

**INSTITUTO FEDERAL DE RONDÔNIA**  
**CAMPUS PORTO VELHO CALAMA**  
**CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE E**  
**DESENVOLVIMENTO DE SISTEMAS**

---

**GUILHERME MOTA BARROSO**

**SISTEMA WEB DE INTERAÇÃO E GERENCIAMENTO DE ATIVIDADES**  
**UNIVERSITÁRIAS EM GRUPO**

**Porto Velho/RO**  
**2025**

**GUILHERME MOTA BARROSO**

**SISTEMA WEB DE INTERAÇÃO E GERENCIAMENTO DE ATIVIDADES  
UNIVERSITÁRIAS EM GRUPO**

Artigo entregue como Trabalho de Conclusão de Curso ao Instituto Federal de Educação, Ciência e Tecnologia de Rondônia (IFRO), *Campus* Porto Velho Calama, como requisito parcial para obtenção do grau de Tecnólogo, junto ao Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas.

Orientador: Leandro Ferrarezi Valiante

**Porto Velho/RO  
2025**

Ficha catalográfica elaborada pelo Sistema Gerador de Ficha Catalográfica do IFRO.

Barroso, Guilherme Mota.

Sistema web de interação e gerenciamento de atividades universitárias em grupo / Guilherme Mota Barroso. - Porto Velho, 2025.

26 f. : il.

Orientador(a): Prof. Leandro Ferrarezi Valiante.

Trabalho de Conclusão de Curso (Superior de Tecnologia em Análise e Desenvolvimento de Sistemas) – Instituto Federal de Educação, Ciência e Tecnologia de Rondônia - IFRO, Porto Velho, 2025.

1. sistema. 2. chat. 3. Atividades. 4. Acadêmico. 5. trabalho. I. Valiante, Leandro Ferrarezi (orient.). II. Instituto Federal de Educação, Ciência e Tecnologia de Rondônia - IFRO. III. Título.

**Bibliotecário(a) Responsável:** Miria Santana Veiga, CRB-11/898

**GUILHERME MOTA BARROSO**

**SISTEMA WEB DE INTERAÇÃO E GERENCIAMENTO DE ATIVIDADES  
UNIVERSITÁRIAS EM GRUPO**

A banca examinadora, abaixo listada, aprova o Trabalho de Conclusão de Curso “SISTEMA WEB DE INTERAÇÃO E GERENCIAMENTO DE ATIVIDADES UNIVERSITÁRIAS EM GRUPO” elaborado por “GUILHERME MOTA BARROSO” como requisito parcial para obtenção do grau de Tecnólogo em Análise e Desenvolvimento de Sistemas, pelo Instituto Federal de Educação, Ciência e Tecnologia de Rondônia.

Porto Velho/RO, 28/11/2025

**Comissão Examinadora**

---

**Leandro Ferrarezi Valiante - IFRO**  
(Orientador)

---

**Prof. Fernando Dall Igna - IFRO**

---

**Luis Fernando Bueno - TCERO**

## **SISTEMA WEB DE INTERAÇÃO E GERENCIAMENTO DE ATIVIDADES UNIVERSITÁRIAS EM GRUPO**

**RESUMO:** Este trabalho apresenta o desenvolvimento de um sistema web voltado para interação e gerenciamento de atividades de grupos universitários. A plataforma permite a interação de grupos por meio de chat integrado para comunicação em tempo real, criação de atividades permitindo o envio e sua conclusão, além de enquetes para votação para decisões coletivas. Para o desenvolvimento do sistema, foi utilizado o TDD (Test Driven Development), sendo utilizado o figma para prototipação, NextJs como tecnologia para desenvolvimento do sistema web e Jest para testes iterativos e automatizados. Como resultado, observa-se que o uso da plataforma proporciona maior organização das tarefas e melhora a comunicação entre os membros dos grupos. Conclui-se que a solução proposta contribui significativamente para a otimização do trabalho em equipe no ambiente universitário.

**PALAVRAS-CHAVE:** sistema. chat. atividades. acadêmico. trabalho.

**ABSTRACT:** This work presents the development of a web system aimed at interaction and management of activities for university groups. The platform enables group interaction through an integrated chat for real-time communication, creation of activities with file submission and completion tracking, as well as polls for collective decision-making. For the development of the system, Test Driven Development (TDD) methodology was applied, using Figma for prototyping, Next.js as the technology for web system development, and Jest for iterative and automated testing. As a result, the use of the platform enhances task organization and improves communication among group members. It is concluded that the proposed solution significantly contributes to optimizing teamwork in the university environment.

**KEYWORDS:** system. chat. activities. academic. work.

## 1 INTRODUÇÃO

O uso de plataformas digitais tornou-se uma prática comum entre estudantes universitários brasileiros. Uma pesquisa realizada com alunos de graduação em medicina revelou que 83% utilizam alguma plataforma digital para estudar, independentemente da instituição ser pública ou privada. Além disso, 91,2% dos usuários perceberam melhorias em seu conhecimento teórico ou prático, e 82,7% recomendariam a plataforma utilizada para outros colegas (RMMG - Revista Médica de Minas Gerais, 2021). Esse cenário reflete uma tendência crescente de adoção de tecnologias educacionais que facilitam o aprendizado e a colaboração acadêmica.

Paralelamente, observa-se uma mudança significativa nos hábitos de leitura dos estudantes, com uma crescente preferência por recursos digitais em detrimento dos livros físicos. Dados indicam que o percentual de pessoas que conhecem e consomem livros digitais aumentou de 30% em 2011 para 48% em 2024. Apesar disso, o formato físico ainda predomina, com 83% dos leitores relatando que o último livro lido foi em papel, contra apenas 16% no formato digital (Jornal da USP, 2024). Essa transição evidencia a importância de integrar recursos digitais no ambiente acadêmico, alinhando-se aos novos comportamentos de consumo de informação.

A crescente utilização de tecnologias digitais no ambiente acadêmico mudou a maneira de como os estudantes se organizam e interagem para o desenvolvimento de um trabalho ou atividade em grupo. No entanto, apesar da ampla disponibilidade de plataformas de comunicação, muitas ferramentas existentes não são específicas para atender às necessidades acadêmicas, carecendo de recursos que integrem comunicação, gestão de atividades e apoio à tomada de decisão de maneira unificada. De acordo com (Moran, 2000), a tecnologia educacional deve ser pensada como um meio para amplificar as possibilidades de aprendizagem, adequando-se às demandas específicas dos usuários.

