline monitoring; Performance reports; Usability; Information systems.

LISTA DE FIGURAS

LISTA DE QUADROS

LISTA DE ABREVIATURAS E SIGLAS

ADS	Análise e Desenvolvimento de Sistemas
DOM	Document Object Model
IFPB	Instituto Federal de Educação, Ciência e Tecnologia da Paraíba
IHC	Interação Humana-Computador
JSON	JavaScript Object Notation
JSX	JavaScript XML
JWT	JSON Web Token
NBR	Norma Brasileira
ORM	Object Relational Mapping
TCC	Trabalho de Conclusão do Curso

SUMÁRIO

1 INTRODUÇÃO

Atualmente, estamos habituados ao constante avanço da tecnologia, com ela atingindo vários campos na nossa vida, tornando o acesso à informação mais imediato e rápido. Porém, a alta demanda por novas informações e serviços se tornou um problema no mercado, já que, com a alta demanda, empresas precisariam implementar novos métodos para entregar com mais rapidez seus serviços. Atualmente, sistemas que auxiliam nesses processos se tornaram presentes no ambiente de trabalho e a busca por esse tipo de sistema tem sido recorrente em empresas de grande e pequeno porte (SANTOS; LIMA, 2019).

Em cenários onde a alta demanda e entrega rápida de serviços é recorrente, corporações que aplicam princípios de gerenciamento de projetos, como definição clara dos objetivos e metas, divisão bem estruturada das etapas de desenvolvimento, garantia das responsabilidades de cada membro, gerenciamento de riscos e estabelecimento de cronogramas, têm uma vantagem significativa em relação a outras companhias (VARGAS, 2018). Para melhorar a aplicação destes princípios no ambiente organizacional, surgiram os sistemas de gerenciamento de projetos.

Os sistemas de gerenciamento de projetos são serviços e softwares online que se baseiam fortemente nos princípios já citados. Surgiram com o intuito de auxiliar empresas e grupos à organizarem seus projetos e serem uma forma de comunicação mais clara e unificada dentro da organização. Tais sistemas assumem, muitas vezes, o papel de um ambiente corporativo virtual, estruturado em setores, equipes e projetos, promovendo maior padronização e integração das informações (REZENDE; ABREU, 2020). Este modelo, junto a ideia de organização de tarefas, datas e metas, tornou o processo mais produtivo, já que, o grupo possui uma visão completa de todo andamento do projeto, ajudando no monitoramento do progresso, onde o gestor pode gerenciar os recursos necessários e controlar prazos e custos (RODRIGUES; COSTA, 2021).

Estes sistemas proporcionaram grande avanço no processo de desenvolvimento, já que simplificaram etapas e métodos, eliminaram a dispersão de informações e fortaleceram a comunicação entre os membros da equipe, garantindo o alinhamento com o escopo definido. Dessa forma, os serviços de gerenciamento de projetos simplificam o processo de gestão, melhoram a produtividade da equipe e garantem que os objetivos sejam atingidos no limite planejado (Project Management Institute, 2017).

1.1 PROBLEMÁTICA

Entretanto, com a grande adesão de sistemas de gerenciamento de projetos e a alta demanda por softwares personalizáveis, muitos sistemas presentes no mercado tornaram-se ferramentas complexas e altamente especializadas. Embora ofereçam grande variedade de recursos e opções de customização, esses sistemas geralmente estão associados a altos custos de licenciamento e manutenção, o que dificulta sua adoção por pequenas empresas e equipes com menor disponibilidade de recursos financeiros.

A alta complexidade dessas plataformas também implica maior tempo de treinamento e adaptação, o que aumenta ainda mais os custos operacionais e pode comprometer a eficiência no uso cotidiano. Nesse contexto, organizações que não possuem capacidade financeira para investir em soluções robustas acabam recorrendo a métodos informais de organização, resultando em falhas de comunicação, descontrole de prazos e perda de produtividade.

Diante desse cenário, evidencia-se a necessidade de um sistema de gerenciamento de projetos mais simples, acessível e financeiramente viável, capaz de atender às demandas essenciais de equipes menores sem os custos e barreiras associados às plataformas consolidadas do mercado.

1.1.1 SOLUÇÃO

A solução apresentada neste trabalho tem como intuito oferecer um ambiente limpo, direcionado unicamente à organização de projetos e gerenciamento de membros, o objetivo é oferecer um sistemas com funcionalidades simples, porém robustas. Capaz de oferecer aos usuários todas as funções necessárias para realização das tarefas através de interfaces limpas. Podendo se adaptar a vários fluxos e ambientes de trabalho, sem perder seu foco inicial e sua base.

Além disso, esta solução oferece um sistema de perfis, onde o ambiente de trabalho muda de acordo com o perfil do usuário conectado, um exemplo é, caso o usuário seja um administrador do sistema, são liberadas funções que usuários comuns não possuem acesso. Restringir usuários de baixo nível garante segurança e integridade à plataforma. Além de contar com um sistema de geração de relatórios, onde usuários de níveis mais elevados podem gerar relatórios sobre projetos em andamento, indicando desempenho de membros, marcos alcançados, cronograma planejado, além das informações básicas sobre o projeto.

1.2 TRABALHOS RELACIONADOS

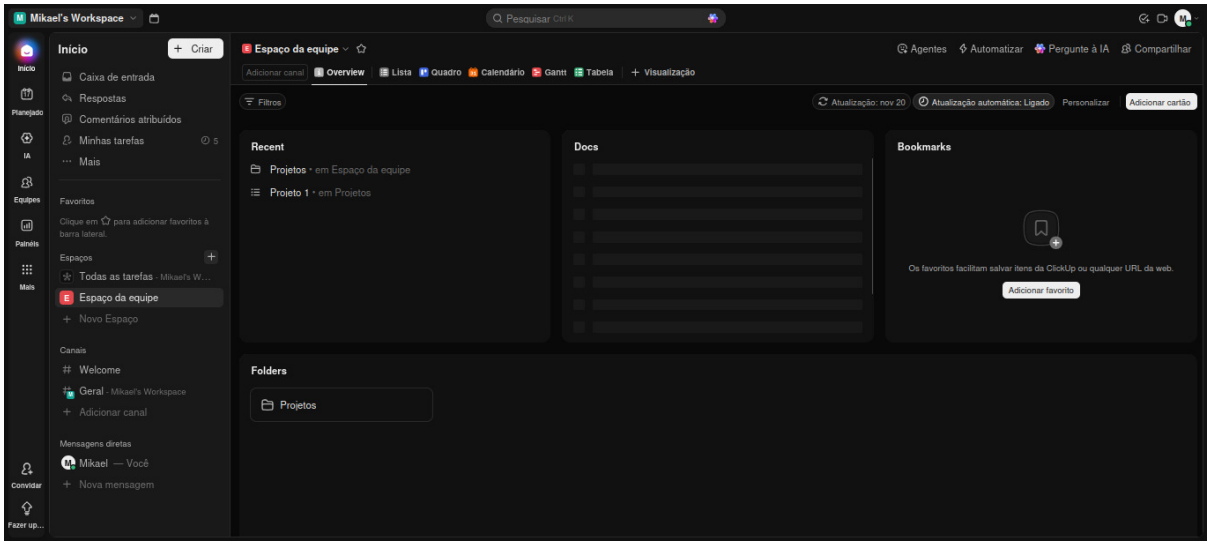
Com o crescimento de equipes e a necessidade de entregas mais rápidas, muitos sistemas de gestão de projetos surgiram para auxiliar equipes na organização de seus projetos. Dentre os vários sistemas. Neste capítulo serão apresentados alguns dos sistemas presentes do mercado e feita uma breve análise sobre suas funções.

1.2.1 ClickUp

O ClickUp é uma plataforma que ajuda no gerenciamento de projetos com uma estrutura muito similar a de uma empresa. Nele é possível criar espaços de trabalho separados, cada um possuindo seus próprios projetos e integrantes. Sendo esta uma das principais funções que o tornam tão interessante e diferente no mercado.

Porém, a grande gama de outras funcionalidades e por conta da sua alta personalização, o tornam uma ferramenta com um grau de aprendizagem elevado. Diversas formas para visualização de dados, tarefas, eventos e desempenho dos membros, além da possibilidade de criação de status para tarefas, adição de novas etapas para quadros Kanban, sem uma padronização certa, ficando a critério da equipe qual será a forma de visualização. Apesar da ideia de forte personalização, usuários ainda gastaram um tempo considerável para organizar da melhor forma o espaço de trabalho. Ferramentas com uma forte personalização exige o treinamento continuo em casos de mudança do espaço de trabalho e outras personalizações pequenas, sem contar a grande parte das funcionalidades sem uso efetivo pelas equipes. A Figura 1 apresenta como se dá a tela de visão geral de um projeto no ClickUp.

Figura 1 – Tela de visão geral de projeto do ClickUp



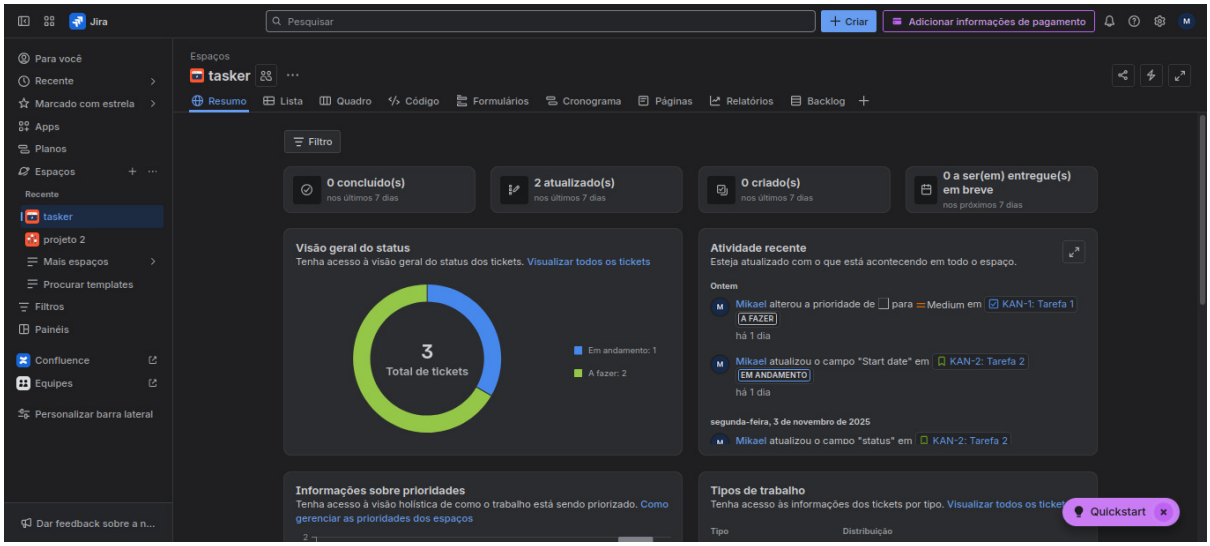
Fonte: Elaborado pelo autor

1.2.2 Jira

A plataforma Jira apresenta uma estrutura robusta, fortemente baseada em processos, permitindo controle total sobre tarefas, backlogs, sprints, épicos, por este motivo é muito utilizada por desenvolvedores. Além destas funcionalidades, esta ferramenta possui forte integração com outros serviços da Atlassian, desenvolvedora da plataforma, permitindo a possibilidade de documentação, além da aplicação de conceitos de gerenciamento de versão, apresentadas no sistema Git.

Para usuários experientes pode ser uma ferramenta poderosa, contudo, a primeira experiência pode ser frustrante, justamente pela necessidade de compreender conceitos específicos da plataforma, bem como, conhecimentos de metodologias ágeis como Scrum e Kanban, isto, aliada à grande quantidade de campos e opções, aumenta significativamente o esforço mental dos usuários, exigindo um treinamento ou estudo prévio da plataforma, para enfim utilizá-la de maneira mais eficiente. Caso contrário, a falta deste estudo pode impactar negativamente o desempenho do usuário. A Figura 2 apresenta como se dá a tela de visão geral de um projeto no Jira.

Figura 2 – Tela de visão geral de projeto do Jira



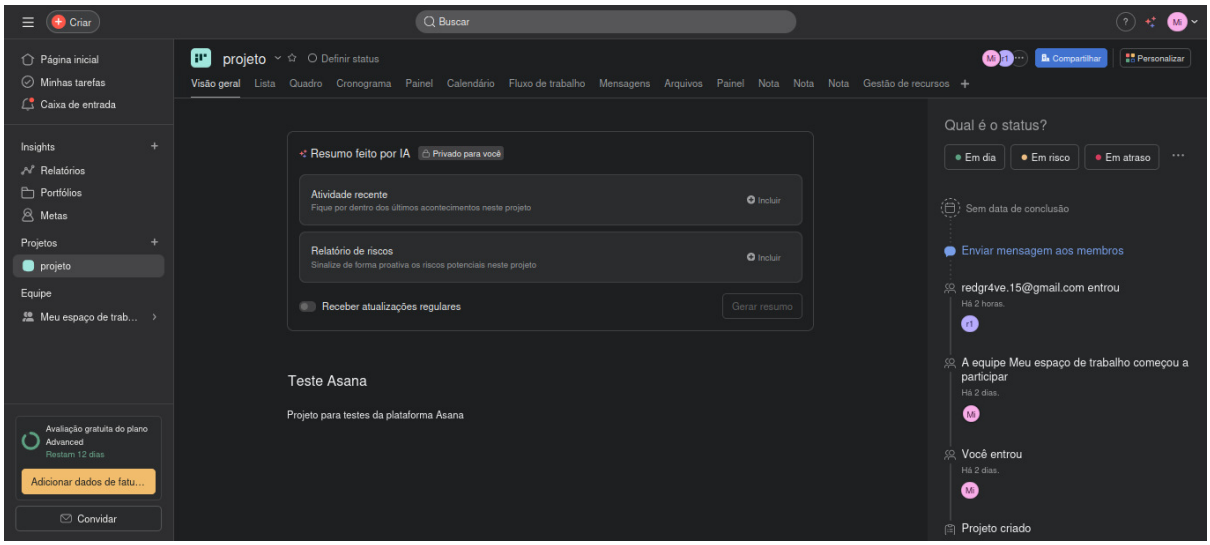
Fonte: Elaborado pelo autor

1.2.3 Asana

O Asana é uma plataforma de gerenciamento de projetos onde seu foco principal está na clareza das tarefas, prazos e responsabilidades, oferecendo uma interface visual mais amigável. A ferramenta se destaca por sua usabilidade mais amigável e por incentivar a colaboração entre os membros da equipe, permitindo a centralização de informações, comentários e atualizações em um único ambiente de trabalho.

Por conta da proposta mais acessível, o Asana apresenta pouca distinção de "cargos"entre membros de cada projeto, sendo separados em Administrador, Editor, Comentador e Visualizador, onde somente o Administrador e Editor têm acesso efetivo ao projeto. Ainda assim, sem muitas distinções entre ambos, onde membros definidos como Editores podem designar tarefas para outros membros, adicionar e modificar arquivos importantes, adição de notas — arquivos de texto simples usados para alinhamento da equipe — além de poder gerenciar recursos avançados como gestão de recursos e análise do desempenho de membros. Apesar da ideia inicialmente ser interessante, isso se mostra um problema, especialmente equipes maiores ou que possuem forte definição de cargos, já que a plataforma não garante que os usuários seguiram firmemente suas funções e restrições. A Figura 3 apresenta como se dá a tela de visão geral de um projeto no Asana.

