

Sistema de Monitoria-IC: Plataforma Web para Gestão Completa de Monitorias Acadêmicas da UFBA

Sistema de Monitoria-IC: Web Platform for Complete Management of Academic Monitoring Programs at UFBA

Luis Felipe Cordeiro Sena   [Universidade Federal da Bahia | luis.sena@ufba.br]

Frederico Araújo Durão  [Universidade Federal da Bahia | fdurao@ufba.br]

 Instituto de Computação, Universidade Federal da Bahia, Av. Milton Santos, s/n - Campus de Ondina, PAF 2, Salvador, BA, 40170-110, Brasil.

Resumo. A monitoria acadêmica é um processo fundamental nas universidades brasileiras que permite aos alunos desenvolverem habilidades pedagógicas enquanto auxiliam no processo de ensino-aprendizagem. Porém, o gerenciamento desse processo frequentemente enfrenta desafios relacionados à burocracia, falta de transparência e múltiplos processos manuais ineficientes. Este trabalho apresenta o desenvolvimento do Sistema de Monitoria-IC, uma plataforma Web projetada para automatizar e simplificar o ciclo de vida dos projetos de monitoria na UFBA. A solução proposta digitaliza desde a criação e assinatura de projetos pelos professores até a aprovação administrativa e geração de planilhas consolidadas para envio ao Instituto/PROGRAD. O sistema foi desenvolvido utilizando tecnologias modernas como Next.js 15, TypeScript, tRPC, PostgreSQL e MinIO, seguindo princípios de engenharia de software que garantem escalabilidade e manutenibilidade. A arquitetura implementada separa claramente as responsabilidades entre um Sistema de Processamento de Transações (SPT) para operações cotidianas e funcionalidades gerenciais para análise e controle. A validação técnica através de testes end-to-end demonstra a robustez da solução. Os resultados esperados indicam que a automação proposta tende a eliminar retrabalho administrativo, aumentar a transparência por meio de histórico auditável e estabelecer uma base sólida para a modernização da gestão acadêmica na universidade.

Abstract. Academic monitoring is a fundamental process in Brazilian universities that allows students to develop pedagogical skills while assisting in the teaching-learning process. However, managing this process often faces challenges related to bureaucracy, lack of transparency, and multiple inefficient manual procedures. This work presents the development of Sistema de Monitoria-IC, a web platform designed to automate and simplify the lifecycle of monitoring projects at UFBA. The proposed solution digitizes the process from project creation and signing by professors to administrative approval and generation of consolidated spreadsheets for the Institute/PROGRAD. The system was developed using modern technologies such as Next.js 15, TypeScript, tRPC, PostgreSQL, and MinIO, following software engineering principles that ensure scalability and maintainability. The implemented architecture clearly separates responsibilities between a Transaction Processing System (TPS) for daily operations and managerial functionalities for analysis and control. Technical validation through end-to-end tests demonstrates the solution's robustness. Expected results indicate that the proposed automation tends to eliminate administrative rework, increase transparency through auditable history, and establish a solid foundation for modernizing academic management at the university.

Palavras-chave: Gestão Acadêmica, Sistema de Monitoria, Arquitetura de Software, Desenvolvimento Web, Automação de Processos.

Keywords: Academic Management, Monitoring System, Software Architecture, Web Development, Process Automation.

1 Introdução

A monitoria acadêmica representa um dos pilares fundamentais do ensino superior brasileiro, estabelecendo-se como uma prática pedagógica que beneficia simultaneamente monitores, estudantes e docentes. Regulamentada pela Lei nº 9.394/96 (Lei de Diretrizes e Bases da Educação Nacional), a monitoria permite que alunos com destacado desempenho acadêmico auxiliem seus pares no processo de aprendizagem, desenvolvendo competências didáticas enquanto aprofundam seus conhecimentos na disciplina Brasil [1996].

Na Universidade Federal da Bahia (UFBA), o programa de monitoria segue um fluxo complexo que envolve múltiplos atores e etapas interdependentes. O processo inicia-se com o planejamento semestral, quando a administração im-

porta dados de disciplinas e professores. Em seguida, os docentes criam e submetem projetos de monitoria que precisam ser aprovados administrativamente. Após a aprovação, ocorre a publicação de editais internos, inscrição de candidatos, processo seletivo, alocação de bolsas, e finalmente, a consolidação de dados para envio à PROGRAD (Pró-Reitoria de Graduação).