Para ilustrar a lacuna de funcionalidades mencionada, e a fim de identificar as limitações das soluções de mercado mais utilizadas, a Figura 1 apresenta um comparativo entre as principais ferramentas de trabalho em grupo (Trello, Discord, Asana, Monday.com e Basecamp) em relação a funcionalidades cruciais para o contexto acadêmico, como chat em tempo real, gerenciamento de tarefas, gerenciamento de enquetes e separação do time por funções/cargos.

Figura 1 – Comparativo de funcionalidades das ferramentas de trabalho em grupo.

	Trello	Discord	Asana	Monday.com	Basecamp
Chat em tempo real	✗	✓	✗	✓	✓
Gerenciador de tarefas	✓	✗	✓	✓	✓
Gerenciador de enquetes	✗	✓	✗	✗	✗
Separação do time por funções/cargos	✓	✓	✓	✓	✓

Fonte: Elaborado pelo autor (2025).

A problemática que motivou esta pesquisa é a dificuldade enfrentada por estudantes na gestão integrada de atividades em grupo, em função da fragmentação de ferramentas atualmente utilizadas. Muitos estudantes recorrem a múltiplas plataformas — como aplicativos de mensagens, gerenciadores de tarefas e sistemas de votação —, o que gera dispersão da informação, desorganização e perda de produtividade (Ramos *et al.*, 2024). Assim, a ausência de um ambiente unificado e especializado impacta negativamente a eficiência dos trabalhos acadêmicos coletivos.

A justificativa para o desenvolvimento deste sistema reside na necessidade de otimizar o trabalho em equipe nas universidades, proporcionando aos estudantes uma ferramenta única, prática e acessível para centralizar a comunicação, gestão de tarefas e decisões em grupo. Além disso, a implementação de metodologias como o *Test Driven Development* (TDD) no processo de construção do sistema assegura maior qualidade e confiabilidade ao produto final, alinhando-se às melhores práticas de desenvolvimento de *software* (Beck, 2003).

Portanto, a criação do sistema proposto visa não apenas solucionar uma lacuna existente no contexto universitário, mas também contribuir para a melhoria dos processos de organização e colaboração entre estudantes, impactando positivamente seu desempenho acadêmico. Ao centralizar funcionalidades essenciais em um único ambiente digital, espera-se reduzir a dispersão de informações e facilitar o acompanhamento das atividades pelos membros do grupo. Além disso, a proposta considera a usabilidade e a acessibilidade como pilares fundamentais, garantindo que a ferramenta possa ser utilizada de forma intuitiva por estudantes com diferentes níveis de familiaridade com tecnologias. Com isso, o sistema busca promover uma cultura de trabalho colaborativo mais eficiente, autônoma e integrada às dinâmicas do ensino superior contemporâneo.

## 1.1 Objetivos

O objetivo central deste trabalho é desenvolver e implementar um sistema *web* que sirva como uma plataforma unificada para a interação e o gerenciamento de atividades de grupos universitários, visando superar a fragmentação de ferramentas e otimizar a colaboração acadêmica.

### 1.1.1 Objetivo Geral

Desenvolver um Sistema *Web* de Interação e Gerenciamento de Atividades Universitárias em Grupo que centralize a comunicação, a gestão de tarefas e o apoio à tomada de decisão, proporcionando uma ferramenta única, prática e acessível para otimizar o trabalho em equipe no ambiente acadêmico.

### 1.1.2 Objetivos Específicos

Para alcançar o objetivo geral, o trabalho se propõe a:

1. Realizar o levantamento e a análise dos requisitos funcionais e não funcionais necessários para a gestão integrada de grupos universitários.
2. Projetar a arquitetura do sistema *web*, definindo as tecnologias e metodologias de desenvolvimento (Next.js, Fastify, PostgreSQL, MongoDB, TDD) mais adequadas.
3. Implementar um módulo de autenticação seguro, utilizando Auth.js e JWT, para gerenciar o acesso e as sessões dos usuários.
4. Desenvolver um módulo de grupos que permita a criação, o convite de membros e a atribuição de permissões através de um sistema flexível de cargos, que podem ser criados e personalizados pelo administrador do grupo.
5. Construir um módulo de comunicação com *chat* em tempo real, utilizando banco de dados NoSQL (MongoDB) para otimizar a performance e a escalabilidade.
6. Criar um módulo de gerenciamento de atividades que possibilite a criação, o envio de arquivos e o acompanhamento do *status* de conclusão das tarefas.
7. Implementar um recurso de enquetes para votação coletiva, facilitando a tomada de decisões dentro dos grupos.
8. Aplicar a metodologia *Test Driven Development* (TDD) e testes automatizados (Jest) para garantir a qualidade, a confiabilidade e a manutenibilidade do código-fonte.

## 2 FUNDAMENTAÇÃO TEÓRICA

Esta seção apresenta a base teórica e técnica que fundamenta o desenvolvimento do Sistema Web de Interação e Gerenciamento de Atividades Universitárias em Grupo. Para garantir a qualidade e a escalabilidade da solução, serão explorados os conceitos de colaboração acadêmica e as tecnologias de software utilizadas. O objetivo é detalhar, com base em referências acadêmicas e literatura especializada, os pilares que sustentam a construção do sistema.

A colaboração em grupo é um pilar essencial no ambiente universitário, sendo um fator crítico para o desenvolvimento de habilidades interpessoais e a construção conjunta de conhecimento. A literatura sobre aprendizagem colaborativa, como a abordada por (Ramos *et al.*, 2024), enfatiza que a eficácia do trabalho em equipe depende da capacidade dos membros de interagirem, negociarem significados e gerenciarem suas atividades de forma coordenada. A fragmentação de ferramentas de comunicação e gestão, no entanto, pode comprometer essa coordenação, justificando a necessidade de plataformas unificadas que promovam a sinergia e a organização, como a proposta neste trabalho.