Figura 3 – Tela de visão geral de projeto do Asana



Fonte: Elaborado pelo autor

1.2.4 Análise Comparativa

Quadro 1 – Comparação de funcionalidades entre ClickUp, Asana, Jira e Tasker

Funcionalidade	ClickUp	Asana	Jira	Tasker
Definição de cargos e permissões	Moderado (Pago)	Fraco	Moderado (Pago)	Forte
Curva de aprendizado	Alta	Média	Alta	Baixa
Eventos e calendário	Sim	Sim	Somente com extensões	Sim
Relatórios de desempenho	Pago	Pago	Pago	Gratuito
Geração de relatório em PDF	Pago	Pago	Pago	Gratuito
Acesso a funcionalidades	Gratuito (Somente o básico)	Pago	Gratuito (Somente o básico)	Gratuito (Todas as funcionalidades)

Fonte: Elaborado pelo autor

1.3 POTENCIAL DE COMERCIALIZAÇÃO

O sistema apresentado tem grande potencial de ser implementado em empresas que buscam maior eficiência e controle em seus projetos. Sua proposta de simplicidade, baixo custo operacional e foco nas funcionalidades essenciais o posiciona como uma alternativa competitiva frente às soluções tradicionais de mercado, que muitas vezes apresentam alto nível de complexidade e custos de licenciamento elevados. Dessa forma, o sistema torna-se especialmente atrativo para pequenas e médias empresas, startups, equipes independentes e organizações em fase de crescimento, que necessitam de uma ferramenta eficiente, acessível e de fácil adoção.

Além disso, o modelo de desenvolvimento modular possibilita a expansão progressiva de recursos conforme a necessidade do cliente, favorecendo estratégias como planos de assinatura escalonados ou licenciamento por uso. Essa característica amplia o alcance comercial do sistema e permite sua adaptação a diferentes segmentos, como escritórios de tecnologia, agências de marketing, empresas de engenharia, equipes acadêmicas e grupos de desenvolvimento de software.

O baixo tempo de treinamento exigido e a interface simplificada também contribuem para a redução de barreiras de entrada no mercado, facilitando sua adoção em ambientes corporativos. Com isso, o sistema apresenta forte potencial para consolidação comercial, tanto como produto independente quanto como solução integrável a outros serviços corporativos.

1.4 OBJETIVOS

1.4.1 Objetivo Geral

Construir um sistema de gerenciamento de projetos, com a capacidade de organizar projetos e suas equipes, definir datas importantes e o principal diferencial, seria a possibilidade de gerar relatórios sobre projetos, com informações sobre marcos e eventos importantes, estatísticas sobre membros e informações importantes sobre o projeto.

1.4.2 Objetivos Específicos

- Realizar a documentação do sistema, identificando as características essenciais, especificando requisitos e definindo a estrutura básica do software.
- Criação de um protótipo como base do visual para o sistema.
- Consolidação do sistema, dentro do escopo estabelecido, por meio da definição organização de seus requisitos e detalhes técnicos.

1.5 ORGANIZAÇÃO DO DOCUMENTO

Como visto até este momento, foi apresentado uma breve contextualização e descrição sobre o sistema desenvolvido, porém, a partir deste capítulo, serão apresentados os seguintes temas.

O capítulo 2 apresentará a metodologia utilizada para a realização das atividades relacionadas ao desenvolvimento do sistema.

O capítulo 3 abordará sobre como o sistema funciona, especificando seus requisitos, arquitetura, bibliotecas e frameworks utilizados.

O capítulo 4 apresentará as considerações finais sobre o presente trabalho.

2 METODOLOGIA

Neste capítulo é apresentada a metodologia utilizada no processo de desenvolvimento da plataforma, descrevendo como ocorreu o desenvolvimento inicial e destacando as atividades realizadas durante sua construção.

2.1 PROCESSO DE DESENVOLVIMENTO

O processo de desenvolvimento inicial da plataforma se deu de forma incremental, de acordo com (PRESSMAN; MAXIM, 2021), o modelo incremental separa o sistema em pequenos módulos ou incrementos, cada um desenvolvido de forma independente. Cada módulo representa um aspecto específico do sistema. Esta metodologia permite que o sistema evolua de forma gradual, conforme cada módulo é desenvolvido e validado. Com o desenvolvimento dos módulos ocorrendo da seguinte maneira.

USUÁRIOS Voltado para o gerenciamento de contas e usuários, responsável pela criação, remoção, edição e autenticação dos usuários. Além de definir a hierarquia do sistema. Representa a base do serviço, já que as demais funcionalidades dependem da identificação e validação dos usuários.

ORGANIZAÇÕES Responsável pela criação e pelo gerenciamento dos espaços de trabalho da plataforma. Este módulo reúne usuários e projetos em um mesmo contexto, estabelece o proprietário da organização e utiliza as afiliações para definir a participação e o papel de cada usuário no ambiente organizacional.

PROJETOS Segmento responsável pelo gerenciamento dos projetos na plataforma. Define o contexto dos projetos, sua principais informações, tarefas, eventos e objetivos. Sendo o principal elo entre usuários e os demais módulos.

MEMBROS Este módulo controla a relação entre usuários e projetos. Tem como principal responsabilidade a organização das equipes em cada projeto e a base para a gestão de usuários e definição de atividades.

TAREFAS Responsável pela organização e a gestão das atividades a serem realizadas em cada projeto. Ele permite a atribuição de responsabilidades, o monitoramento do progresso e o controle do estado das atividades, sendo um dos principais recursos para o gerenciamento do fluxo de trabalho na plataforma.

EVENTOS Tem como responsabilidade registrar prazos, reuniões, marcos e outras ocorrências relevantes relacionadas aos projetos. Este segmento contribui para a organização temporal do sistema, auxiliando os usuários no cumprimento de prazos e na visualização de compromissos importantes associados às atividades do projeto.

COMENTÁRIOS Representa a principal forma de comunicação entre os membros. Promovendo a troca de informações e registro de decisões relacionadas a projetos em um ambiente centralizado, evitando assim a necessidade de serviços externos.

O processo incremental tem como fundamentos a definição clara de objetivos, a priorização de funcionalidades, o desenvolvimento contínuo, a realização de testes e a implantação progressiva de cada incremento. Esta abordagem possibilita que sejam realizados ajustes, correções e validação antecipada de cada parte do sistema antes de prosseguir para os próximos módulos, (PRESSMAN; MAXIM, 2021). Por conta dessas características, a implementação do modelo incremental contribuiu para garantir que os módulos subsequentes não herdassem problemas estruturais ou conceituais das etapas anteriores, uma vez que cada módulo está diretamente acoplado ao anterior. Desta forma, foi possível assegurar maior estabilidade do sistema ao longo de seu desenvolvimento.

2.2 ATIVIDADES REALIZADAS

Seguindo o modelo de implementação incremental e os requisitos e estruturas especificados neste trabalho, o processo de desenvolvimento envolveu a implementação dos requisitos levantados e a adaptação do serviço, do *backend* e do *frontend* ao modelo apresentado. Para facilitar a organização e o acompanhamento do trabalho, as atividades realizadas foram divididas da seguinte maneira.

2.2.1 Desenvolvimento da aplicação base

Inicialmente, foi desenvolvida a estrutura fundamental da aplicação, responsável por sustentar o restante do sistema. Nessa etapa, os módulos apresentados foram transformados em entidades de domínio, definindo, para cada recurso, os dados necessários, relacionamentos com outras entidades, regras de validação, regras de negócio e restrições de domínio. Em seguida, foram implementadas as operações básicas de criação, leitura, atualização e exclusão de dados, conhecidas como CRUD (Amazon Web Services, 2026). Essas operações permitiram validar a persistência dos dados e comportamento individual de cada recurso, antes da implementação de

funcionalidades mais complexas. Resultando em uma base funcional e padronizada para a extensão do restante da plataforma.

2.2.2 Análise da situação inicial do sistema

Após a construção da aplicação base descrita no ponto anterior, foi realizada uma análise técnica para determinar como cada requisito definido para o sistema poderia ser aplicado individualmente à estrutura existente. Nessa etapa, os requisitos foram relacionados às entidades de domínio, aos serviços, às operações CRUD e às regras de negócio já implementadas, permitindo identificar quais componentes poderiam ser reutilizados e quais precisariam ser adaptados ou ampliados.

A análise também considerou os dados necessários para cada funcionalidade, os relacionamentos entre as entidades, as validações, as restrições de acesso e os possíveis impactos sobre os demais módulos. Como resultado, foram definidos os ajustes estruturais e as novas operações necessárias para incorporar cada requisito à aplicação base. Esse mapeamento orientou a etapa posterior de implementação, reduzindo inconsistências e evitando alterações que comprometessem o funcionamento dos recursos anteriormente desenvolvidos.

2.2.3 Aplicação dos requisitos no *backend*

Com base no mapeamento realizado na etapa anterior, os requisitos foram incorporados ao *backend* da aplicação. Para cada funcionalidade, foram implementadas ou ajustadas as operações responsáveis por receber as solicitações, validar os dados, aplicar as regras de negócio e realizar a persistência das informações. Dessa forma, a estrutura básica deixou de oferecer apenas operações genéricas e passou a representar o comportamento específico esperado para cada recurso do Tasker.

Também foram aplicadas as regras de autenticação e autorização relacionadas aos diferentes perfis de usuário. As operações passaram a considerar o papel exercido pelo usuário na organização, sua participação nos projetos e as permissões necessárias para acessar ou modificar cada recurso. Foi durante estapa, que também foram implementados requisitos referentes às estatisticas e relatórios, o módulo de projetos foi estendido para essa nova funcionalidade, com o sistema compilando as principais informações do . Essa etapa consolidou no *backend* as regras funcionais e de acesso definidas para o sistema, preparando a API para ser consumida pela interface.

2.2.4 Adaptação do protótipo visual

Após a consolidação das funcionalidades no *backend*, o protótipo visual foi revisado para representar os requisitos e os fluxos de interação definidos para o sistema. Cada operação foi associada a uma tela, formulário, botão ou elemento de navegação, permitindo visualizar como o usuário acessaria as funcionalidades e quais informações seriam apresentadas durante cada ação.

Nessa etapa, também foram definidos a organização dos componentes, a hierarquia das informações e os caminhos de navegação entre as telas. As interfaces foram diferenciadas conforme os perfis de Membro, Gestor e Proprietário, de modo que o protótipo representasse as permissões e responsabilidades de cada usuário. Como resultado, foi estabelecida uma referência visual coerente para orientar a implementação do *frontend*.

2.2.5 Adaptação do *frontend*

Com o protótipo revisado, suas telas e fluxos foram convertidos em componentes funcionais no *frontend*. Foram implementados os elementos de navegação, formulários, painéis, listagens e mensagens de retorno necessários para permitir a interação do usuário com a aplicação. Os componentes compartilhados foram reutilizados entre diferentes telas, contribuindo para a padronização visual e para a redução de código repetido.

Além da construção visual, foram definidos os comportamentos da interface, como validação dos campos, abertura de formulários, atualização das informações exibidas e apresentação de estados de carregamento, sucesso ou erro. A exibição das ações também foi ajustada conforme o perfil do usuário, mantendo no *frontend* a mesma organização de permissões estabelecida pelo sistema. Ao final dessa etapa, a interface estava preparada para trocar dados com a API.

2.2.6 Integração entre a API e a interface

Por fim, o *frontend* foi conectado aos recursos disponibilizados pela API. As ações realizadas nas telas passaram a enviar requisições HTTP ao *backend*, enquanto as respostas recebidas foram utilizadas para atualizar os dados apresentados ao usuário. Também foram configurados o envio das informações de autenticação, o tratamento dos códigos de resposta e a apresentação de mensagens adequadas para operações bem-sucedidas ou rejeitadas.

Durante a integração, os fluxos completos foram verificados desde a entrada

dos dados na interface até sua validação e persistência no banco de dados. Essa verificação permitiu identificar incompatibilidades nos formatos das informações, falhas de comunicação e diferenças entre os comportamentos previstos no protótipo e aqueles fornecidos pela API. Os ajustes realizados asseguraram o funcionamento conjunto das camadas e concluíram a integração das principais funcionalidades da aplicação.

3 DESCRIÇÃO DO SISTEMA

O sistema Tasker é uma aplicação para gerenciamento de projetos e equipes, desenvolvido com o intuito de melhorar o fluxo de trabalho tanto para grandes equipes, quanto para pequenos grupos ou projetos individuais. O sistema se baseia na simplicidade, através de interfaces limpas, ações diretas e a utilização de somente informações necessárias.

3.1 ATORES

- **Membro**: Usuário comum, sem muitas permissões, representando o funcionário comum da empresa.
- **Gestor**: Usuário de nível mais alto, possui permissões de edição e criação de projetos, adição de pessoas e geração de relatórios. Representa o gerente de projetos ou o *Project Owner* da empresa.
- **Proprietário**: Usuário de nível máximo no sistema, com permissões para remoção e adição de membros, criação e edição de projetos e geração de relatórios.

3.2 REQUISITOS

Esta seção apresenta os requisitos funcionais e não funcionais atendidos pelo sistema Tasker. Os requisitos abrangem a persistência e o gerenciamento de usuários, organizações, projetos, tarefas e eventos, a geração de relatórios de desempenho e as restrições de acesso.

Para fins de padronização, os requisitos são divididos em blocos e identificados da seguinte maneira:

- **Requisitos funcionais**: definem as funcionalidades e ações que o sistema deve disponibilizar aos usuários. Cada requisito utiliza o padrão [Requisito.RF000], no qual “Requisito” representa seu nome e RF000 corresponde ao código identificador.
- **Requisitos não funcionais**: definem as características e restrições que o sistema deve atender e que influenciam seu funcionamento. Cada requisito utiliza o padrão [Requisito.NF000], no qual “Requisito” representa seu nome e NF000 corresponde ao código identificador.

3.2.1 Requisitos funcionais

[RF1] Cadastro de conta: O sistema deve permitir que um usuário informe e-mail e senha para criar uma nova conta. O e-mail deve ser válido, a senha deve atender às regras de segurança definidas e seu conteúdo deve ser armazenado de forma criptografada antes da persistência no banco de dados.

Atores: Usuário

Prioridade: ☒ **Essencial** ☐ Importante ☐ Desejável

[RF2] Exclusão de conta: O sistema deve permitir a remoção de uma conta por meio de uma requisição autenticada, utilizando o e-mail associado como identificador. Ao concluir a operação, o sistema deve retornar a confirmação da exclusão.

Atores: Usuário

Prioridade: ☒ **Essencial** ☐ Importante ☐ Desejável

[RF3] Cadastro de usuário: O sistema deve permitir que um usuário informe nome, *username* e chave de conta para criar um novo cadastro. O *username* deve ser único, e o sistema deve impedir a criação caso já exista um usuário com o mesmo identificador.

Atores: Usuário

Prioridade: ☒ **Essencial** ☐ Importante ☐ Desejável

[RF4] Consulta de usuário: O sistema deve permitir que um usuário autenticado pesquise um usuário específico por nome ou *username*, retornando os dados correspondentes quando o registro existir.

Atores: Usuário

Prioridade: ☒ **Essencial** ☐ Importante ☐ Desejável

[RF5] Exclusão de usuário: O sistema deve permitir que um usuário autenticado remova um usuário informando o *username* correspondente. Ao concluir a operação, o sistema deve retornar a confirmação da exclusão.

Atores: Usuário

Prioridade: ☒ **Essencial** ☐ Importante ☐ Desejável

[RF6] Cadastro de organização: O sistema deve permitir que um usuário autenticado informe o nome da organização para criar um novo cadastro. Ao concluir a operação, o sistema deve associar automaticamente o usuário autenticado como proprietário da organização.

Atores: Usuário

Prioridade: ☒ **Essencial** ☐ Importante ☐ Desejável

[RF7] Listagem de afiliações: O sistema deve permitir que um usuário autenticado consulte as afiliações relacionadas à organização informada, retornando os vínculos associados à organização selecionada.

Atores: Usuário

Prioridade: ☒ **Essencial** ☐ Importante ☐ Desejável

[RF8] Exclusão de afiliação: O sistema deve permitir que o proprietário de uma organização remova uma afiliação informando o identificador do vínculo. Ao concluir a operação, o sistema deve retornar a confirmação da remoção.

Atores: Proprietário

Prioridade: ☒ **Essencial** ☐ Importante ☐ Desejável

[RF9] Promoção de afiliação: O sistema deve permitir que o proprietário de uma organização promova a função de uma afiliação, elevando o nível de acesso do vínculo conforme as regras definidas.

Atores: Proprietário

Prioridade: ☒ **Essencial** ☐ Importante ☐ Desejável

[RF10] Rebaixamento de afiliação: O sistema deve permitir que o proprietário de uma organização rebaixe a função de uma afiliação, reduzindo o nível de acesso do vínculo conforme as regras definidas.