A literatura de Sistemas de Informação fornece evidências de que a digitalização e a automação de processos reduzem custos transacionais, eliminam retrabalho e aumentam a confiabilidade e a auditabilidade dos registros Laudon and Laudon [2011], Davenport [1993], Hammer and Champy [1993]. Em contextos públicos e acadêmicos, iniciativas de governo digital e transformação digital têm sido associadas a ganhos de eficiência administrativa e transparência, desde que

apoiadas por arquitetura tecnológica moderna e governança de dados adequada World Bank [2022], UNESCO [2023]. No escopo deste trabalho, tais fundamentos são adotados para sustentar a proposta de um sistema unificado de monitoria, substituindo fluxos fragmentados por um *workflow* padronizado, rastreável e escalável.

Apesar de sua importância reconhecida, a gestão dos programas de monitoria na UFBA ainda depende predominantemente de processos manuais e fragmentados. Formulários em papel, planilhas eletrônicas dispersas, comunicação via e-mail e ausência de um sistema centralizado caracterizam o cenário atual, resultando em ineficiências operacionais significativas. Essa realidade contrasta com a tendência global de digitalização e automação de processos administrativos no ambiente acadêmico. Laudon and Laudon [2011] argumentam que a tecnologia da informação constitui uma das principais ferramentas para alcançar excelência operacional.

1.1 Identificação do Problema

O processo de monitoria na UFBA é lento, caro e opaco. Do lado docente, há retrabalho sistemático (recriação de projetos a cada semestre), dispersão de documentos (planilhas, PDFs e e-mails) e ausência de trilhas de auditoria — combinação que consome horas de atividades-fim e eleva o risco de erro. A seleção permanece predominantemente manual e heterogênea entre departamentos, impedindo comparabilidade de critérios e produzindo falhas de registro.

Para estudantes, a descoberta de vagas é imprevisível e a jornada é fragmentada: múltiplos formulários, anexos por canais distintos e ausência de um painel único para acompanhar prazos, status e resultados — o que reforça a percepção de opacidade e desestimula a participação.

Administrativamente, a consolidação de dados heterogêneos gera retrabalho, dificulta a conformidade com regulamentos e prazos e fragiliza a prestação de contas. Sem um repositório transacional e analítico integrado, relatórios custam caro e análises históricas confiáveis tornam-se inviáveis para decisões como distribuição de bolsas e avaliação de efetividade de projetos.

Em síntese, há uma lacuna tecnológica clara: falta um sistema de informação *fim-a-fim* específico para monitoria que integre criação e aprovação de projetos, publicação de editais, inscrições, seleção, aceite de vagas e consolidações finais. A literatura indica que a automação e a padronização, apoiadas por arquitetura adequada e governança de dados, reduzem custos transacionais, retrabalho e erros e elevam a confiabilidade e a auditabilidade dos registros Davenport [1993], Hammer and Champy [1993], Laudon and Laudon [2011], World Bank [2022], UNESCO [2023].

1.2 Objetivos

Este trabalho documenta o desenvolvimento e implementação do Sistema de Monitoria-IC, uma plataforma Web completa para gestão dos projetos de monitoria do Instituto de Computação da UFBA. O objetivo central é substituir fluxos manuais e dispersos por um sistema único que garanta transparência, rastreabilidade e padronização de todo o processo de monitoria.

Os objetivos específicos incluem:

1. Digitalizar o ciclo completo de projetos de monitoria:

desde a criação com templates reutilizáveis, assinatura digital pelo professor, submissão, até aprovação administrativa e publicação de editais

2. Automatizar o processo seletivo: permitindo inscrições online, captura automática de notas e CR do histórico, consideração de equivalências entre disciplinas, e publicação transparente de resultados
3. Sistematizar a alocação de bolsas: com geração automática de planilhas para o Instituto, configuração do total de bolsas informado pela PROGRAD, e alocação por projeto com validações automáticas
4. Eliminar trabalhos manuais repetitivos: através de automação de notificações por e-mail, geração de documentos PDF, e integração com armazenamento de arquivos
5. Fornecer base analítica para tomada de decisões: com dashboards administrativos, APIs para integração, e relatórios consolidados para análises institucionais

1.3 Fora do Escopo

Embora o sistema cubra o ciclo de vida completo da monitoria acadêmica, algumas funcionalidades foram deliberadamente excluídas do escopo deste trabalho. Primeiro, o módulo de relatórios finais e certificados (Fase 6 do fluxo de monitoria) foi especificado nos requisitos mas não implementado, sendo priorizado como trabalho futuro devido a restrições de tempo e à necessidade de validação do fluxo principal antes de sua expansão. Segundo, a integração automática com o sistema acadêmico institucional (SIAC ou SIGAA) para captura de históricos e notas permanece manual, exigindo que estudantes anexam documentos comprobatórios durante a inscrição. Terceiro, o sistema foi desenvolvido especificamente para o Instituto de Computação da UFBA, com adaptações necessárias para expansão a outros departamentos ou instituições. Quarto, funcionalidades de analytics avançado, como sistema de recomendação para correspondência entre aluno e projeto ou previsão de desempenho com técnicas de aprendizado de máquina, foram identificadas como melhorias futuras mas não integradas nesta versão inicial. Por fim, um aplicativo móvel nativo não foi desenvolvido, embora a interface Web seja responsiva e funcione adequadamente em dispositivos móveis via navegador.