### 2.1 Tecnologias de Desenvolvimento

A prototipagem é uma etapa fundamental no ciclo de vida do desenvolvimento de sistemas, pois permite a transformação de conceitos abstratos em representações tangíveis, facilitando a validação de requisitos e a minimização de riscos associados a mudanças (Soares; Resende, 2017). A utilização de ferramentas como o Figma para a criação de protótipos interativos e realistas, como mencionado por (Theis *et al.*, 2021), auxilia na representação visual das ideias e na obtenção de *feedback* precoce dos usuários, garantindo que o sistema final possua um *design* atrativo e uma usabilidade otimizada para o contexto acadêmico.

O Next.js é um *framework* de desenvolvimento *web* baseado em React, reconhecido por sua capacidade de criar aplicações de alto desempenho através de funcionalidades como a Renderização do Lado do Servidor (SSR) e a Geração de Sites Estáticos (SSG) (Vercel, 2025). A adoção do Next.js é justificada pela necessidade de tempos de carregamento rápidos e uma melhor experiência do usuário, fatores cruciais em plataformas de interação em tempo real. Pesquisas recentes, como as de (Lazuardy; Anggraini, 2022), destacam o papel do Next.js na arquitetura *web* moderna, permitindo a construção de sistemas escaláveis e responsivos.

Jest é um *framework* de testes JavaScript amplamente utilizado, especialmente em ecossistemas React e Next.js, devido à sua API acessível e rica em recursos para

---

testes unitários e de integração. Sua escolha se alinha à metodologia TDD, permitindo a criação de testes automatizados que garantem a correção e a estabilidade de qualquer base de código JavaScript (Donvir, 2024). A eficácia do Jest em ambientes de desenvolvimento moderno é confirmada em análises comparativas de *frameworks* de teste, que o apontam como uma solução robusta para a garantia de qualidade em aplicações *web*.

Fastify é um *framework web* para Node.js projetado com foco em alta performance e baixo *overhead*, sendo considerado um dos mais rápidos em seu ecossistema (Spigolon, 2023). Sua arquitetura leve e baseada em *plugins* o torna ideal para a construção de uma API (*Application Programming Interface*) eficiente, capaz de lidar com a carga de requisições de um sistema de gerenciamento e chat em tempo real. A escolha do Fastify para o *back-end* visa garantir uma resposta rápida sob carga, essencial para a experiência fluida dos usuários universitários.

O *JSON Web Token* (JWT) é um padrão aberto (RFC 7519) que define uma forma compacta e segura de transmitir informações entre partes como um objeto JSON. No contexto de autenticação, o JWT é utilizado para gerenciar sessões de usuário de forma segura e sem estado, permitindo que o servidor verifique a identidade do usuário sem a necessidade de consultar o banco de dados a cada requisição (Okta, 2025). A implementação do JWT, frequentemente em conjunto com bibliotecas de autenticação modernas, é uma prática recomendada para reforçar a segurança e a integridade das sessões em aplicações *web*.

Neste contexto, o uso de bibliotecas como o Auth.js, se justifica por oferecer uma solução completa e flexível para autenticação, abstraindo a complexidade da implementação segura de JWT e de outros protocolos de autenticação, e seguindo as melhores práticas de segurança para aplicações *web* modernas (Auth.js Contributors, 2025). A adoção do Auth.js, portanto, permite que o desenvolvimento se concentre na lógica de negócio, enquanto a segurança da sessão é tratada por uma ferramenta robusta e amplamente utilizada pela comunidade.

O *Test Driven Development* (TDD) é uma metodologia de desenvolvimento de *software* que combina *design* e teste em um ciclo iterativo e incremental, conforme popularizado por (Beck, 2003). A prática consiste em escrever um teste automatizado que falha antes de escrever o código mínimo necessário para fazê-lo passar, e então refatorar o código. A adoção do TDD não apenas assegura a correção do código e reduz a incidência de defeitos, mas também contribui significativamente para a melhoria da qualidade do *design* e a manutenibilidade a longo prazo do *software*.

## 2.2 Persistência de Dados

O PostgreSQL é um sistema gerenciador de banco de dados objeto-relacional de código aberto, amplamente reconhecido por sua robustez, confiabilidade e estrita conformidade com os padrões ACID (Atomicidade, Consistência, Isolamento, Durabilidade) (PostgreSQL Global Development Group, 2025). Sua capacidade de lidar com tipos de dados avançados e consultas complexas o torna a escolha ideal para o armazenamento de dados estruturados e críticos do sistema, como informações de usuários, grupos e atividades, onde a integridade e a consistência transacional são prioritárias (Wodyk; Skublewska-Paszkowska, 2020).

O MongoDB é um banco de dados NoSQL orientado a documentos, escolhido especificamente para o armazenamento das conversas do *chat* em tempo real devido à sua flexibilidade e otimização para grandes volumes de escrita com baixa latência. Sua arquitetura permite modelar mensagens com todos os seus metadados em um único documento BSON, eliminando a necessidade de *joins* complexos e facilitando a leitura eficiente, o que é crucial para sistemas de comunicação em tempo real (Chodorow, 2013). A escalabilidade horizontal nativa do MongoDB via *sharding* o torna vantajoso para lidar com o crescimento exponencial de dados de *chat*, conforme discutido por (Yuan; Wang; Zhang, 2014).

## 3 METODOLOGIA

A metodologia empregada neste artigo foi estruturada para garantir o desenvolvimento de um produto de software funcional e relevante, o *Sistema Web de Interação e Gerenciamento de Atividades Universitárias em Grupo*, com base em um rigoroso processo de pesquisa e engenharia de software.

O presente trabalho se enquadra como uma Pesquisa Aplicada, visto que seu objetivo principal é gerar conhecimentos para aplicação prática, visando a solução de um problema específico, conforme a definição de Prodanov (Prodanov; Freitas, 2013). Neste contexto, o problema central é a fragmentação de ferramentas e a dificuldade na gestão integrada de atividades em grupo no ambiente universitário, cuja solução proposta é o desenvolvimento do sistema web.

Quanto à sua natureza, a pesquisa possui uma fase inicial de caráter Exploratório, buscando maior familiaridade com o problema e o ambiente acadêmico por meio da revisão bibliográfica e da coleta de requisitos.