Atores: Proprietário

Prioridade: ☒ **Essencial** ☐ Importante ☐ Desejável

[RF11] Convite para organização: O sistema deve permitir que o Proprietário gere convites por *token* para ingresso em sua organização. O sistema deve permitir que o usuário consulte os dados públicos do convite e, após autenticar-se, aceite-o ou rejeite-o. No aceite, o sistema deve vincular o usuário à organização com o cargo de Membro.

Atores: Proprietário e Usuário

Prioridade: ☒ **Essencial** ☐ Importante ☐ Desejável

Restrições: Cada convite deve ser temporário, de uso único e conceder exclusivamente o cargo de Membro. O sistema deve invalidar o convite após sua expiração, aceite ou rejeição.

[RF12] Cadastro de projeto: O sistema deve permitir que o usuário informe título, descrição e data de entrega para criar um novo projeto. O projeto deve ser vinculado à organização do usuário autenticado.

Atores: Proprietário

Prioridade: ☒ **Essencial** ☐ Importante ☐ Desejável

Restrições: exige usuário autenticado; a criação é permitida somente para usuário com permissão de proprietário da organização.

[RF13] Listagem de projetos: O sistema deve permitir que um usuário autenticado consulte os projetos vinculados à organização em uso. A listagem deve se adaptar ao papel exercido pelo usuário na organização.

Atores: Usuário

Prioridade: ☒ **Essencial** ☐ Importante ☐ Desejável

Restrições: exige autenticação e contexto de organização; o acesso à listagem é validado pelo sistema de permissões.

Especificações: Para Membros (*Members*), a lista retorna são somente os projetos que o usuário participa, para Gestores (*Managers*) é retornado a lista de projetos que ele gerencia e para Proprietários (*Owner*) é a lista de todos os projetos da organização.

[RF14] Consulta de projeto: O sistema deve permitir que um usuário autenti-

cado consulte um projeto específico pelo identificador, retornando seus dados e seus membros associados.

Atores: Usuário

Prioridade: ☒ **Essencial** ☐ Importante ☐ Desejável

Restrições: exige autenticação e validação de permissão pelo sistema antes de retornar os dados do projeto.

[RF15] Edição de projeto: O sistema deve permitir que o usuário altere os dados de um projeto existente, incluindo título, descrição, data de entrega, status, estágio e gestor do projeto.

Atores: Proprietário

Prioridade: ☒ **Essencial** ☐ Importante ☐ Desejável

Restrições: exige usuário autenticado; a edição é permitida somente para usuário com permissão de proprietário da organização.

[RF16] Exclusão de projeto: O sistema deve permitir que o usuário remova um projeto informando o identificador correspondente.

Atores: Proprietário

Prioridade: ☒ **Essencial** ☐ Importante ☐ Desejável

Restrições: exige usuário autenticado; a exclusão é permitida somente para usuário com permissão de proprietário da organização.

[RF17] Cadastro de membro: O sistema deve permitir que um usuário informe os identificadores do projeto e do usuário para adicionar um novo membro ao projeto. Ao concluir a operação, o sistema deve registrar o vínculo entre o usuário e o projeto.

Atores: Proprietário

Prioridade: ☒ **Essencial** ☐ Importante ☐ Desejável

Restrições: exige usuário autenticado; a inclusão é permitida apenas para usuário com permissão de criação no recurso de membros.

[RF18] Listagem de membros: O sistema deve permitir que um usuário autenticado consulte os membros vinculados ao projeto informado, retornando os registros associados ao projeto em uso.

Atores: Usuário

Prioridade: ☒ **Essencial** ☐ Importante ☐ Desejável

Restrições exige autenticação; a consulta é feita a partir do identificador do projeto informado e depende da validação de permissão pelo sistema.

[RF19] Exclusão de membro: : O sistema deve permitir que um usuário remova um membro informando o identificador do vínculo. Ao concluir a operação, o sistema deve retornar a confirmação da remoção.

Atores: Proprietário

Prioridade: ☒ **Essencial** ☐ Importante ☐ Desejável

Restrições: exige usuário autenticado; a exclusão é permitida apenas para usuário com permissão de remoção no recurso de membros.

[RF20] Gerenciamento de tarefas: O sistema deve permitir que um usuário crie, consulte, liste, edite e remova tarefas, informando nome, descrição, projeto, responsável, prioridade, data de entrega e estágio conforme a operação desejada. Ao concluir a operação, o sistema deve manter o vínculo da tarefa com o projeto informado e retornar as informações correspondentes ao registro processado.

Atores: Usuário

Prioridade: ☒ **Essencial** ☐ Importante ☐ Desejável

Restrição: O sistema deve restringir a definição e a alteração do responsável pela tarefa aos usuários com perfil de Proprietário ou Gestor. O sistema deve permitir que usuários com níveis inferiores atribuam tarefas somente a si mesmos.

[RF21] Verificar prazo de tarefas: O sistema deve sinalizar ao usuário quando uma tarefa estiver atrasada ou dentro do prazo esperado.

Atores: Usuário

Prioridade: ☐ Essencial ☒ **Importante** ☐ Desejável

[RF22] Gerenciamento de comentários: : O sistema deve permitir que um usuário crie, consulte, liste e remova comentários, informando conteúdo, data, projeto e autor conforme a operação desejada. Ao concluir a operação, o sistema deve manter o vínculo do comentário com o projeto informado e retornar as informações correspondentes ao registro processado.

Atores: Usuário

Prioridade: ☒ **Essencial** ☐ Importante ☐ Desejável

Restrições: exige usuário autenticado; o acesso às operações depende de validação de permissão no recurso de comentários disponível somente para membros do projeto.

[RF23] Gerenciamento de eventos: : O sistema deve permitir que um usuário crie, consulte, liste, edite e remova eventos, informando título, projeto, data e categoria conforme a operação desejada. Ao concluir a operação, o sistema deve manter o vínculo do evento com o projeto informado e retornar as informações correspondentes ao registro processado.

Atores: Gestor

Prioridade: ☒ **Essencial** ☐ Importante ☐ Desejável

Restrições: exige usuário autenticado; o acesso às operações depende de validação de permissão no recurso de eventos.

[RF24] Geração de relatórios: O sistema deve gerar relatórios sobre projetos e membros, indicando o andamento do projeto, o desempenho de cada membro e a situação das tarefas atribuídas.

Atores: Gestor e Proprietário

Prioridade: ☒ **Essencial** ☐ Importante ☐ Desejável

[RF25] Verificar estatísticas: O sistema deve exibir estatísticas relacionadas aos membros participantes dos projetos, incluindo o total de tarefas atribuídas, atrasadas e concluídas.

Atores: Gestor e Proprietário

Prioridade: ☐ Essencial ☐ Importante ☒ **Desejável**

[RF26] Definir tipos de eventos: O sistema deve catalogar os eventos como *RELEASE* (Entrega), *MEETING* (Reunião), *REVIEW* (Revisão), *PLANNING* (Planejamento), *TESTS* (Testes) ou *LAUNCH* (Lançamento).

Atores: Usuário

Prioridade: ☐ Essencial ☐ Importante ☒ **Desejável**

[RF27] Atribuição do estado PENDING na criação de um projeto: O sistema deve atribuir automaticamente o estado *PENDING* (Pendente) ao criar um projeto, utilizando-o como indicador do estado inicial do projeto.

Atores: Sistema

Prioridade: ☐ Essencial ☐ Importante ☒ **Desejável**

[RF28] Registro da data de início de um projeto: O sistema deve registrar a data de início do projeto. Ao alterar o *status* do projeto de *PENDING* (Pendente) para *STARTED* (Iniciado), o sistema deve registrar a data da modificação.

Prioridade: ☐ Essencial ☐ Importante ☒ **Desejável**

3.2.2 Requisitos não-funcionais

Usabilidade

Esta seção apresenta os requisitos relacionados à usabilidade, ao comportamento da aplicação nos computadores e à sua forma de distribuição.

[NF1] Acesso web: O sistema deve ser disponibilizado por meio de uma aplicação web acessível pela internet. O sistema deve permitir o acesso por computadores que utilizem navegadores atualizados.

Prioridade: ☒ **Essencial** ☐ Importante ☐ Desejável

[NF2] Interface amigável: O sistema deve apresentar uma interface intuitiva, permitindo que novos usuários aprendam suas funcionalidades básicas sem treinamento prévio.

Prioridade: ☒ **Essencial** ☐ Importante ☐ Desejável

Segurança

Esta seção apresenta os requisitos relacionados à segurança das informações dos usuários e à integridade e autenticidade dos dados do sistema.

[NF3] Autorização de acesso ao sistema: O sistema deve restringir o acesso às informações de acordo com o perfil de cada usuário. Cada usuário deve acessar somente as funcionalidades definidas para seu perfil.

Prioridade: ☒ **Essencial** ☐ Importante ☐ Desejável

[NF4] Logout automático: O sistema deve encerrar automaticamente a sessão do usuário após um período de cinco dias de inatividade, removendo as informações de autenticação mantidas durante a sessão.

Prioridade: ☐ Essencial ☒ **Importante** ☐ Desejável

3.3 PRINCIPAIS FUNCIONALIDADES

Esta seção apresenta as principais funcionalidades do Tasker, destacando os mecanismos de controle de acesso, ingresso em organizações e acompanhamento do desempenho de projetos e equipes.

3.3.1 Permissionamento por Papel

O sistema de permissionamento do Tasker foi desenvolvido por meio de uma abordagem híbrida que combina o controle de acesso baseado em papéis, denominado *Role-Based Access Control* (RBAC), com o controle de acesso baseado em atributos, conhecido como *Attribute-Based Access Control* (ABAC). No RBAC, as permissões são associadas aos papéis existentes na organização e os usuários recebem esses papéis de acordo com suas responsabilidades. Essa organização reduz a necessidade de atribuir permissões individualmente e simplifica a administração e a revisão dos privilégios concedidos (SILVA et al., 2023).

O ABAC complementa esse modelo ao considerar atributos relacionados ao usuário, ao recurso solicitado, à operação pretendida e ao contexto em que a requisição ocorre. Dessa forma, a autorização não depende somente do papel do usuário, mas também de regras e relações que determinam se a operação pode ser realizada sobre um objeto específico. A combinação das duas técnicas permite que o papel estabeleça o conjunto máximo de permissões do usuário, enquanto os atributos atuam como restrições adicionais, unindo a organização do RBAC à flexibilidade e à granularidade do ABAC (SANTOS, 2024).

No Tasker, o RBAC utiliza os papéis `OWNER`, `MANAGER` e `MEMBER`, correspondentes aos perfis de Proprietário, Gestor e Membro. Como o papel depende da organização ativa, ele é obtido a partir da afiliação do usuário, e não armazenado no token JWT. Após a autenticação, os decorators de papel, ação e recurso descrevem as permissões exigidas pelo endpoint, enquanto o `PermissionGuard` reúne a identidade, a organização e o recurso solicitado, formando assim um contexto de acesso. Esse contexto é enviado ao serviço de permissão, que verifica se o papel atual do usuário permite a operação.

Após a validação do papel, o sistema monta uma chave composta pelo papel, pelo recurso e pela ação, separados por dois-pontos, e consulta o registro de validadores de acesso, preenchido durante a inicialização da aplicação. Cada chave é associada a uma *policy* responsável pela verificação específica da operação. Essas políticas aplicam os atributos do modelo ABAC ao verificar condições como propriedade da organização, participação ou gerenciamento de um projeto, responsabilidade por uma tarefa e autoria de um comentário. Assim, possuir um papel compatível não é suficiente para liberar uma ação: o usuário também deve satisfazer as relações exigidas pelo recurso solicitado.

O mecanismo adota uma estratégia de negação por padrão. Quando não existe uma política registrada, quando o contexto organizacional não é informado ou quando alguma condição de domínio não é satisfeita, o acesso é bloqueado. A separação entre guards, serviço de permissão, registro e políticas evita condicionais de autorização espalhadas pelos controllers, facilita a inclusão de novos recursos e mantém as regras específicas isoladas e testáveis. No frontend, o papel pode ser utilizado para ocultar ou desabilitar controles da interface, porém a decisão definitiva permanece no backend, impedindo que alterações na camada de apresentação contornem o permissionamento.

3.3.2 Convites via tokens criptografados

O sistema de convites permite que somente o Proprietário autorize o ingresso de novos usuários em sua organização. Cada convite é temporário, de uso único e concede exclusivamente o papel de Membro. O usuário precisa estar autenticado para aceitar ou rejeitar o convite, enquanto a organização, o papel e a validade são definidos pelo backend.

A proteção do convite utiliza uma assinatura criptográfica específica, independente do segredo empregado nos tokens JWT de sessão. O conteúdo funcional assinado contém somente o identificador da organização e o instante de expiração. Informações como nome da organização, usuário convidado, papel, URL do frontend

não são incluídas no token. Embora o frontend possa decodificar seus dados para apresentação inicial, apenas o backend verifica a assinatura e determina se o convite ainda pode ser utilizado.

O token bruto é devolvido somente no momento da criação. Antes da persistência, o sistema calcula seu resumo criptográfico por meio do algoritmo SHA-256 e armazena apenas o resultado no banco de dados. O SHA-256 é uma função de resumo que transforma os dados recebidos em uma sequência de tamanho fixo, formada por 256 bits e normalmente representada por 64 caracteres hexadecimais, letras e números. Uma alteração nos dados de entrada produz um resumo diferente, permitindo verificar se o mesmo conteúdo foi apresentado novamente sem armazená-lo diretamente (Instituto Nacional da Propriedade Industrial, 2018). Dessa maneira, uma exposição da tabela de convites não fornece diretamente as credenciais utilizadas nos links. A validação posterior calcula novamente o hash do token recebido, localiza o registro correspondente e compara o identificador da organização armazenado com aquele protegido pela assinatura.

O estado do convite registra sua organização, expiração, utilização ou rejeição. Sua validade é calculada no momento da consulta e exige que a assinatura seja autêntica, que o prazo ainda não tenha terminado, que o registro exista e que nenhum estado terminal tenha sido preenchido. Desse modo, um token corretamente assinado ainda pode ser considerado indisponível quando já tiver sido utilizado, rejeitado, revogado ou expirado. A consulta prévia apresenta apenas os dados públicos da organização e não altera o estado do convite.

O aceite e a rejeição são realizados de forma transacional para garantir que apenas uma ação consiga consumir o convite. No aceite, a afiliação como Membro e a marcação do token como utilizado acontecem em uma única operação; caso alguma etapa falhe, nenhuma alteração é mantida. A rejeição e a revogação também encerram definitivamente a validade do convite, impedindo seu uso posterior.

O backend retorna somente o token e sua expiração. O frontend monta o endereço de compartilhamento com a própria URL da aplicação e mantém a credencial apenas durante o fluxo de ingresso. Essa separação reduz a exposição do token e mantém no backend todas as decisões relacionadas à validade e à criação da afiliação.

3.3.3 Estatísticas e Relatórios

O sistema de estatísticas reúne as principais informações operacionais de um projeto para apresentar uma visão consolidada de seu andamento e do desempenho

da equipe. A funcionalidade é disponibilizada somente para Gestores e Proprietários e abrange o resumo das tarefas, o prazo, a saúde do projeto, a produtividade dos participantes e a geração de relatórios.

Os dados são obtidos por meio das relações entre projeto, tarefas, membros, afiliações e usuários. O vínculo do membro com a afiliação permite recuperar nome e nome de usuário, enquanto as tarefas fornecem estágio, responsável, prazo e datas de início e conclusão. Os dados para a geração das estatísticas são filtrados pelo mês que foi realizada a requisição. Após a consulta, o serviço transforma essas entidades em um objeto agregado dividido em informações do projeto, resumo quantitativo, prazo, saúde do projeto, desempenho, produtividade, membros.

Os indicadores básicos são derivados do estado das tarefas. O progresso é calculado pela divisão da quantidade de tarefas concluídas pelo total de tarefas e convertido em percentual. As demais atividades são agrupadas como abertas, iniciadas ou em revisão. O atraso é tratado independentemente do estágio, considerando tarefas marcadas como atrasadas, atividades abertas cujo prazo foi ultrapassado e tarefas concluídas depois de sua data limite. Os dias restantes são obtidos pela diferença entre a data de entrega do projeto e a data atual.