1.4 Contribuições

Este trabalho apresenta quatro contribuições principais para a área de Sistemas de Informação aplicados à gestão acadêmica. Primeira, propõe e implementa uma arquitetura em três camadas (Router-Service-Repository) específica para o domínio de monitoria acadêmica, demonstrando como separar responsabilidades entre apresentação, lógica de negócios e acesso a dados em um sistema transacional com múltiplos atores e requisitos formais de auditoria. Segunda, desenvolve um workflow automatizado completo que cobre as cinco primeiras fases do ciclo de monitoria (planejamento, aprovação, alocação, seleção e consolidação), incluindo funcionalidades inovadoras como templates reutilizáveis de projetos, assinaturas digitais integradas, consideração automática de equivalências curriculares e geração de documentos formais em PDF, resultando em significativa redução no tempo gasto em tarefas administrativas repetitivas. Terceira, documenta um estudo de caso real de transformação digital em instituição pública brasileira,

fornecendo evidências técnicas (57 testes automatizados com 100% de aprovação) e organizacionais (análise do estado da prática em dez universidades públicas) que podem orientar iniciativas similares em outras instituições de ensino superior. Quarta, disponibiliza uma solução de código aberto baseada em tecnologias modernas (Next.js 15, tRPC v11, PostgreSQL, TypeScript) com documentação técnica completa, permitindo replicação, adaptação e extensão por outras universidades que enfrentem desafios similares na gestão de programas de monitoria.

1.5 Estrutura do Trabalho

Este artigo está organizado da seguinte forma: a Seção 2 apresenta os fundamentos teóricos sobre Sistemas de Informação e monitoria acadêmica. A Seção 3 analisa trabalhos relacionados e o estado da prática em universidades brasileiras. A Seção 4 detalha o Sistema de Monitoria-IC, iniciando pela metodologia de desenvolvimento adotada (4.1), seguida pelos requisitos funcionais e não-funcionais (4.2), arquitetura em camadas (4.3), modelo de dados normalizado (4.4), fluxo de processos do ciclo completo de monitoria (4.5), stack tecnológico implementado (4.6) e interfaces especializadas por perfil de usuário (4.7). A Seção 5 apresenta a avaliação experimental, incluindo metodologia de testes, resultados técnicos obtidos e limitações identificadas. Por fim, a Seção 6 conclui o trabalho e aponta direções futuras.

2 Fundamentação Teórica

2.1 Monitoria Acadêmica

A monitoria acadêmica constitui uma modalidade de ensino-aprendizagem que contribui para a formação integrada do aluno nas atividades de ensino, pesquisa e extensão dos cursos de graduação. Conforme estabelecido pela Lei de Diretrizes e Bases da Educação Nacional (Lei nº 9.394/96), as universidades devem aproveitar estudantes de bom rendimento acadêmico em tarefas de ensino e pesquisa Brasil [1996].

O processo de monitoria na UFBA segue diretrizes institucionais que estabelecem critérios de seleção baseados no desempenho acadêmico (nota na disciplina e Coeficiente de Rendimento), definem modalidades de participação (bolsista e voluntário), delimitam responsabilidades de monitores, professores e coordenação, regulam o fluxo de aprovação de projetos e alocação de recursos e definem requisitos de certificação ao final do período.

Estudos sobre programas de monitoria demonstram benefícios significativos: desenvolvimento de habilidades didáticas nos monitores Natário and Santos [2010], melhoria no desempenho acadêmico dos estudantes assistidos Frison [2016], e apoio essencial aos docentes na condução de atividades práticas Dantas [2014]. Entretanto, esses mesmos estudos apontam desafios na gestão administrativa desses programas, especialmente relacionados à burocracia e falta de ferramentas adequadas.

2.2 Sistemas de Informação na Gestão Acadêmica

Sistemas de Processamento de Transações (SPTs) são “sistemas informatizados que realizam e registram as transações rotineiras necessárias ao funcionamento organizacional” Lau-

don and Laudon [2011]. No contexto acadêmico, gerenciam operações como matrículas, lançamento de notas, controle de frequência e, neste trabalho, o ciclo de vida de projetos de monitoria. Suas características essenciais incluem alta precisão e confiabilidade dos dados, processamento eficiente de grandes volumes transacionais, mecanismos de auditoria e rastreabilidade e alta disponibilidade para sustentar operações críticas.