Posteriormente, o trabalho se torna Descritivo ao detalhar as características, funcionalidades e tecnologias empregadas no sistema desenvolvido. Por fim, a pesquisa culmina no Desenvolvimento de Produto (*Software*), sendo o sistema web o pro-

duto final, o que alinha o trabalho com a metodologia de desenvolvimento de software, focando na criação e validação de uma solução tecnológica para o problema identificado.

## 4 DESENVOLVIMENTO

Após a conclusão das fases iniciais de planejamento e validação dos requisitos e funcionalidades do sistema, iniciou-se a etapa de desenvolvimento. Esta fase concentrou-se na implementação da interface do usuário (*front-end*) e da lógica de negócios e serviços (*back-end*), utilizando as tecnologias definidas na Fundamentação Teórica. O desenvolvimento foi conduzido de forma modular, com foco na criação das funcionalidades essenciais para a aplicação.

### 4.1 Passo-a-Passo do Desenvolvimento

O processo de desenvolvimento do trabalho seguiu um ciclo iterativo e incremental, adaptado para a criação de um sistema *web*, conforme detalhado a seguir:

1. Revisão da Literatura e Fundamentação Teórica: Inicialmente, foi realizada uma revisão da literatura para estabelecer a fundamentação teórica do trabalho. Esta etapa focou em conceitos de colaboração acadêmica, metodologias de desenvolvimento de *software* (como TDD) e as tecnologias-chave (Next.js, Fastify, PostgreSQL, MongoDB, JWT, Figma), garantindo o embasamento científico e técnico do projeto.
2. Coleta de Requisitos e Enquetes: Em seguida, foi realizada uma coleta de requisitos por meio de enquetes (questionários) aplicadas a usuários-chave. Esta fase exploratória foi crucial para levantar as opiniões dos usuários sobre a necessidade de um sistema unificado e para validar os requisitos funcionais e não funcionais, identificando as funcionalidades essenciais para os grupos universitários, como a necessidade de *chat* em tempo real, gerenciamento de atividades e enquetes.
3. Modelagem e Prototipagem (Documento AR): Tendo as informações necessárias, foi construído um documento de Análise de Requisitos (AR) detalhado. Paralelamente, foi utilizada a ferramenta Figma para a prototipagem da interface do usuário (UI) e da experiência do usuário (UX), transformando os requisitos em representações visuais e interativas.
4. Desenvolvimento Iterativo (TDD): A fase de implementação do *software* foi conduzida utilizando a metodologia Test Driven Development (TDD). O desenvol-

vimento foi dividido em módulos (Autenticação, Grupos, Chat, Atividades, Enquetes), sendo cada funcionalidade construída em ciclos curtos de teste-código-refatoração, garantindo a qualidade e a estabilidade do código.

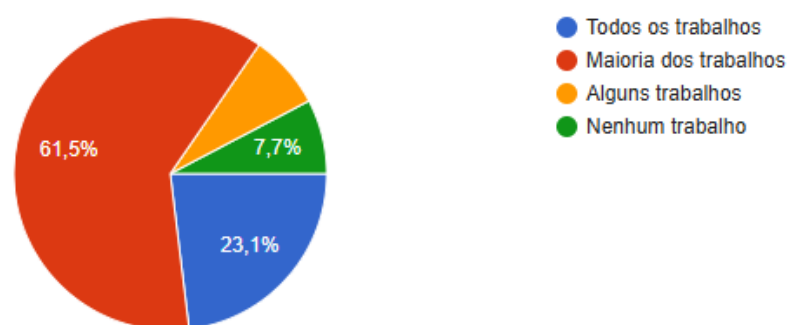
5. Testes e Validação: Após o desenvolvimento de cada módulo, foram realizados testes automatizados (com Jest) e testes de integração. A validação final do sistema foi realizada por meio de testes de usabilidade e, idealmente, com a aplicação de um questionário (mencionado na seção de Resultados Esperados do artigo) para avaliar a eficácia da solução na melhoria da organização e comunicação dos grupos universitários.

## 4.2 Levantamento de Requisitos com Questionário

Para embasar a proposta de desenvolvimento do sistema e validar a problemática da fragmentação de ferramentas, foi aplicado um questionário a 13 estudantes universitários de diferentes períodos e graduações. A análise dos dados coletados forneceu subsídios cruciais para a definição dos requisitos e a justificativa da solução.

O primeiro dado relevante aponta que 84,6% dos respondentes realizam trabalhos em grupo com frequência, o que evidencia a natureza essencial e recorrente das atividades colaborativas no ambiente acadêmico. Este alto índice reforça a necessidade de ferramentas que otimizem e suportem este processo.

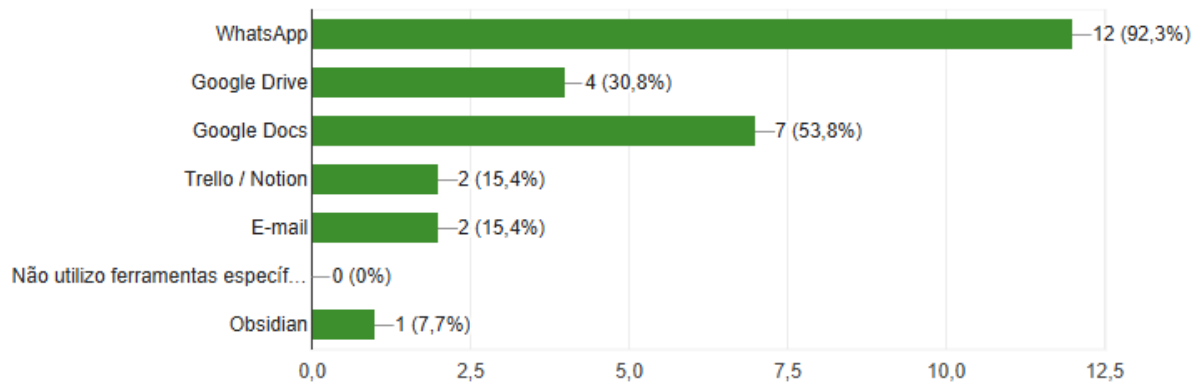
Figura 2 – Frequência de trabalhos na faculdade



Fonte: Elaborado pelo autor (2025).