O cálculo da saúde do projeto combina esses indicadores para estimar o risco de descumprimento do prazo. Inicialmente, o sistema compara o percentual de trabalho concluído com a quantidade de tarefas ainda abertas. Em seguida, calcula a proporção de tarefas atrasadas e observa o ritmo de conclusão registrado no período. Esse ritmo é utilizado para estimar quanto tempo seria necessário para finalizar o trabalho restante e produzir uma previsão de entrega. A previsão é então comparada com a data limite do projeto, considerando também a quantidade de dias disponíveis.

Quando o ritmo observado indica que as tarefas restantes podem ser concluídas antes ou até a data de entrega e o volume de atrasos é baixo, o projeto recebe a classificação `SAFE`. A classificação `WARNING` é utilizada quando a previsão se aproxima do prazo, o progresso é inferior ao esperado ou existe uma quantidade relevante de tarefas atrasadas. O estado `CRITICAL` é atribuído quando a projeção ultrapassa a data de entrega, o prazo já venceu com trabalho pendente ou o volume de atrasos compromete significativamente a conclusão. Esses fatores também são sintetizados em uma pontuação normalizada entre zero e cem e em uma justificativa textual. Como os valores mudam conforme as tarefas evoluem, a saúde é recalculada a cada consulta.

O desempenho dos membros é calculado a partir das tarefas atribuídas, concluídas e atrasadas. Essas informações permitem comparar produtividade e participação

no período. Os mesmos dados são utilizados para gerar relatórios em PDF, oferecendo um registro consolidado do estado do projeto e apoiando a identificação de dificuldades e a tomada de decisões.

3.4 ARQUITETURA

A arquitetura do sistema foi idealizada com base no princípio de separação em camadas, onde cada camada representa um ambiente único, possuindo características e regras próprias. Apesar de haver um certo nível de acoplamento, o intuito é que cada camada funcione de forma independente. Assim, a arquitetura foi organizada em três camadas: Camada de Visualização, Camada de Serviços e Camada de Dados. A Camada de Serviços, por sua vez, é subdividida em módulos, cada um responsável por um aspecto do sistema.

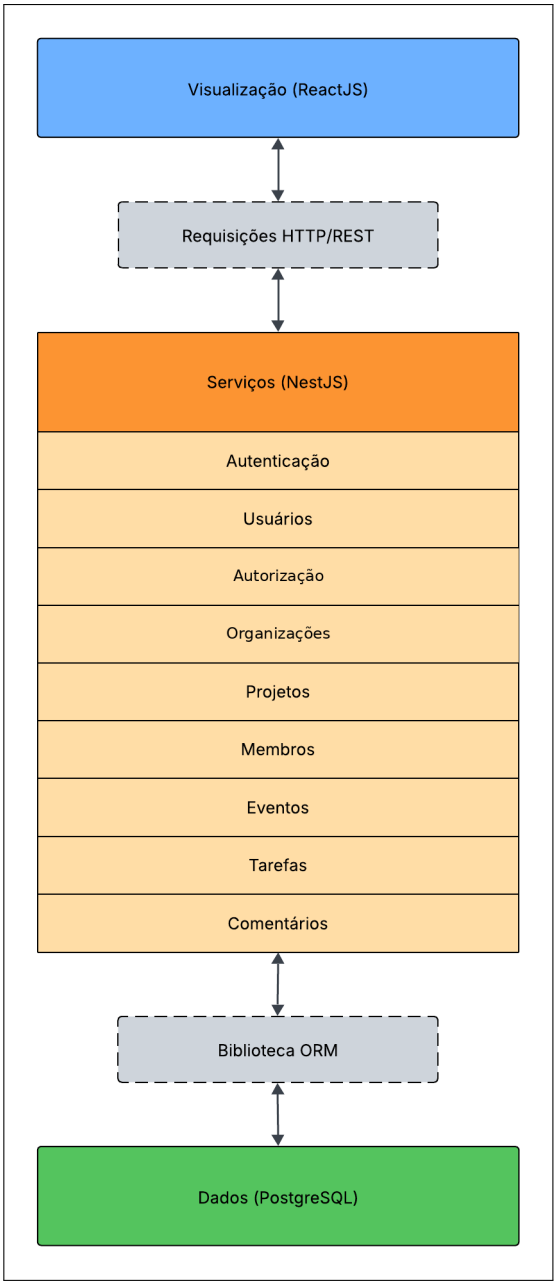
Camada de Visualização: Constitui o principal meio de interação entre o usuário e o sistema, sendo responsável pela representação visual das funcionalidades disponibilizadas na camada de Serviços.

Camada de Serviços: Responsável pelo processamento e tratamento de dados, bem como pela execução das principais funções do serviço. Está dividida em módulos, previamente descritos neste trabalho, sendo construída com base no modelo arquitetural REST.

Camada de Dados: Encarregada de armazenar os dados e principais informações do serviço, utilizando um banco de dados relacional, no qual os dados são estruturados em tabelas.

Para que o sistema funcione de forma harmoniosa, é preciso criar também um nível de comunicação entre cada camada, servindo como um *link* para troca de dados. A comunicação entre as camadas de Visualização e Serviço ocorre através de requisições HTTP baseadas no padrão REST. Já a integração entre as camadas de Serviços e Dados é realizada por meio de uma biblioteca ORM, que atua como intermediária na manipulação das entidades do banco de dados. A Figura 4 ilustra o diagrama geral da arquitetura do sistema.

Figura 4 – Exemplo visual da arquitetura



Fonte: Elaborado pelo autor

Graças a esta estrutura, foi possível desenvolver individualmente cada camada de forma que houvesse uma validação prévia antes da implementação da integração entre as camadas. Esta estrutura possibilitou a detecção antecipada de inconsistências, diminuiu o retrabalho e assegurou uma integração mais consistente, resultando em uma maior confiabilidade neste estágio. Além disso, a organização em camadas favoreceu a

manutenção evolutiva, possibilitando a modificação ou substituição de componentes específicos sem impactos significativos nas demais partes da aplicação.

3.4.1 Especificação dos principais módulos

Autenticação e Autorização: O módulo de Autenticação é responsável por validar as credenciais do usuário e identificar sua sessão por meio de um token JWT. Após essa identificação, o módulo de Autorização verifica se o usuário possui permissão para executar a ação solicitada, considerando seu papel na organização ativa, o recurso acessado e sua relação com a entidade envolvida. Essa separação garante que o usuário seja inicialmente identificado e, em seguida, tenha seu acesso limitado às funcionalidades compatíveis com suas responsabilidades.

Usuários e Contas: A camada de Contas é responsável pelas credenciais de acesso, armazenando o e-mail e a senha protegida, enquanto a camada de Usuários mantém os dados públicos do perfil, como nome e *username*. Cada usuário é associado a uma única conta por meio de sua chave identificadora. Dessa forma, a conta identifica e autentica o indivíduo, enquanto o usuário representa sua identidade dentro das funcionalidades do sistema. Operações de edição ou exclusão tratam as duas entidades em conjunto para manter a consistência dos dados.

Organizações e Afiliações: A camada de Organizações representa os espaços de trabalho nos quais os projetos e participantes são agrupados. A relação entre um usuário e uma organização é registrada por uma afiliação, que também define seu papel como Proprietário, Gestor ou Membro. Ao criar uma organização, o sistema associa automaticamente o usuário responsável como Proprietário. As afiliações permitem que um mesmo usuário participe de diferentes organizações e possua permissões distintas em cada uma delas.

Projetos e Estatísticas: A camada de Projetos organiza as atividades de uma organização, reunindo membros, tarefas, eventos, prazos e um possível gestor responsável. A camada de Estatísticas utiliza esses dados para consolidar informações sobre progresso, atrasos, produtividade, desempenho dos membros e saúde do projeto. Os indicadores são agrupados por período e também podem ser utilizados na geração de relatórios, oferecendo aos Gestores e Proprietários uma visão gerencial do andamento do trabalho.

3.5 TECNOLOGIAS UTILIZADAS

Conforme apresentado em tópicos anteriores, o sistema está dividido em camadas, sendo que cada uma abrange uma área do sistema. Para fins de organização,

essas camadas serão classificadas em dois grupos: Comunicação, responsáveis por realizar a integração e troca de dados entre os componentes do sistema. E camadas de Sistema, que representam o núcleo da aplicação e são responsáveis pelas principais funcionalidades do sistema.

Por desempenharem diferentes aspectos do sistema, é necessário escolher uma tecnologia adequada para cada camada, de maneira que suas funcionalidades sejam incorporadas de forma eficiente. Assim, garante-se um melhor desempenho e qualidade de código. Nesse contexto, esta seção apresenta as tecnologias adotadas na construção da aplicação proposta neste trabalho.

3.5.1 Camada de Sistema

1. **Typescript**

   Foi escolhida como linguagem principal e está presente em todas as camadas do sistema. Desenvolvida pela Microsoft, essa linguagem estende o JavaScript com um sistema de verificação estática de tipos, além de oferecer recursos como interfaces, classes e diferentes mecanismos para definição e composição de tipos (Microsoft, 2026).

   Por ser descendente do JavaScript, esta tecnologia possui uma integração quase perfeita em navegadores, uma característica relevante na construção de um aplicativo web. Justamente a característica que levou a escolha desta tecnologia. Além disso, possui um vasto ecossistema de bibliotecas e *frameworks* com alta compatibilidade com navegadores web.

2. **NestJS**

   O NestJS foi adotado como principal *framework* para o desenvolvimento da API do sistema. Baseado em Node.js, com suporte completo ao TypeScript, o NestJS combina conceitos de programação orientada a objetos, programação funcional e programação funcional reativa, oferecendo uma estrutura organizada para a criação de aplicações no *backend* (NestJS, 2026).

   A escolha desta tecnologia se deu por sua estrutura robusta fortemente baseada na arquitetura modular e orientação a componentes, características importantes no desenvolvimento incremental. A divisão da aplicação em módulos independentes possibilita a implementação, validação e evolução de funcionalidades de forma isolada, reduzindo impactos entre diferentes partes do sistema. Essa abordagem facilita tanto a manutenção quanto a expansão da API ao longo do desenvolvimento. Além disso, possui compatibilidade com várias bibliotecas com-

plementares, como *ORMs* e bibliotecas para autenticação JWT, que ajudam a ter um sistema mais seguro e consistente.

3. **ReactJS**

   O ReactJS foi utilizado como a principal biblioteca para o desenvolvimento visual do sistema. Mantido pela Meta, o React permite estruturar interfaces por meio de componentes que reúnem sua própria lógica e apresentação, podendo ser combinados e reutilizados em diferentes partes da aplicação (Meta Platforms, 2026).

   A principal vantagem que levou à escolha do ReactJS foi sua praticidade. Graças à sintaxe JSX, que incorpora elementos HTML dentro do JavaScript, essa combinação torna o desenvolvimento mais eficaz, já que todo o código de um componente estará em um único arquivo. Sua forte vinculação entre HTML e código JavaScript permite que cada componente possua suas próprias chamadas à API, com as requisições sendo efetuadas somente caso o componente esteja carregado em tela, melhorando assim o desempenho. Além disso, o modelo baseado em componentes favorece a reutilização de código e a padronização da interface, reduzindo redundâncias e facilitando a manutenção.

4. **PostgreSQL**

   O PostgreSQL foi adotado como sistema de gerenciamento de banco de dados, sendo responsável pela construção da camada de Dados. Trata-se de um SGBD objeto-relacional de código aberto que utiliza e estende a linguagem SQL, destacando-se por sua confiabilidade, integridade de dados, extensibilidade e capacidade de lidar com cargas de trabalho complexas (PostgreSQL Global Development Group, 2026).

   A escolha do PostgreSQL neste projeto está relacionada à sua estabilidade e ao suporte eficiente a aplicações de médio e grande porte. Sua compatibilidade com ferramentas ORM, como o Prisma, facilitou a integração com a Camada de Serviços, permitindo uma manipulação estruturada e tipada dos dados. Além disso, o desempenho aliado à confiabilidade do PostgreSQL contribuiu para a construção de um sistema consistente, escalável e alinhado aos requisitos definidos, favorecendo a manutenção e evolução incremental da aplicação.

3.5.2 Camada de Comunicação

1. **Prisma**

   O Prisma foi adotado como a ferramenta de mapeamento objeto-relacional (ORM) utilizada na construção da camada de Dados do sistema, auxiliando na comu-

nicação entre a API (*backend*) e o banco de dados. Trata-se de um *ORM* para Node.js e TypeScript que fornece acesso tipado ao banco de dados, recursos de migração e um cliente de consultas gerado a partir de um esquema centralizado (Prisma Data, 2026).

As principais características que levaram à escolha do Prisma foram sua simplicidade e compatibilidade com vários bancos de dados. Graças à tecnologia de esquema genérico, é possível criar um modelo de dados com tipos genéricos com suporte para quaisquer bancos presentes no Prisma. Esta característica facilita a migração entre diversos bancos de dados, um aspecto a ser observado caso seja necessária uma migração à medida que os dados se ampliam.

2. **Axios**

   O Axios foi utilizado como a principal biblioteca para a realização de requisições HTTP entre a camada de Visualização e a camada de Serviços do sistema. Trata-se de um cliente HTTP baseado em *Promises*, compatível com navegadores e Node.js, que fornece métodos para o envio de requisições e recursos para configurar cabeçalhos, interceptar requisições e respostas e centralizar o tratamento de erros (Axios, 2026).

3. **JWT**

   O JSON Web Token (JWT) foi utilizado como o mecanismo principal de autenticação e autorização do sistema. Definido pela RFC 7519, o JWT é um formato compacto e seguro para URLs, utilizado para representar declarações transmitidas entre duas partes. Essas declarações são codificadas em um objeto JSON e podem ser assinadas digitalmente ou protegidas por um código de autenticação, permitindo verificar sua integridade (JONES et al., 2015).

3.6 TELAS DO SISTEMA

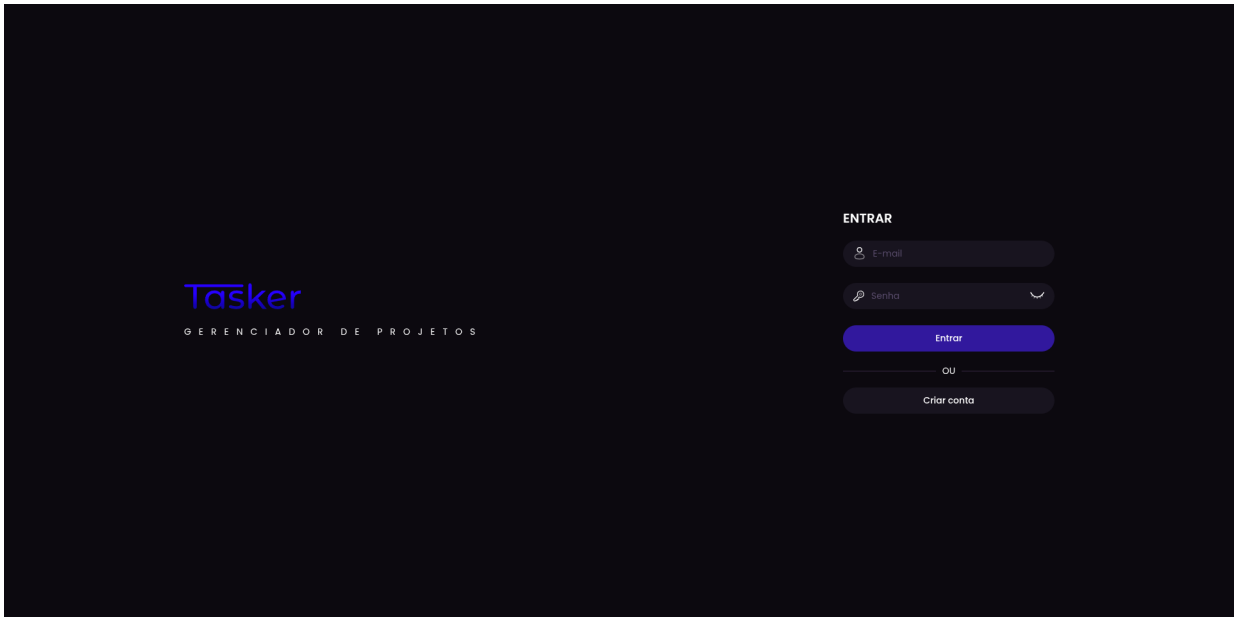
Nesta seção será apresentado as principais telas do sistema. As telas ilustram a organização da interface e a disposição dos elementos visuais que permitem a interação do usuário com a aplicação, considerando a etapa final do projeto.