A investigação sobre as ferramentas atualmente utilizadas revelou que a maioria dos estudantes recorre a plataformas como WhatsApp, Google Docs e Google Drive para a execução de trabalhos em grupo. Este achado demonstra que a comunicação em tempo real, a edição colaborativa de documentos e o armazenamento em nuvem são funcionalidades percebidas como indispensáveis para o sucesso das atividades.

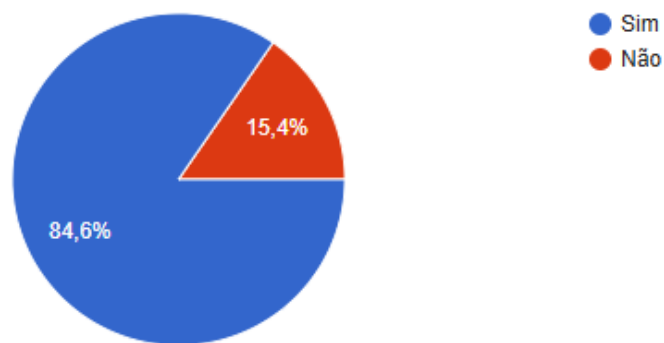
Figura 3 – Ferramentas utilizadas para realização dos trabalhos



Fonte: Elaborado pelo autor (2025).

Um ponto crítico levantado pelo questionário diz respeito à gestão de tempo e prazos: 84,6% dos estudantes relataram ter enfrentado problemas com o cumprimento de prazos em trabalhos em grupo. Este resultado sublinha a urgência de um sistema que ofereça recursos robustos de gerenciamento de atividades e acompanhamento de *deadlines*, mitigando a dispersão e a desorganização.

Figura 4 – Problemas com prazos de trabalhos

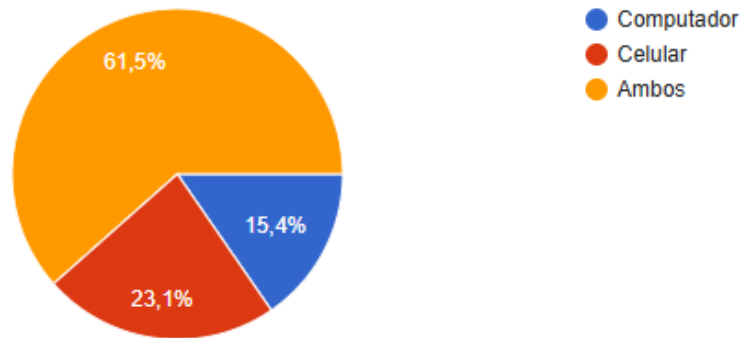


Fonte: Elaborado pelo autor (2025).

Por fim, 61,5% dos estudantes consideram o uso de dispositivos móveis (celulares) e computadores essenciais para a realização de trabalhos. Este dado estabelece um requisito fundamental de *design*: o sistema deve ser responsivo e acessível em diferentes plataformas, garantindo a usabilidade e a ampla adoção por parte do público-alvo.

Em síntese, os resultados do questionário confirmam a alta demanda por ferramentas de suporte ao trabalho em grupo e validam a hipótese de que a dispersão de atividades entre múltiplas plataformas (WhatsApp, Google Docs, Google Drive) gera ineficiência. A solução proposta, ao unificar comunicação, gestão de tarefas e arma-

Figura 5 – Preferência por dispositivos em trabalhos



Fonte: Elaborado pelo autor (2025).

zenamento em um único sistema responsivo, atende diretamente às necessidades e aos desafios identificados pelos próprios estudantes universitários.

### 4.3 Arquitetura e Tecnologias

O *front-end* do sistema foi desenvolvido utilizando Next.js, aproveitando sua capacidade de *Server-Side Rendering* (SSR) e *Static Site Generation* (SSG) para garantir alto desempenho e tempos de carregamento rápidos. A linguagem de programação utilizada foi o TypeScript, um *superset* do JavaScript que adiciona tipagem estática, aumentando a segurança e a manutenibilidade do código. Para os testes automatizados, o *framework* Jest foi empregado, alinhando-se à metodologia *Test Driven Development* (TDD) adotada.

No *back-end*, optou-se pela utilização da API com Fastify, um servidor Node.js reconhecido por sua eficiência e baixa sobrecarga, garantindo uma ótima capacidade de resposta sob carga. O gerenciamento de sessões e a autenticação dos usuários foram implementados com a biblioteca Auth.Js, utilizando *JSON Web Token* (JWT) para o intercâmbio seguro de informações e a manutenção da sessão nos *cookies* do site.

A persistência de dados foi dividida entre dois sistemas de gerenciamento de banco de dados:

- PostgreSQL: Adotado para armazenar dados estruturados e críticos, como informações de usuários, grupos, atividades e convites, devido à sua estabilidade e conformidade com os padrões ACID.
- MongoDB: Utilizado para o armazenamento das conversas do chat em tempo real, devido à sua adequação a grandes volumes de escrita com baixa latência e sua flexibilidade como banco de dados NoSQL orientado a documentos.

## 4.4 Módulos Centrais e Funcionalidades

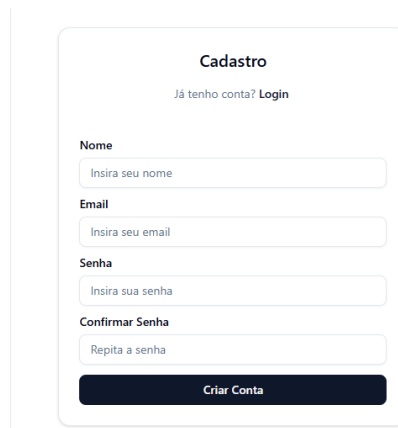
O sistema foi estruturado em módulos centrais, cada um responsável por um conjunto específico de funcionalidades essenciais para a interação e o gerenciamento de atividades dos grupos universitários.

### 4.4.1 Módulo de Autenticação

O módulo de autenticação foi projetado para garantir a segurança dos acessos. A interface de *login* permite que os usuários acessem o sistema utilizando e-mail e senha. O processo inclui:

- Criação de Conta: O usuário insere nome de usuário, e-mail e senha. Após a validação do e-mail e a verificação de unicidade do nome de usuário, a conta é criada. Na Figura 6, é possível verificar a *interface* da tela de cadastro, onde o usuário insere as informações necessárias para a criação da conta.
- Login e Sessão: Após o *login*, o JWT do usuário é criado e mantido nos *cookies* do site, proporcionando uma forma prática e segura de manter a sessão e verificar o acesso do usuário, reforçando a segurança da aplicação.

Figura 6 – Tela de cadastro



Cadastro

Já tenho conta? [Login](#)

Nome

Email

Senha

Confirmar Senha

[Criar Conta](#)

Fonte: Elaborado pelo autor (2025).

### 4.4.2 Módulo de Grupo

O módulo de grupos é o componente central para a organização do trabalho em equipe. Ele permite que o usuário crie seu próprio grupo ou seja convidado a participar de um existente. As funcionalidades incluem:

Figura 7 – Tela de criação do grupo (etapa 1)

**Criar grupo**

Nome do Grupo  
Programação Web

Descrição (Opcional)  
Grupo voltado para o trabalho de programação web

Avançar etapa

Fonte: Elaborado pelo autor (2025).

- **Criação e Convite:** O usuário pode criar um grupo com nome e descrição e enviar convites por e-mail para os integrantes. Na Figura 7 (etapa 1), é possível verificar a interface para a inserção do nome e descrição do grupo. Em seguida, na Figura 8 (etapa 2), é apresentada a tela para o envio de convites por e-mail aos participantes.

Figura 8 – Tela de criação do grupo (etapa 2)

**Adicionar pessoas**

Email do integrante  
joanafigueredo@gmail.com Adicionar usuário

J joanafigueredo@gmail.com

A anavictoria@gmail.com

R ricardomello@gmail.com

F fernandoaugusto@gmail.com

R ronaldorodrigues@gmail.com

G guimota22@gmail.com

Avançar etapa

Fonte: Elaborado pelo autor (2025).

- **Gestão de Cargos e Permissões Personalizadas:** O sistema implementa um modelo flexível onde os cargos não são predefinidos. Em vez disso, cada grupo pode criar cargos personalizados e atribuir a eles um conjunto granular de permissões. Na Figura 9 (etapa 3), é possível verificar a interface para a criação de

cargos personalizados e a atribuição de permissões. As permissões disponíveis incluem:

- Administrador: Permissão de administrador geral.
- Gerenciar atividades: Gerenciar (criar/excluir) atividades.
- Criar atividades: Criar novas atividades.
- Gerenciar cargos: Gerenciar (criar/editar/excluir) cargos existentes.
- Atualizar grupo: Atualizar informações do grupo (nome, descrição).
- Gerenciar participantes: Remover ou gerenciar cargos de participantes.
- Convidar participantes: Convidar novos participantes para o grupo.

Figura 9 – Tela de criação do grupo (etapa 3)

Criar grupo

Adicionar pessoas

**Criar cargos**

Atribuir cargos

Finalizar

**Criar cargos** + Adicionar

Defina os cargos personalizados para o seu grupo e atribua as permissões específicas para cada um.

**Administrador** [Ícone de edição] [Ícone de exclusão]

Permissões:

Administrador

**Tarefeiro** [Ícone de edição] [Ícone de exclusão]

Permissões:

Criar atividades Gerenciar atividades

**Sociável** [Ícone de edição] [Ícone de exclusão]

Permissões:

Convidar participantes Criar cargos Gerenciar cargos Gerenciar participantes

Avançar etapa

Fonte: Elaborado pelo autor (2025).

Após a criação dos cargos, a Figura 10 (etapa 4) ilustra a tela onde esses cargos são atribuídos aos participantes.

Figura 10 – Tela de criação do grupo (etapa 4)

The screenshot shows a web interface for creating a group. On the left, a vertical sidebar contains five buttons: 'Criar grupo', 'Adicionar pessoas', 'Criar cargos', 'Atribuir cargos' (highlighted in dark blue), and 'Finalizar'. The main area is titled 'Atribuir Cargos' and displays a list of six participants, each with a circular profile icon and an email address. The first participant, 'G guimota22@gmail.com', has a dark blue 'Líder (você)' badge. At the bottom of the main area is a dark blue button labeled 'Avançar etapa'.

Initial	Email	Role
G	guimota22@gmail.com	Líder (você)
R	ronaldorodrigues@gmail.com	
F	fernandoaugusto@gmail.com	
R	ricardomello@gmail.com	
A	anavictoria@gmail.com	
J	joanafigueredo@gmail.com	

Fonte: Elaborado pelo autor (2025).

- **Interação:** Após a criação e inserção dos participantes, o grupo tem acesso ao *chat* em tempo real, ao gerenciamento de atividades e à votação de enquetes, com as funcionalidades acessíveis de acordo com as permissões atribuídas ao seu cargo. Na Figura 11 (resumo), é possível verificar a tela final de resumo da criação do grupo.

Figura 11 – Tela de criação do grupo (resumo)

**Criar grupo**

**Adicionar pessoas**

**Criar cargos**

**Atribuir cargos**

**Finalizar**

**Revise as informações**

Clique nas etapas ao lado para fazer alterações se necessário.