As telas a serem apresentadas representam a concretização dos requisitos funcionais implementados para a primeira versão do sistema, com o objetivo de demonstrar visualmente o fluxo de navegação e auxiliar na compreensão do funcionamento dos requisitos implementados. Por fins de organização, as telas serão apresentadas através dos principais fluxos de interação com o sistema.

3.6.1 Cadastro e Login

A Figura 5 representa a tela de abertura, sendo a primeira interação do usuário com o sistema. A interface foi organizada para que o usuário tenha uma apresentação intuitiva do sistema, já tendo uma noção de como será a identidade visual do aplicativo, incluindo cores, formas e tipos de letra. Com a disposição bem estruturada dos componentes, a tela foi projetada para transmitir equilíbrio, com o logotipo do aplicativo posicionado à esquerda e os espaços para interação localizados à direita.

Figura 5 – Tela de Login

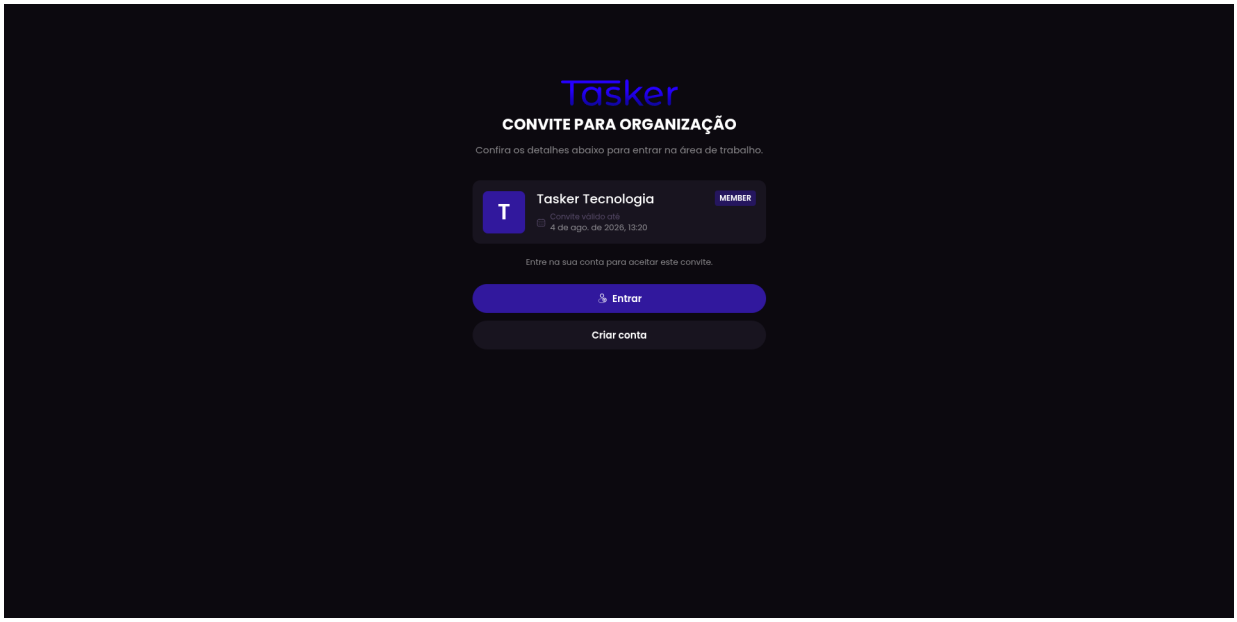


Fonte: Elaborado pelo autor

A Figura 6 apresenta a tela de convite para ingresso em uma organização. Na região central, a interface exibe um cartão contendo a identificação visual e o nome da organização, o prazo de validade do convite e o cargo de Membro que será atribuído ao usuário após o aceite. Essas informações permitem que o convidado confirme a organização de destino e as condições de seu ingresso antes de prosseguir.

Como o aceite exige autenticação, a parte inferior da tela disponibiliza o botão *Entrar*, destinado aos usuários que já possuem uma conta, e a opção *Criar conta*, voltada aos novos usuários. Após a autenticação ou o cadastro, o usuário poderá retornar ao fluxo do convite para concluir sua entrada na organização.

Figura 6 – Tela de convite

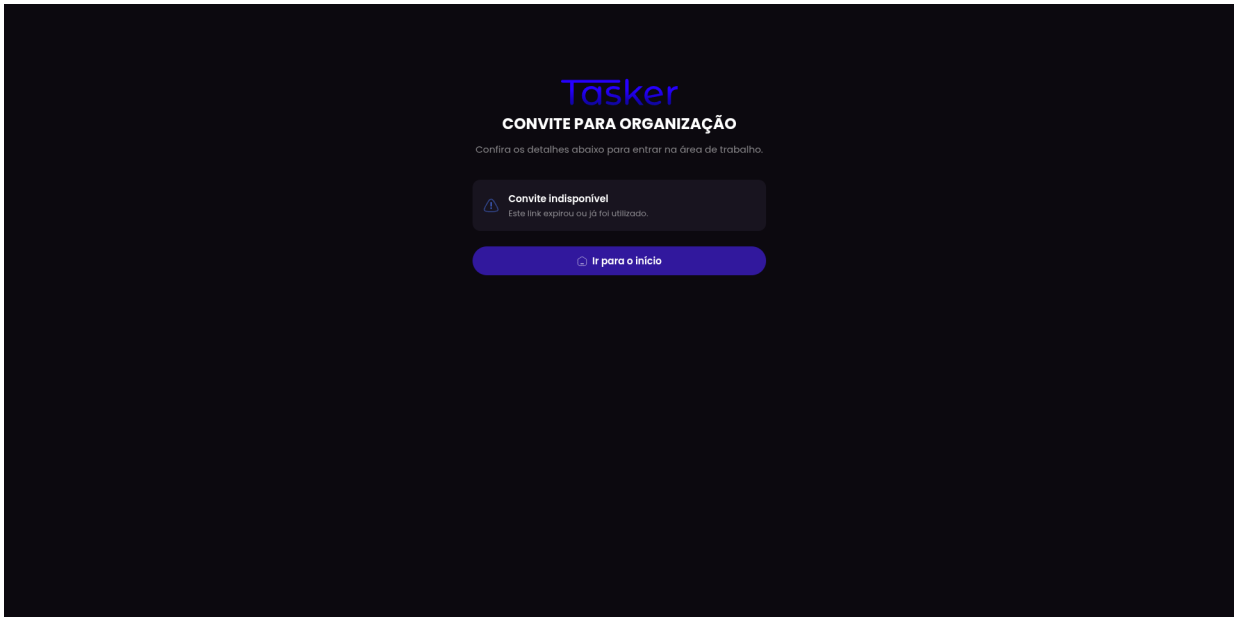


Fonte: Elaborado pelo autor

A Figura 7 apresenta a tela exibida quando o convite não está mais disponível. A interface mantém a identidade visual da página de ingresso, mas substitui os dados da organização por uma mensagem de alerta, informando que o endereço expirou ou já foi utilizado. Dessa forma, o sistema impede a continuidade do fluxo por meio de um convite inválido e comunica sua situação ao usuário de maneira direta.

Abaixo da mensagem encontra-se o botão *Ir para o início*, que permite abandonar o fluxo do convite e retornar à página inicial da aplicação. Essa alternativa orienta a navegação do usuário e evita que ele permaneça em uma tela sem ações válidas disponíveis.

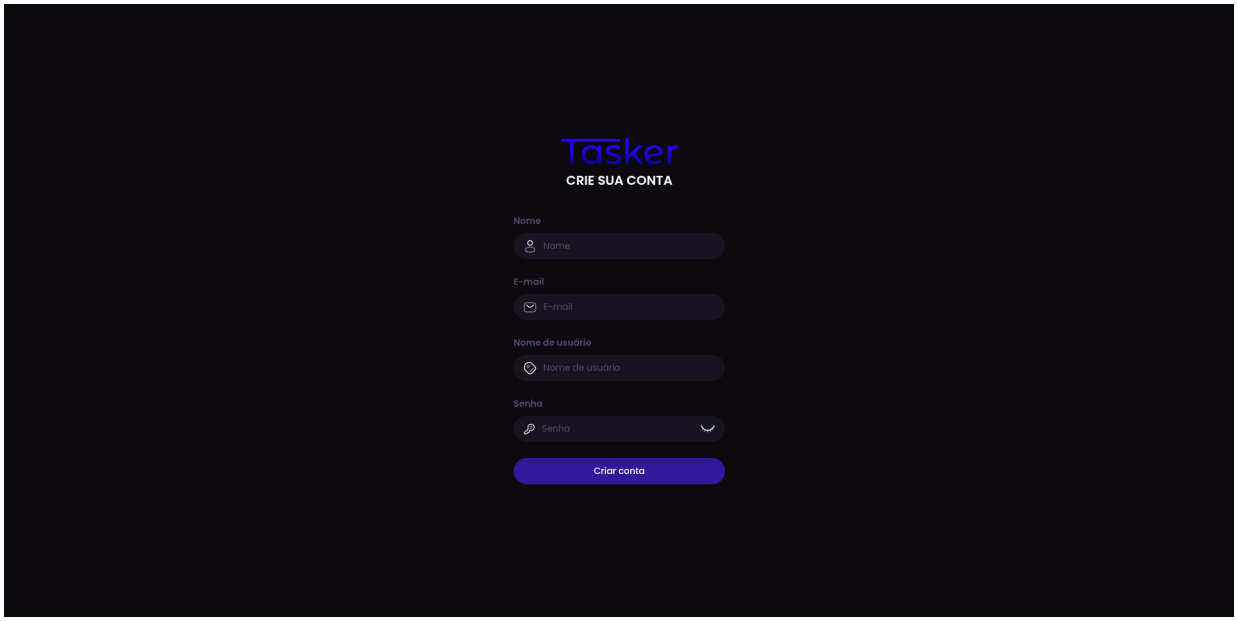
Figura 7 – Tela de convite inválido



Fonte: Elaborado pelo autor

O fluxo de criação de conta é apresentado na Figura 8, por meio de um formulário para preenchimento das informações da conta. Ele inclui os campos *username* (nome de usuário), nome, *e-mail* e senha. Cada campo foi organizado para ter um ícone que simbolize o campo, permitindo que o usuário entenda seu propósito de maneira rápida, sem a necessidade de leitura. No final do formulário, o botão de criação de conta se destaca, adotando a cor azul como predominante para eventos de confirmação no sistema.

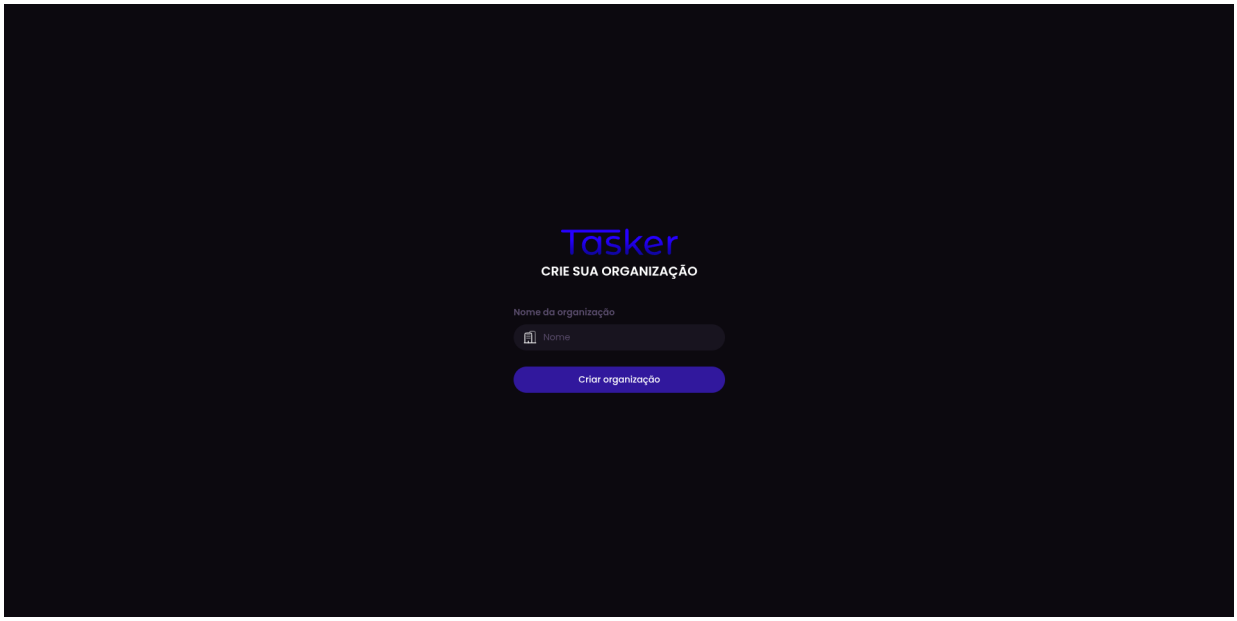
Figura 8 – Tela para criação de contas



Fonte: Elaborado pelo autor

Caso o usuário não possua ou não participe de uma organização, a Figura 9 apresenta a próxima tela do fluxo. Nessa interface, é solicitado apenas o nome da organização que será cadastrada. O formulário é composto por um campo de texto acompanhado por um ícone representativo e por um botão responsável por concluir a operação. Após a confirmação, a organização é criada e associada à conta do usuário responsável pelo cadastro. Esta tela possui uma relação de redirecionamento com a tela *Escolha sua área de trabalho*, Figura 10, que será explicada no próximo ponto.

Figura 9 – Tela de criação de organização



Fonte: Elaborado pelo autor

3.6.2 Entrada no sistema

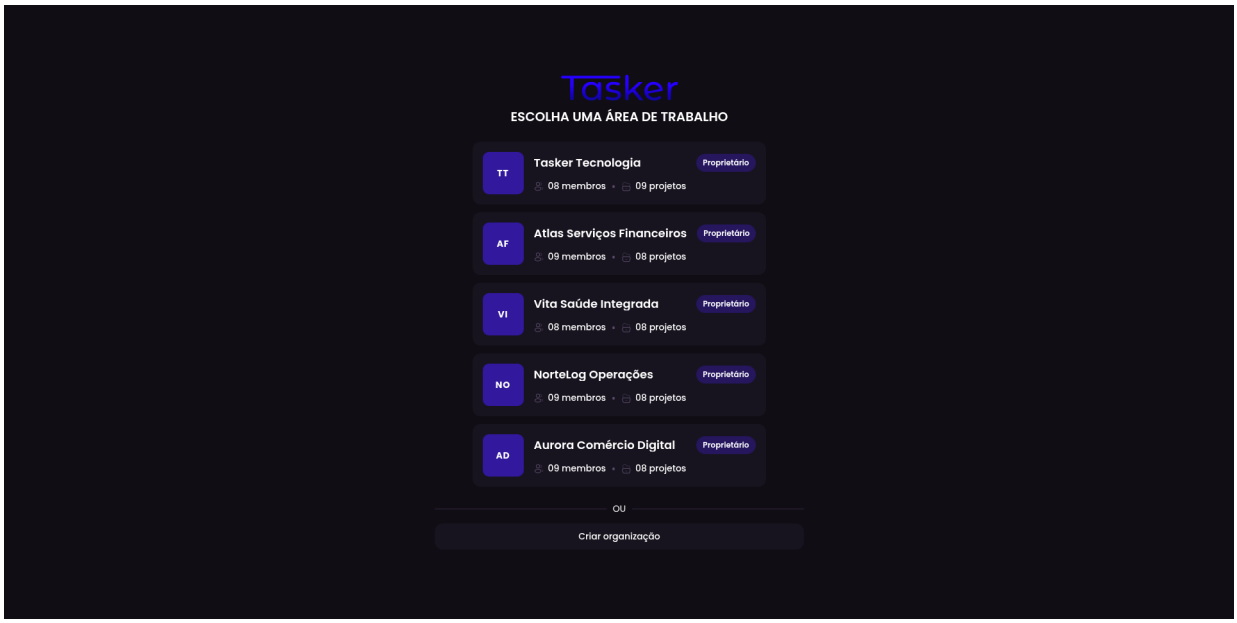
O fluxo apresentado nas Figuras 10, 11, 12 e 13 demonstra a entrada no sistema. Após a autenticação, o usuário é direcionado para a tela de seleção de *workspace*. No contexto da aplicação, um *workspace* representa uma organização da qual o usuário participa, podendo assumir diferentes níveis de permissão, como proprietário, gestor ou membro. A partir da organização selecionada, o sistema apresenta uma interface personalizada de acordo com o papel desempenhado pelo usuário.