**Nome do grupo**

Programação Web

**Descrição do grupo**

Grupo voltado para o trabalho de programação web

**Participantes**

G	guimota22@gmail.com	Líder (você)
R	ronaldorodrigues@gmail.com	Sociável
F	fernandoaugusto@gmail.com	Sociável
R	ricardomello@gmail.com	Tarefeiro
A	anavictoria@gmail.com	Tarefeiro
J	joanafigueredo@gmail.com	Administrador

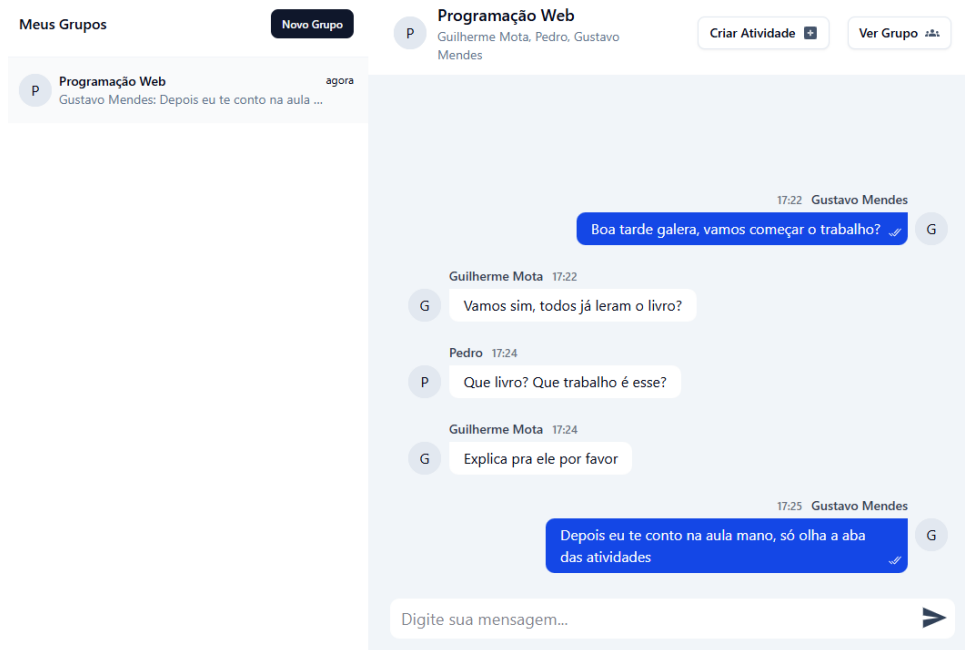
**Voltar** **Finalizar**

Fonte: Elaborado pelo autor (2025).

#### 4.4.3 Módulo de *Chat* em Tempo Real

O módulo de *chat* em tempo real é criado automaticamente com a formação do grupo, permitindo a conversa e a interação imediata entre os participantes. Este recurso elimina a necessidade de sistemas terceiros para comunicação, centralizando a interação no ambiente do sistema. O MongoDB foi fundamental para este módulo, otimizado para o alto volume de escrita e baixa latência exigidos por um sistema de chat (conforme a Figura 12).

Figura 12 – Tela do chat iterativo



Fonte: Elaborado pelo autor (2025).

#### 4.4.4 Módulo de Gerenciamento de Atividades

Esta funcionalidade é controlada pelas permissões `Criar atividades` e `Gerenciar atividades`, permitindo que apenas os participantes com os cargos adequados criem e gerenciem as atividades para a equipe. As principais características são:

- **Criação de Atividades:** Usuários com a permissão `Criar atividades` têm a possibilidade de destinar a atividade para participantes específicos ou para todos, com ou sem a opção de anexar arquivos. Na Figura 13, é possível verificar a interface de criação de atividades, onde o usuário define os detalhes e destinatários da tarefa.

Figura 13 – Tela de criação de tarefa

**Criar tarefa**

**Título**  
Insira o título

**Descrição**  
Insira a descrição

**Data de início**  
23/11/2025 11:50

**Data de término**  
24/11/2025 11:50

Obrigatório anexar arquivos?

Atribuir tarefas à

Administrador  
 Tarefairo  
 Sociável

**Criar**

Fonte: Elaborado pelo autor (2025).

- Acompanhamento e Gestão: Usuários com a permissão Gerenciar atividades podem visualizar e gerenciar o envio das atividades, verificando quem realizou a tarefa e os arquivos enviados, garantindo o acompanhamento do progresso do grupo. A Figura 14 ilustra a tela de envio de tarefas, permitindo aos gestores acompanhar o status de cada participante.

Figura 14 – Tela de envio de tarefas

**Tarefa para envio de documentação** Em progresso

Envie sua documentação da criação do projeto

Início: 18/11/2025 16:59 Fim: 03/04/2026 16:59

**Anexar Documento (PDF)**

Esta tarefa requer o anexo de um arquivo PDF. Salve o documento antes de concluir a atividade.

Documento OPO 1.pdf

**Salvar Documento**

**Arquivos Anexados**

Documento OPO 2.pdf

**Voltar** **Marcar como Finalizado**

Fonte: Elaborado pelo autor (2025).

#### 4.4.5 Módulo de Enquetes

O módulo de enquetes é projetado para facilitar a tomada de decisões coletivas e transparentes. A criação de enquetes é controlada pela permissão *Criar atividade* e *Gerenciar atividades*, garantindo que apenas membros com autoridade para gerir tarefas possam iniciar votações. Na Figura 15, é possível verificar a tela de criação de votação, onde são configurados o título, as opções, o período e os pesos de votação. As opções de criação incluem:

- Configuração: Criação de título e adição de opções de escolha.
- Tipos de votação: Opções de múltipla escolha ou escolha única.
- Período de votação: Configuração da data de início e término da votação.
- Pesos na votação: Possibilidade de cargos possuírem pesos diferentes nos votos.
- Visualização de Resultados: O resultado da enquete pode ser visto por todos os participantes, independentemente do cargo, promovendo transparência na tomada de decisões.

Figura 15 – Tela de criação de votação

A imagem mostra a interface de usuário para a criação de uma votação. O formulário é dividido em seções:

- Criar Votação:** Título principal da seção.
- Nome da votação:** Campo de texto com o placeholder "Insira o nome da votação".
- Data de início:** Campo com data "23/11/2025" e hora "11:56".
- Data de término:** Campo com data "24/11/2025" e hora "11:56".
- Votação com pesos personalizados?** Seção com um interruptor "Habilitar votação com pesos personalizados" desativado e o texto "Ative essa opção para personalizar o peso dos votos de cada cargo".
- Pesos do voto:** Campos para definir pesos para "Sociável" e "Administrador", cada um com opções 1, 2 e 3.
- Pergunta da votação:** Campo de texto com o placeholder "Insira a pergunta da votação".
- Opções:** Duas linhas de campo de texto com o placeholder "Insira a opção...".
- Permitir várias votações:** Um interruptor desativado.
- Botão Criar:** Um botão azul escuro no rodapé do formulário.