Como apresentado anteriormente, a Figura 10 apresenta a tela de seleção da área de trabalho. Em casos onde o usuário não possua organização ou vínculo com alguma organização, o sistema direciona-o para a tela *Crie uma Organização*, representada na figura 9. A interface mantém a identidade visual adotada pelo sistema, com o logotipo centralizado na parte superior, seguido pelo título *Escolha uma área de trabalho*, responsável por orientar o usuário na escolha da organização que deseja acessar.

A região central da tela é composta por uma lista de cartões representando as organizações vinculadas ao usuário. Cada cartão exibe informações resumidas da organização, incluindo sua identificação visual, nome, quantidade de membros e quantidade de projetos cadastrados. Adicionalmente, é exibido um indicador do papel exercido pelo usuário naquela organização, permitindo a identificação rápida de suas responsabilidades e permissões.

Ao selecionar a área de trabalho, ou organização, o usuário será redirecionado para a tela inicial, onde se adaptará de acordo com seu papel na organização selecionada.

Figura 10 – Tela de escolha de organização

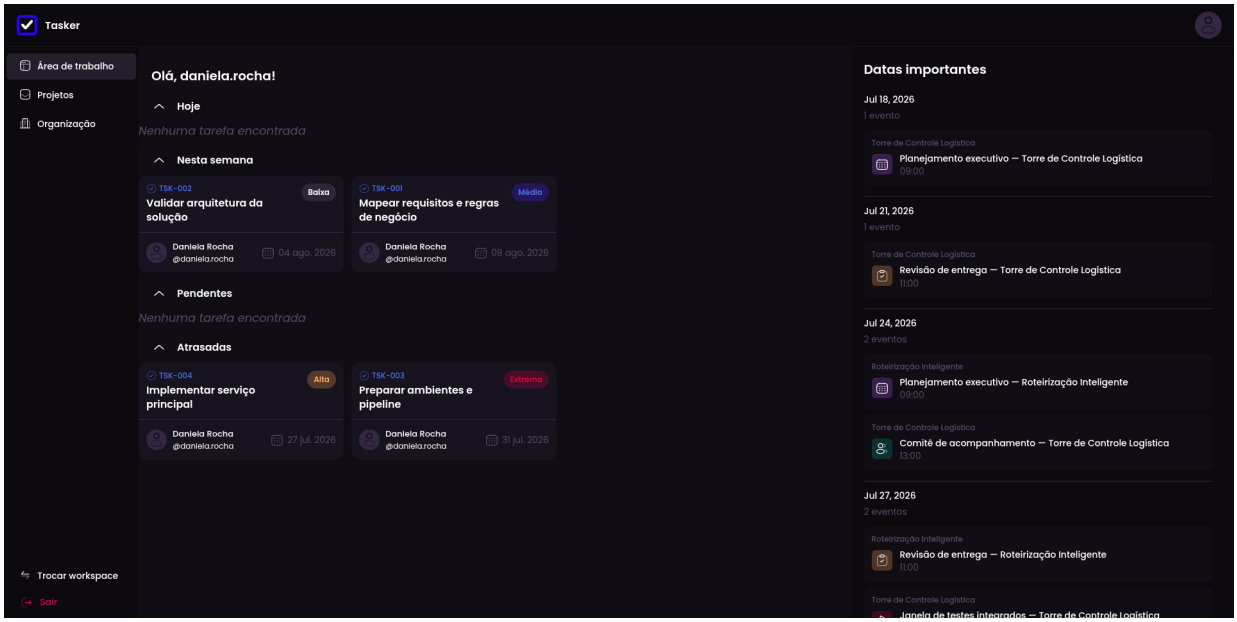


Fonte: Elaborado pelo autor

A Figura 11 apresenta o painel inicial destinado aos usuários com perfil de membro (Member). Essa interface possui foco na visualização das atividades atribuídas ao próprio usuário, simplificando o acesso às informações necessárias para a execução das tarefas. Na área principal, as tarefas são exibidas por categorias temporais e situacionais, como atividades do dia, tarefas previstas para a semana, tarefas pendentes e tarefas atrasadas. Cada atividade é apresentada por meio de cartões contendo informações resumidas, como título, prioridade, responsável e data prevista para conclusão. Essa organização facilita a identificação das tarefas que demandam atenção imediata.

Na lateral direita permanece disponível o painel de datas importantes, permitindo que o usuário acompanhe compromissos e prazos relevantes relacionados às suas atividades. O objetivo desta interface é fornecer uma visão clara e objetiva das responsabilidades atribuídas ao membro, favorecendo o acompanhamento das demandas e a organização do trabalho diário.

Figura 11 – Painel de Membro

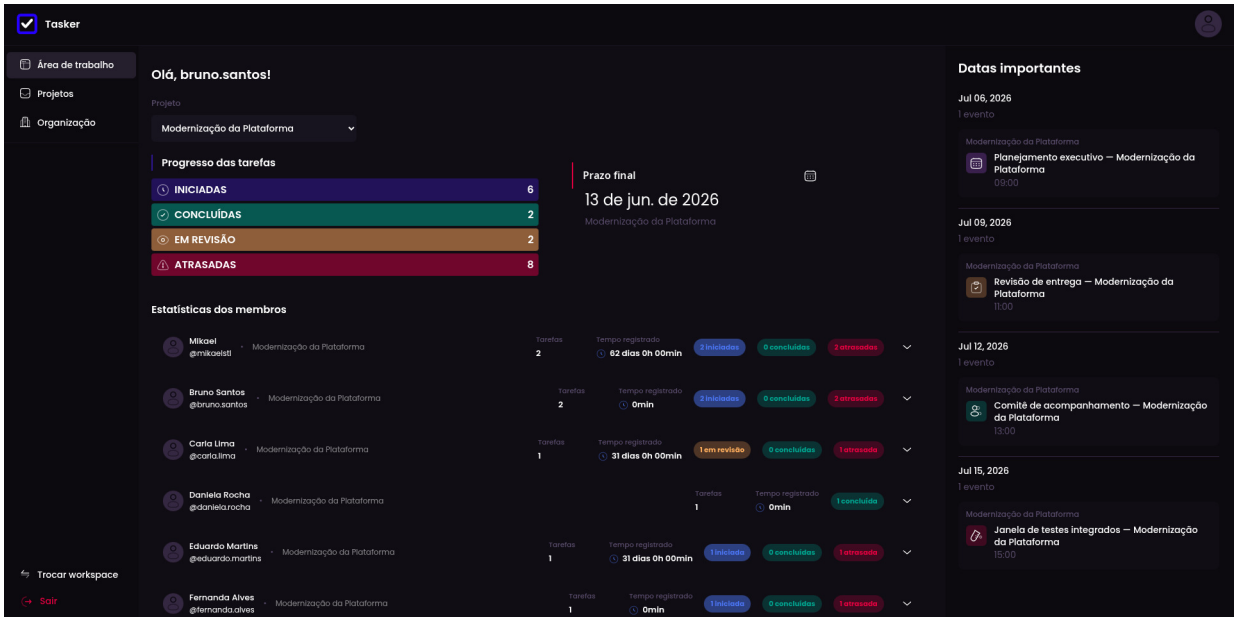


Fonte: Elaborado pelo autor

A Figura 12 apresenta a interface disponibilizada para usuários com perfil de gestor (Manager). Dentro do sistema, o gestor possui foco no acompanhamento operacional dos projetos sob sua responsabilidade e no gerenciamento das equipes envolvidas. Por isso, a área central apresenta indicadores relacionados ao progresso das tarefas de um projeto indicado pelo usuário, organizados em categorias como iniciadas, concluídas, atrasadas e em revisão. Também é disponibilizada uma seleção de projetos, permitindo que o gestor visualize métricas específicas de um determinado contexto de trabalho.

Abaixo dos indicadores, são exibidas informações referentes aos membros participantes dos projetos, acompanhadas de métricas individuais de desempenho e progresso das atividades atribuídas. Na lateral direita encontra-se um painel de datas importantes, destinado ao acompanhamento de entregas, eventos e prazos relevantes para os projetos em execução. Esta interface concentra informações necessárias para o planejamento, supervisão e controle das atividades da equipe, proporcionando uma visão detalhada do andamento dos trabalhos.

Figura 12 – Painel de Gestor

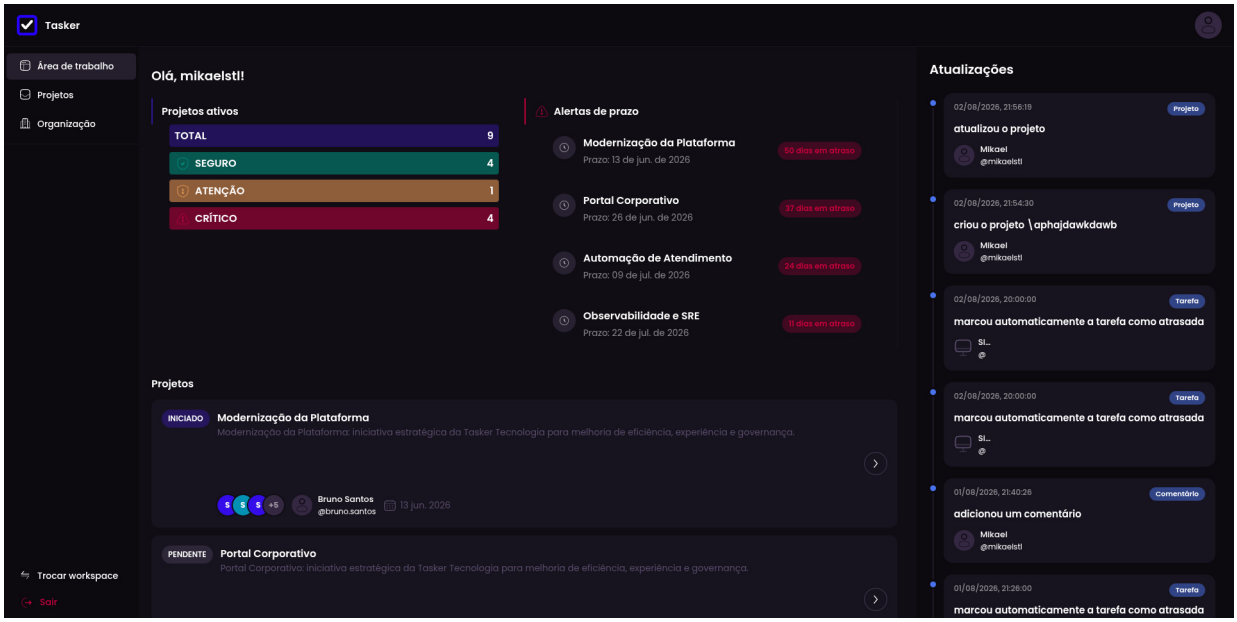


Fonte: Elaborado pelo autor

A Figura 13 apresenta o painel inicial disponibilizado para usuários com perfil de proprietário (Owner) da organização. Esse perfil possui acesso às funcionalidades administrativas e aos indicadores globais relacionados aos projetos e membros da organização. Na parte central, são exibidos indicadores estratégicos da entidade, como estatísticas de projetos em andamento, avisos de prazos e atalhos para funcionalidades comumente usadas. Também é exibida uma listagem resumida dos projetos em andamento, contendo informações relevantes para acompanhamento.

À direita encontra-se um painel de atualizações recentes, destinado à exibição de notificações e atividades realizadas pelos membros da organização. Essa disposição permite que o proprietário acompanhe simultaneamente o desempenho dos projetos e as interações ocorridas no ambiente organizacional. O painel foi projetado para fornecer uma visão consolidada da organização, auxiliando na tomada de decisões e no monitoramento das atividades desenvolvidas.

Figura 13 – Painel de Proprietário

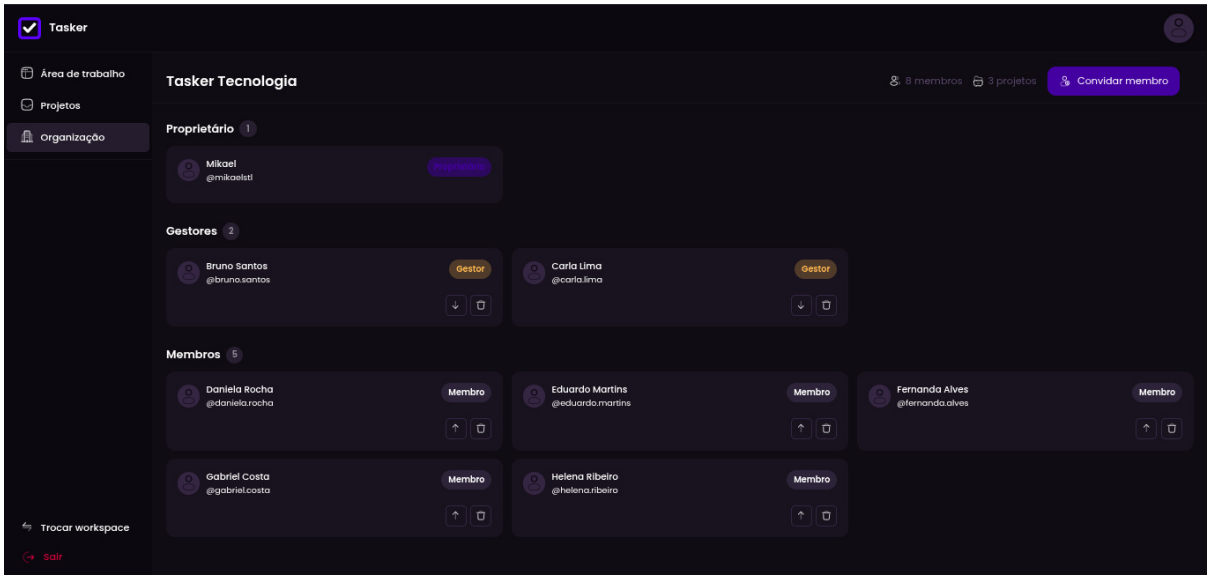


Fonte: Elaborado pelo autor

A Figura 14 apresenta a tela de detalhes da organização, utilizada para consultar e administrar seus participantes. No cabeçalho são exibidos o nome da organização, a quantidade de membros e o total de projetos associados. O botão *Convidar membro* permite iniciar o fluxo de geração e compartilhamento de um convite para ingresso na organização.

Os participantes são agrupados conforme os papéis de Proprietário, Gestor e Membro, permitindo identificar a hierarquia de acesso de forma direta. Cada cartão apresenta a identificação visual, o nome, o nome de usuário e o cargo da pessoa. Para usuários autorizados, também são disponibilizadas ações de promoção, rebaixamento e remoção da afiliação, representadas por botões posicionados no próprio cartão. Essa organização centraliza o gerenciamento da equipe e facilita a aplicação das regras de acesso da organização.

Figura 14 – Detalhes da organização



Fonte: Elaborado pelo autor

3.6.3 Módulo de Gerenciamento de Projetos

O módulo de gerenciamento de projetos concentra as funcionalidades relacionadas ao planejamento, acompanhamento e execução das atividades desenvolvidas pelas equipes. As interfaces disponibilizadas permitem o monitoramento do progresso dos projetos, administração de membros, organização de tarefas e acompanhamento de prazos, oferecendo recursos específicos conforme o perfil de acesso do usuário.

A Figura 15 apresenta a tela de Visão Geral do projeto, responsável por reunir as informações necessárias para seu acompanhamento cotidiano. No cabeçalho são exibidos o nome do projeto, seu estado atual, a data de início e o prazo de entrega. Em seguida, a descrição apresenta resumidamente os objetivos e o escopo do trabalho. O botão *Editar*, disponibilizado conforme as permissões do usuário, permite alterar os dados cadastrais do projeto.

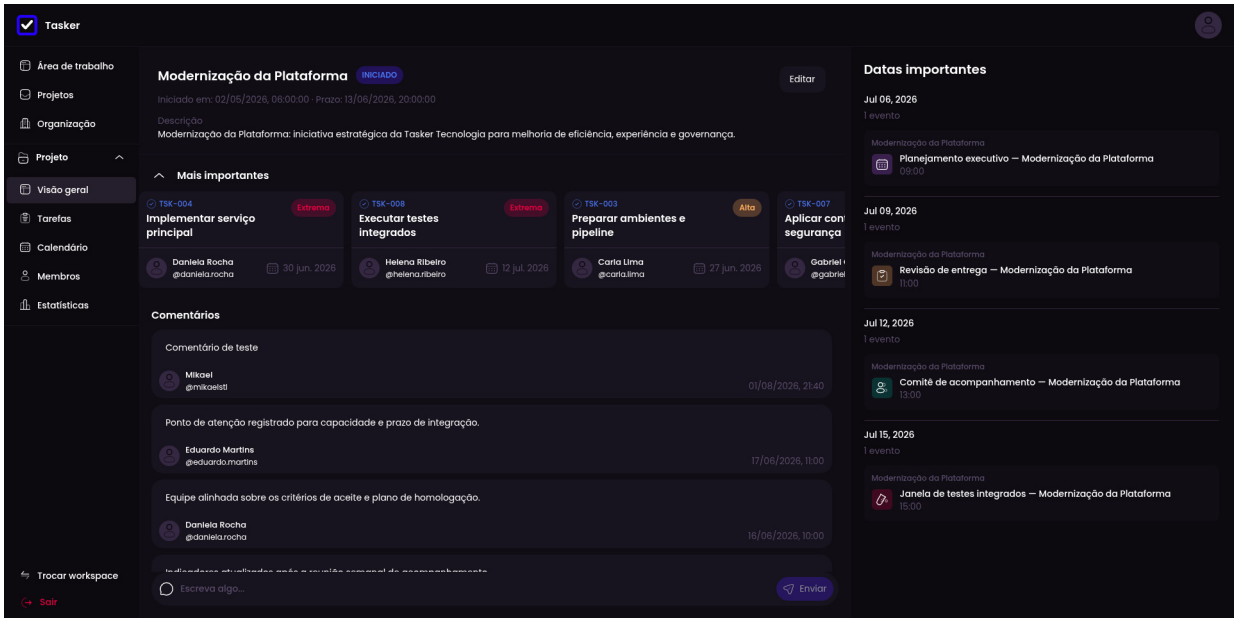
Na seção *Mais importantes*, as tarefas prioritárias são organizadas horizontalmente em cartões. Cada cartão apresenta o código e o título da tarefa, o nível de prioridade, o membro responsável e a data prevista para conclusão. Essa disposição permite comparar rapidamente as atividades críticas e reconhecer aquelas que exigem acompanhamento imediato. A seção pode ser recolhida para ampliar o espaço destinado aos demais conteúdos da página.

A área de comentários concentra as mensagens registradas pelos participantes do projeto. Cada comentário contém o conteúdo publicado, a identificação do autor, seu

nome de usuário e a data de envio. Na parte inferior, um campo de texto acompanhado pelo botão *Enviar* possibilita registrar uma nova mensagem, mantendo a comunicação vinculada ao contexto do projeto.

Na lateral direita, o painel *Datas importantes* apresenta os compromissos em ordem cronológica e agrupados por dia. Para cada data, a interface informa a quantidade de eventos e exibe cartões contendo sua categoria, título e horário. Essa área reúne reuniões, revisões, entregas e outros marcos relevantes, permitindo consultar os próximos compromissos sem sair da visão geral.

Figura 15 – Visão Geral do Projeto

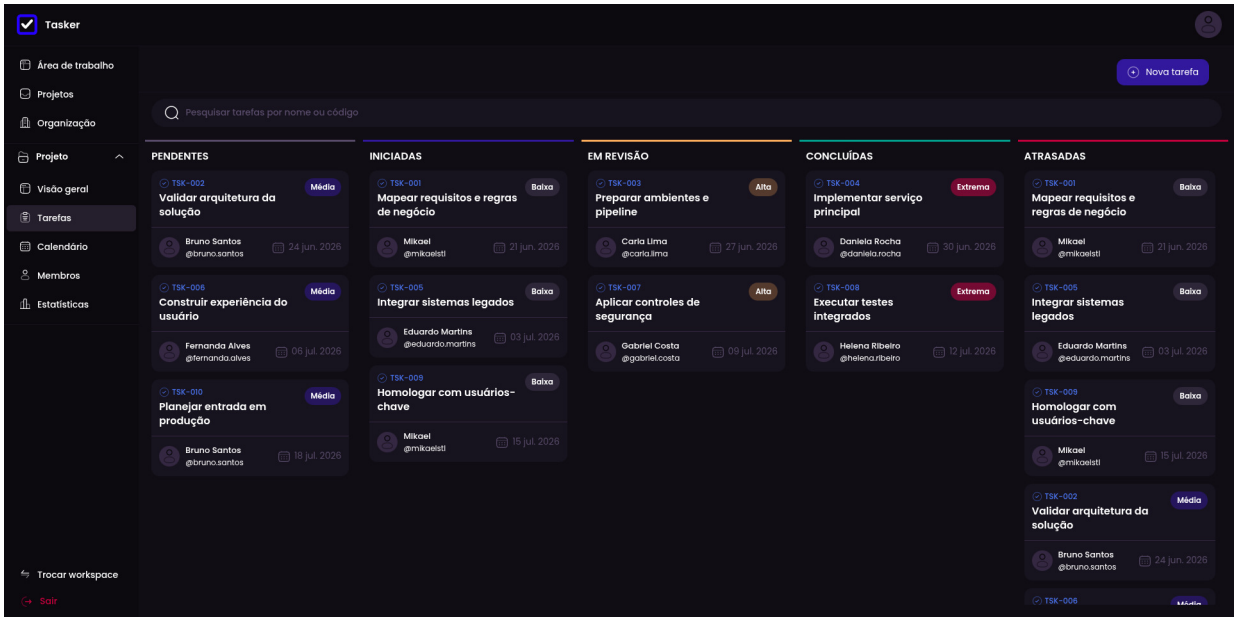


Fonte: Elaborado pelo autor

A Figura 16 apresenta a interface de gerenciamento de tarefas do projeto. Essa funcionalidade utiliza uma abordagem visual baseada em quadros organizados por status, permitindo acompanhar o ciclo de vida das atividades. Na parte superior encontra-se uma barra de pesquisa acompanhada por ferramentas de filtragem e ordenação. Essas funcionalidades facilitam a localização de tarefas específicas e a organização das informações conforme diferentes critérios.

As tarefas são distribuídas em colunas que representam seus estados de execução, como pendente, iniciada, em revisão, concluída e atrasada. Cada tarefa é apresentada por meio de um cartão contendo informações resumidas, incluindo título, prioridade, responsável e prazo de entrega. A organização visual das tarefas permite acompanhar facilmente o progresso do trabalho e identificar atividades que necessitam de atenção imediata.

Figura 16 – Quadro de Tarefas

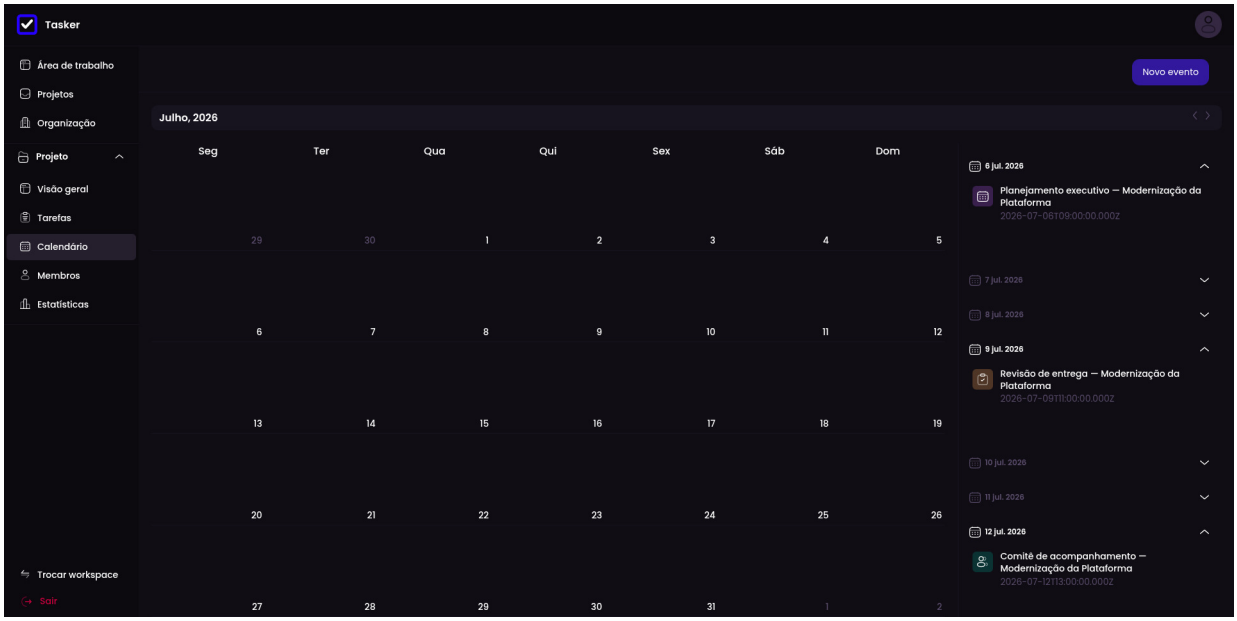


Fonte: Elaborado pelo autor

A Figura 17 apresenta a funcionalidade de calendário do projeto, destinada ao acompanhamento visual de tarefas, eventos e prazos. A interface é composta por um calendário mensal organizado em formato matricial, permitindo visualizar a distribuição das atividades ao longo dos dias do mês. Cada célula do calendário pode apresentar indicadores referentes à quantidade de tarefas e eventos programados para uma determinada data.

Na lateral direita encontra-se uma área complementar que exibe os compromissos associados ao dia selecionado, permitindo uma visualização detalhada das atividades previstas. Na parte superior da interface está disponível o botão *Novo Evento*, utilizado para registrar novos eventos relacionados ao projeto, tais como reuniões, entregas, apresentações ou marcos importantes. Essa funcionalidade auxilia na organização temporal das atividades, permitindo um melhor gerenciamento dos cronogramas e prazos estabelecidos.

Figura 17 – Calendário

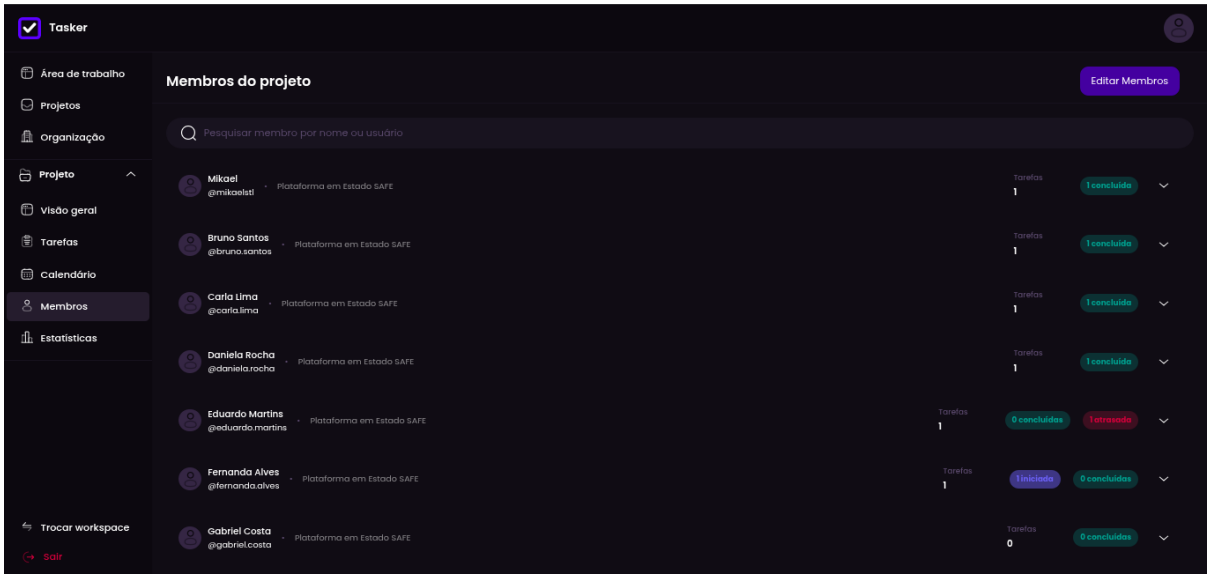


Fonte: Elaborado pelo autor

A Figura 18 apresenta a tela de membros disponibilizada aos usuários com perfil de Proprietário ou Gestor. A interface possui um campo de pesquisa para localizar participantes por nome ou nome de usuário e o botão *Editar Membros*, que permite alterar a composição da equipe conforme as permissões do usuário autenticado.

Além da identificação de cada participante, a visão administrativa apresenta a quantidade de tarefas associadas e indicadores coloridos que resumem as atividades iniciadas, concluídas ou atrasadas. A seta posicionada ao final de cada registro permite expandir suas informações. Dessa forma, Proprietários e Gestores podem consultar a equipe e acompanhar resumidamente a situação das tarefas de seus integrantes.

Figura 18 – Tela de Membros do projeto (Proprietário e Gestor)

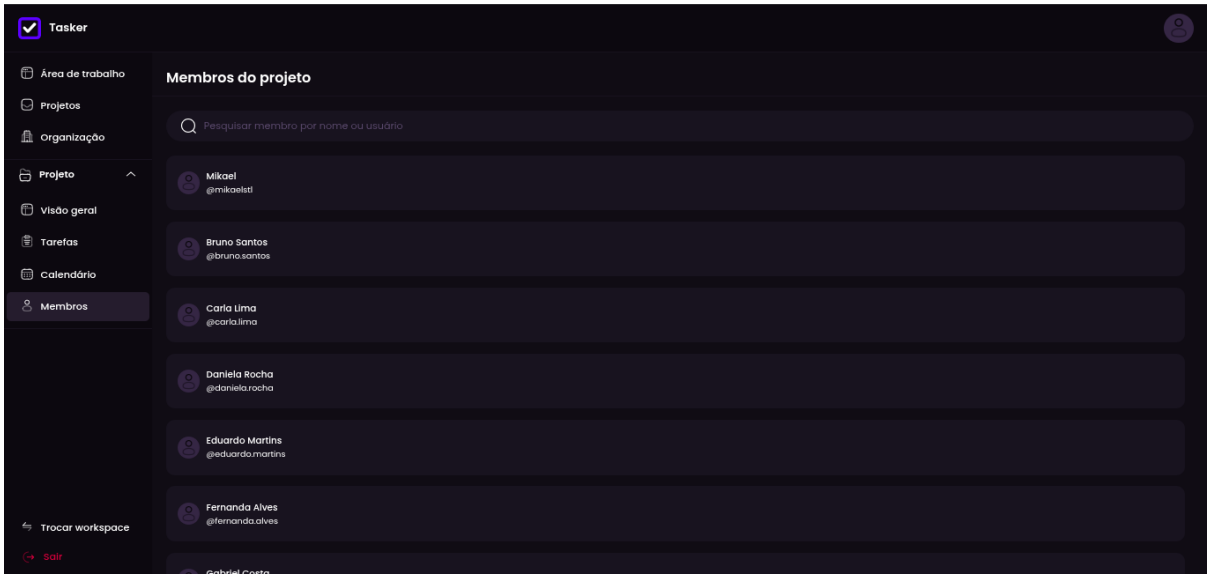


Fonte: Elaborado pelo autor

A Figura 19 apresenta a mesma área sob a perspectiva de um usuário com perfil de Membro. Essa visualização mantém o campo de pesquisa e a relação dos participantes do projeto, exibindo somente a identificação visual, o nome e o nome de usuário de cada integrante.