Fonte: Elaborado pelo autor (2025).

## 5 CONCLUSÃO

O presente trabalho alcançou seu objetivo principal ao propor e desenvolver um Sistema *Web* de Interação e Gerenciamento de Atividades Universitárias em Grupo. A

problemática central, identificada pela fragmentação de ferramentas e a consequente ineficiência na gestão de trabalhos colaborativos, foi abordada por meio da criação de uma plataforma unificada e especializada.

A base teórica e metodológica adotada, que incluiu a revisão bibliográfica sobre colaboração acadêmica e a aplicação de metodologias ágeis como o *Test Driven Development* (TDD), garantiu o rigor científico e a qualidade técnica do produto. A análise dos resultados do questionário aplicado a estudantes universitários validou a necessidade do sistema, confirmando que a maioria dos alunos realiza trabalhos em grupo frequentemente e enfrenta problemas críticos com a gestão de prazos e a dispersão de informações.

O sistema desenvolvido, utilizando tecnologias modernas como Next.js, Fastify, PostgreSQL e MongoDB, demonstrou ser uma solução robusta e escalável, capaz de centralizar a comunicação (*chat* em tempo real), a gestão de tarefas (atividades e enquetes) e o controle de acesso (cargos e permissões personalizadas).

Em suma, a solução desenvolvida contribui de maneira significativa para a melhoria da gestão de trabalhos em grupo no contexto universitário, promovendo uma cultura de colaboração mais estruturada, eficiente e integrada às demandas do ensino superior contemporâneo.

Apesar do sucesso na conclusão dos objetivos propostos, o desenvolvimento do sistema abre portas para diversas possibilidades de expansão e aprimoramento, que podem ser exploradas em trabalhos futuros. As principais sugestões incluem:

- **Integração com Sistemas Acadêmicos:** Implementar a integração com sistemas de gestão acadêmica (como SIGAA, Moodle ou Google Classroom) para facilitar a importação de listas de alunos e a sincronização de atividades e prazos.
- **Notificações Inteligentes e Lembretes:** Aprimorar o sistema de notificações com lembretes inteligentes e personalizáveis, que utilizem algoritmos para prever atrasos e alertar os membros do grupo proativamente.
- **Funcionalidade de Videoconferência Integrada:** Adicionar um recurso de videoconferência nativo ao sistema, eliminando a necessidade de ferramentas externas para reuniões rápidas de grupo.
- **Análise de Dados e Relatórios:** Criar um painel de análise de dados para que os usuários possam visualizar métricas de produtividade do grupo, como tempo médio de conclusão de tarefas e engajamento nas enquetes.

## REFERÊNCIAS

AUTH.JS CONTRIBUTORS. **Auth.js | Authentication for the Web**. 2025. <https://authjs.dev/>. Acesso em: [Data de Acesso].

BECK, Kent. **Test Driven Development: By Example**. Addison-Wesley, 2003.

CHODOROW, Kristina. **MongoDB: The Definitive Guide**. O'Reilly Media, 2013.

DONVIR, A. A Comparative Analysis of JavaScript Unit and End-to-End Testing Frameworks: Enhancing Quality Assurance in Modern Web Applications. **IEEE Xplore**, 2024.

JORNAL DA USP. **A digitalização da leitura e o consumo de informações**. 2024. <https://jornal.usp.br/artigos/a-digitalizacao-da-leitura-e-o-consumo-de-informacoes/>. Acesso em: 22 mai. 2025.

LAZUARDY, M. F. S.; ANGGRAINI, D. Modern front end web architectures with react.js and next.js. **Research Journal of Advanced Engineering and Science**, v. 7, n. 1, p. 162–167, 2022.

MORAN, José Manuel. **A educação que desejamos: novos desafios e como chegar lá**. Papyrus Editora, 2000.

OKTA. **Introduction to json web tokens**. 2025. <https://jwt.io/introduction/>. Acesso em: 08/05/2025.

POSTGRESQL GLOBAL DEVELOPMENT GROUP. **Postgresql: Documentation for object-relational database system**. 2025. <https://www.postgresql.org/docs/>. Acesso em: 02/05/2025.

PRODANOV, Cleber Cristiano; FREITAS, Everaldo Cezar de. **Metodologia do trabalho científico: métodos e técnicas da pesquisa e do trabalho acadêmico**. Feevale, 2013.

RAMOS, Celso de Ávila *et al.* PLATAFORMA INTEGRADA UNIVERSITÁRIA PARA GERENCIAMENTO DE PROJETOS EM PBL. **Revista Científica da UNIFENAS**, v. 6, n. 5, 2024. Disponível em: <https://revistas.unifenas.br/index.php/revistaunifenas/article/view/1061>.

RMMG - REVISTA MÉDICA DE MINAS GERAIS. Uso de Tecnologia Educacional durante a graduação médica. **Revista Médica de Minas Gerais**, v. 31, n. 1, p. 1–10, 2021. Acesso em: 22 mai. 2025.

SOARES, B. C.; RESENDE, R. S. Requisitos para utilização de prototipagem evolutiva nos processos de desenvolvimento de software para web. **Universidade Federal de Minas Gerais (UFMG)**, 2017.

SPIGOLON, M. Accelerating Server-Side Development with Fastify: A Complete Guide. **IEEE Xplore**, 2023.

THEIS, M. R. *et al.* A importância da prototipagem no processo de design e suas relações como mídia do conhecimento. *In: ANAIS do Congresso Internacional de Conhecimento e Inovação – CIKI*. 2021. v. 1.

VERCEL. **Next.js: Server-side rendering (ssr) and static site generation (ssg)**. 2025. <https://nextjs.org/docs/pages/building-your-application/rendering>. Acesso em: 02/05/2025.

WODYK, R.; SKUBLEWSKA-PASZKOWSKA, M. Performance comparison of relational databases SQL Server, MySQL and PostgreSQL using a web application and the Laravel framework. **Journal of Computer Sciences Institute**, v. 15, p. 1–10, 2020.

YUAN, D.; WANG, Z.; ZHANG, H. Evaluation of NoSQL Databases for Massive IoT Data Storage. *In: IEEE*. 2014.