Por não possuir permissão para administrar a equipe ou consultar os indicadores individuais dos demais participantes, o Membro não visualiza o botão *Editar Membros*, os resumos de tarefas nem as ações de expansão presentes na visão administrativa. Essa adaptação reduz a quantidade de informações exibidas e mantém a interface compatível com as restrições definidas para esse perfil.

Figura 19 – Tela de Membros do projeto (Membros)

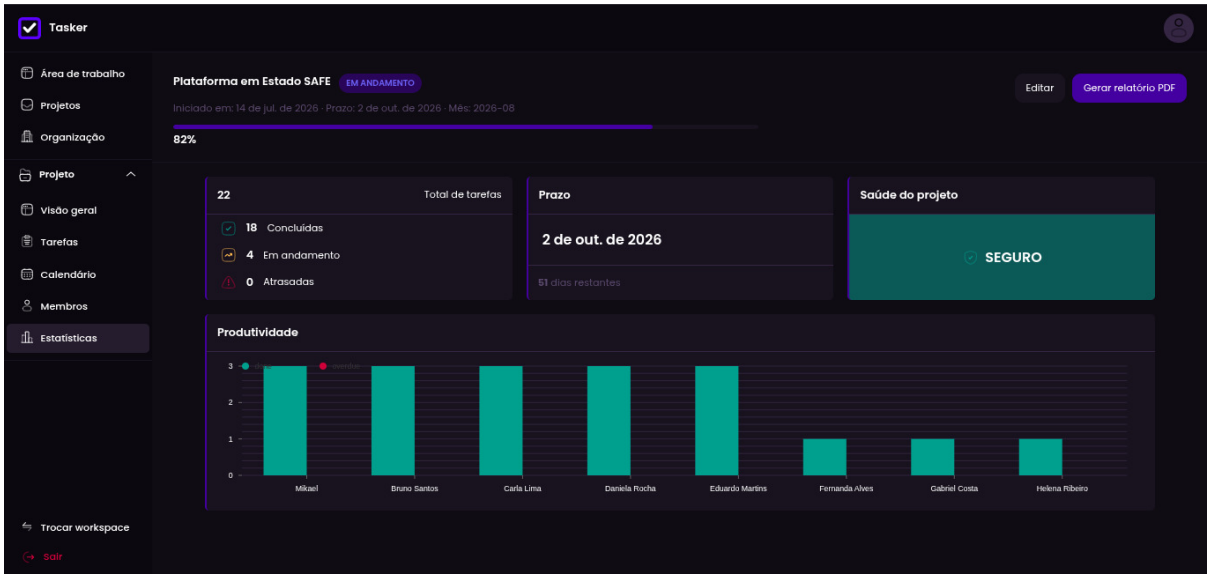


Fonte: Elaborado pelo autor

A Figura 20 apresenta a versão atualizada do painel estatístico do projeto. Na região superior são apresentados o nome e o estado atual do projeto, suas datas de início e entrega e uma barra com o percentual de progresso. Também estão disponíveis os botões de edição e geração do relatório em PDF, conforme as permissões do usuário.

Na área central, três cartões resumem os principais indicadores: a distribuição das tarefas entre concluídas, em andamento e atrasadas; o prazo final, acompanhado da quantidade de dias restantes; e a Saúde do Projeto, classificada visualmente como segura, em atenção ou crítica. Na parte inferior, o gráfico de produtividade compara, por participante, as tarefas concluídas e atrasadas. O painel concentra, assim, as informações necessárias para acompanhar o andamento do projeto e identificar situações que exigem atenção.

Figura 20 – Estatísticas do Projeto



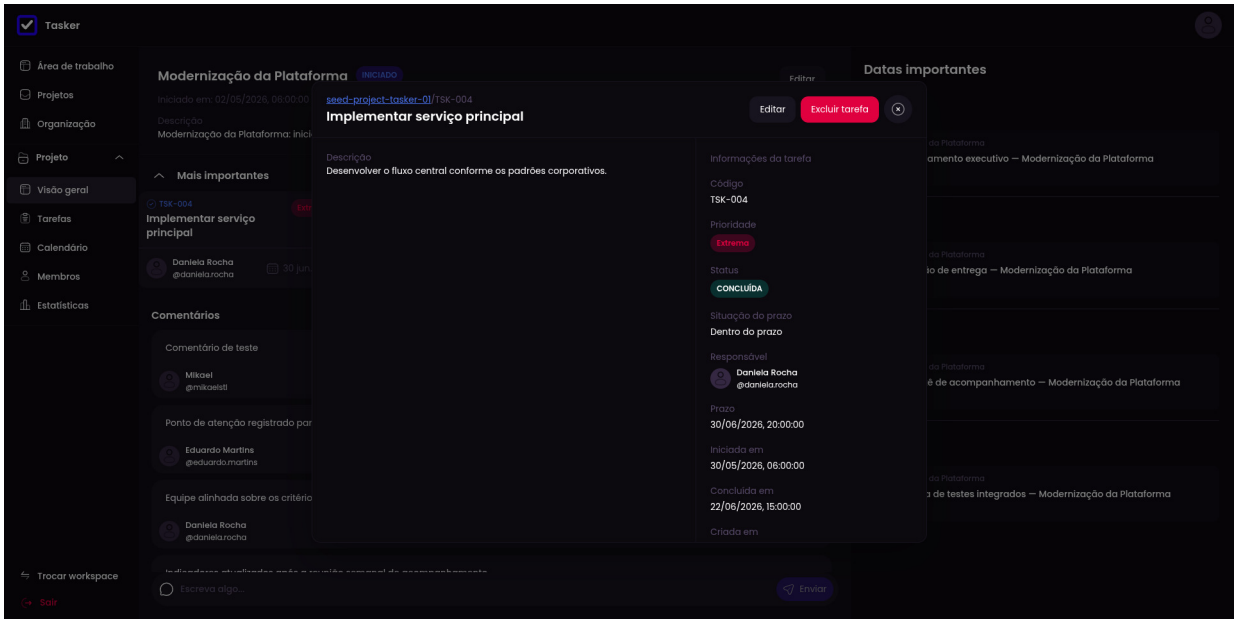
Fonte: Elaborado pelo autor

A Figura 21 apresenta a interface de detalhes de uma tarefa específica, acessada a partir do painel do projeto. O modal adota um layout dividido em três setores principais: o cabeçalho, o conteúdo centralizado com a descrição da tarefa e um painel lateral com mais detalhes e informações sobre o projeto.

A cabeçalho exibe o mapeamento estrutural indicando o projeto e o código da tarefa, seguido pelo título principal em destaque. Ao lado, exibe botões de navegação para edição e remoção da tarefa, e fechamaneto do modal. Abaixo do cabeçalho, exitem duas seções, uma aesquerda contendo a descrição da tarefa, enquanto a direita é reservada para os metadados da tarefa.

O painel lateral, denominado *Informações da Tarefa*, reúne os metadados associados à atividade, incluindo seu código identificador, nível de prioridade, *status*, situação em relação ao prazo, usuário responsável, prazo de entrega e datas de início, conclusão e criação. A apresentação dessas informações permite que o usuário acompanhe rapidamente o estado atual da tarefa, identifique seu responsável e verifique os principais marcos temporais de sua execução.

Figura 21 – Detalhes da tarefa



Fonte: Elaborado pelo autor

4 CONSIDERAÇÕES FINAIS

O desenvolvimento do Tasker representou uma experiência de grande relevância para a consolidação dos conhecimentos adquiridos ao longo do curso de Análise e Desenvolvimento de Sistemas. Mais do que a construção de uma ferramenta para gestão de equipes e projetos, este trabalho proporcionou uma oportunidade prática de aplicar conceitos de engenharia de software, arquitetura de sistemas, modelagem de requisitos e desenvolvimento web em um projeto com características próximas às encontradas no mercado de trabalho.

Durante a elaboração do sistema, foi possível aprofundar conhecimentos relacionados ao processo de desenvolvimento de software, compreendendo de forma mais ampla as etapas de levantamento de requisitos, definição arquitetural, organização de módulos, desenvolvimento incremental e documentação técnica. A necessidade de transformar uma ideia inicial em uma solução estruturada contribuiu significativamente para o desenvolvimento da capacidade de planejamento, análise de problemas e tomada de decisões técnicas.

A seleção das ferramentas proporcionou não só um aumento do conhecimento técnico, mas também a exposição a boas práticas de desenvolvimento, tais como a divisão em camadas, a modularização de componentes, a tipagem robusta de dados e a aprendizagem sobre técnicas de autenticação e gestão de acesso.

Outro aspecto relevante foi o aperfeiçoamento das habilidades de programação e resolução de problemas. Durante o desenvolvimento do Tasker, diversos desafios técnicos surgiram, exigindo pesquisa, experimentação e adaptação de soluções. Esse processo contribuiu para o fortalecimento do raciocínio lógico, da capacidade de aprendizagem contínua e da autonomia na busca por novos conhecimentos, competências fundamentais para a atuação profissional na área de tecnologia.

Embora o sistema tenha atingido os objetivos estabelecidos neste trabalho, o Tasker representa uma etapa inicial da solução proposta, sendo caracterizado como um *Minimum Viable Product* (MVP). Essa primeira versão foi desenvolvida com o propósito de concretizar e organizar as principais ideias, funcionalidades e objetivos da plataforma em uma aplicação funcional. Dessa forma, foi possível validar os requisitos centrais, a arquitetura adotada, as tecnologias selecionadas e a integração entre os módulos, demonstrando a viabilidade técnica da proposta e estabelecendo uma base estruturada para a continuidade do desenvolvimento.

Dessa forma, conclui-se que o desenvolvimento do Tasker não apenas resultou na concepção de uma solução para auxiliar a gestão de equipes e projetos, mas também representou um importante marco de crescimento acadêmico e profissional. Os conhecimentos adquiridos, as habilidades desenvolvidas e os desafios superados ao longo deste trabalho contribuíram significativamente para a formação do autor, reforçando a importância da aprendizagem contínua e da evolução técnica no contexto da área de Desenvolvimento de Sistemas.

4.1 TRABALHOS FUTUROS

Como próximos passos, será o melhoramenta da plataforma iniciada neste trabalho, abrangendo o refinamento dos fluxos, o aprimoramento das interfaces e a correção de inconsistências. Após a estabilização dessa versão, o Tasker será estendido com novas funcionalidades que ampliem sua capacidade de gerenciamento e colaboração. Entre as evoluções previstas estão o aperfeiçoamento dos relatórios gerenciais, a implementação de notificações em tempo real, integrações com plataformas de comunicação e a expansão dos recursos de acompanhamento de desempenho e produtividade.

Além das melhorias técnicas, será o estudo e a possibilidade de aplicação do sistema em pequenos negócios e equipes de menor porte. Muitas dessas organizações ainda realizam o gerenciamento de tarefas, projetos e comunicação interna por meio de ferramentas genéricas ou processos manuais. Nesse contexto, o Tasker poderá atuar como uma alternativa centralizada para organização das atividades, acompanhamento de prazos e colaboração entre membros, contribuindo para a melhoria dos processos internos e para o aumento da produtividade dessas organizações.

Dessa forma, os trabalhos futuros deverão conduzir o Tasker da condição de MVP inicial para uma plataforma finalizada e estável e, posteriormente, para uma solução progressivamente ampliada, capaz de incorporar novos recursos e de se adaptar às necessidades do mercado e de seus usuários.

REFERÊNCIAS

Amazon Web Services. **O que é um sistema de gerenciamento de banco de dados?** 2026. Acesso em: 7 ago. 2026. Disponível em: <https://aws.amazon.com/pt/what-is/dbms/>.

Axios. **Axios Documentation: First Steps**. 2026. Acesso em: 7 ago. 2026. Disponível em: <https://axios-http.com/docs/intro>.

Instituto Nacional da Propriedade Industrial. **Manual do usuário para o registro eletrônico de programas de computador**. Versão 1.7.2. Rio de Janeiro, 2018. Acesso em: 4 ago. 2026. Disponível em: <https://www.gov.br/inpi/pt-br/assuntos/arquivos-programa-de-computador/ManualdoUsurioV1.7.2.pdf>.

JONES, M.; BRADLEY, J.; SAKIMURA, N. **JSON Web Token (JWT)**. [S.l.], 2015. Acesso em: 7 ago. 2026. Disponível em: <https://www.rfc-editor.org/rfc/rfc7519>.

Meta Platforms. **React: Quick Start**. 2026. Acesso em: 7 ago. 2026. Disponível em: <https://react.dev/learn>.

Microsoft. **The TypeScript Handbook**. 2026. Acesso em: 7 ago. 2026. Disponível em: <https://www.typescriptlang.org/docs/handbook/intro.html>.

NestJS. **NestJS Documentation**. 2026. Acesso em: 7 ago. 2026. Disponível em: <https://docs.nestjs.com/>.

PostgreSQL Global Development Group. **PostgreSQL: About**. 2026. Acesso em: 7 ago. 2026. Disponível em: <https://www.postgresql.org/about/>.

PRESSMAN, R. S.; MAXIM, B. R. **Engenharia de software-9**. [S.l.]: McGraw Hill Brasil, 2021.

Prisma Data. **What is Prisma ORM?** 2026. Acesso em: 7 ago. 2026. Disponível em: <https://www.prisma.io/docs/orm>.

Project Management Institute. **Guia PMBOK: Um guia para o conjunto de conhecimentos em gerenciamento de projetos**. 6. ed. Newtown Square: PMI, 2017.

REZENDE, D. A.; ABREU, A. F. d. **Tecnologia da informação aplicada a sistemas de informação empresariais**. 9. ed. São Paulo: Atlas, 2020.

RODRIGUES, L. P.; COSTA, E. M. Ferramentas digitais de gestão de projetos e seu impacto na produtividade de equipes. **Revista Gestão & Tecnologia**, 2021.

SANTOS, G. P. **Implementação de controle de acesso federado baseado em atributos para ambientes de nuvem Apache CloudStack**. Florianópolis: [s.n.], 2024. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) – Universidade Federal de Santa Catarina. Acesso em: 4 ago. 2026. Disponível em: <https://repositorio.ufsc.br/bitstream/handle/123456789/262171/TCC.pdf>.

SANTOS, M. A.; LIMA, R. F. Tecnologia da informação aplicada ao gerenciamento de projetos nas organizações brasileiras. **Revista de Administração e Inovação**, 2019.

SILVA, F. S.; OLIVEIRA, I. d.; SILVA, M. B. C. d.; CARVALHO, L. G. d. Modelos de controle de acesso em bancos de dados na nuvem: uma análise comparativa. **Revista EduFatec: Educação, Tecnologia e Gestão**, v. 2, n. 6, p. 110–124, 2023. Acesso em: 4 ago. 2026. Disponível em: <https://revistaedufatec.fatecfranca.edu.br/wp-content/uploads/2024/04/edufatec-n06v2a07.pdf>.

VARGAS, R. V. **Manual prático do gerenciamento de projetos**. 10. ed. Rio de Janeiro: Brasport, 2018.

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA PARAÍBA
Campus Cajazeiras - Código INEP: 25008978
Rua José Antônio da Silva, 300, Jardim Oásis, CEP 58.900-000, Cajazeiras (PB)
CNPJ: 10.783.898/0005-07 - Telefone: (83) 3532-4100

Documento Digitalizado Ostensivo (Público)

Trabalho de conclusão de curso

Assunto:	Trabalho de conclusão de curso
Assinado por:	Mikael Stanley
Tipo do Documento:	Anexo
Situação:	Finalizado
Nível de Acesso:	Ostensivo (Público)
Tipo do Conferência:	Cópia Simples

Documento assinado eletronicamente por:

- **Mikael Stanley Trajano Lima**, **DISCENTE (202122010017) DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS - CAJAZEIRAS**, em 20/08/2026 06:10:56.

Este documento foi armazenado no SUAP em 20/08/2026. Para comprovar sua integridade, faça a leitura do QRCode ao lado ou acesse https://suap.ifpb.edu.br/verificar-documento-externo/ e forneça os dados abaixo:

Código Verificador: 1958499
Código de Autenticação: 5e9833670b