Já os Sistemas de Informações Gerenciais (SIG) “resumem e relatam as operações básicas da organização usando dados fornecidos pelos SPTs” Laudon and Laudon [2011]. No Sistema de Monitoria-IC, o componente gerencial materializa-se em painéis e relatórios consolidados para acompanhar o andamento dos processos, analisar tendências históricas, auxiliar a distribuição de recursos e gerar documentos formais para órgãos superiores.

A separação de responsabilidades entre SPT (camada operacional) e SIG (camada analítica) fundamenta a arquitetura do Sistema de Monitoria-IC. O SPT gerencia o ciclo de vida de cada transação — cadastros, aprovações, inscrições — enquanto o SIG consome registros consolidados para análises e relatórios. Essa divisão favorece escalabilidade independente, facilita manutenção e evolução, reduz acoplamento entre operações e análises e permite otimizações específicas de desempenho em cada camada.

3 Trabalhos Relacionados

3.1 Levantamento do Estado da Prática

Para avaliar o estado atual da gestão de monitoria em universidades brasileiras, foi realizado um levantamento documental nos cursos de Computação das dez universidades públicas mais bem classificadas segundo o Ranking Universitário Fólio (RUF) 2024 de S.Paulo [2024]. A análise concentrou-se em fontes públicas: páginas institucionais dos sistemas acadêmicos, editais de monitoria e documentos de orientação disponíveis em 2024–2025. Não houve acesso a sistemas internos autenticados, o que impõe limites importantes à generalização dos resultados. A Tabela 1 sintetiza os resultados desse levantamento.

Com base nesse levantamento, não foi possível confirmar a existência de um sistema específico e completo de monitoria em nenhuma das instituições analisadas apenas a partir de fontes públicas. Em vários casos, há evidência de sistemas acadêmicos consolidados (como Júpiter, SIGA, SIGAA e CAGR) que oferecem algum suporte ao processo, mas as descrições dos editais e das orientações complementares indicam que etapas críticas — como submissão de projetos, consolidação de inscrições, definição de bolsas e emissão de relatórios — permanecem distribuídas entre formulários eletrônicos genéricos, documentos administrativos e fluxos manuais conduzidos por departamentos.

Outro padrão observado é a fragmentação dos canais de interação: o estudante frequentemente acessa uma combinação de páginas institucionais, formulários externos e comunicados por e-mail para compreender o processo de monitoria, enquanto professores e coordenações dependem de planilhas, documentos anexados em sistemas de processo eletrônico e rotinas internas de consolidação. Mesmo nos casos em que há módulos de “atividades acadêmicas” ou programas de

apoio didático, a documentação pública não descreve workflows fim-a-fim de monitoria equivalentes ao ciclo completo apresentado na Seção 4.

Essas limitações metodológicas são importantes: o levantamento baseia-se apenas em evidências disponíveis publicamente, em um recorte temporal específico (2024–2025) e focado nos cursos de Computação. É possível que existam soluções internas não documentadas, iniciativas departamentais em fase piloto ou sistemas legados de uso restrito que não foram capturados nessa análise. Ainda assim, o panorama obtido é consistente com a literatura sobre transformação digital na gestão acadêmica Laudon and Laudon [2011], Davenport [1993], Hammer and Champy [1993], World Bank [2022], UNESCO [2023], que aponta para a coexistência de sistemas institucionais robustos com processos específicos de alto teor manual e fragmentado.

À luz desse cenário, o Sistema de Monitoria-IC distingue-se por cinco dimensões principais. Primeiro, é específico para o workflow de monitoria, ao invés de reutilizar módulos genéricos concebidos para múltiplos processos acadêmicos. Segundo, cobre as etapas centrais do ciclo de vida, desde a criação e aprovação de projetos até a consolidação final para a PROGRAD, reduzindo a fragmentação observada nas instituições analisadas. Terceiro, automatiza etapas críticas via integrações com armazenamento de arquivos, geração de PDFs, assinaturas digitais e notificações, diminuindo significativamente a necessidade de retrabalho manual. Quarto, adota um stack tecnológico moderno (Next.js 15, tRPC v11, PostgreSQL, MinIO) com foco em experiência do usuário, *type-safety* e desempenho, em contraste com soluções frequentemente baseadas em sistemas legados e formulários genéricos. Por fim, garante transparência e rastreabilidade através de histórico auditável e acesso diferenciado por papéis (admin, professor, aluno, chefe de departamento), alinhando-se às recomendações de boas práticas em governo digital e governança de dados em educação superior.

3.2 Projetos Correlatos no Instituto de Computação

No contexto do próprio Instituto de Computação da UFBA, há Trabalhos de Conclusão de Curso recentes que tratam de sistemas Web para automatização de processos acadêmico-administrativos, oferecendo um referencial importante para situar o Sistema de Monitoria-IC. O trabalho “IC-Requests: um sistema sensível ao contexto para suporte a solicitações acadêmicas”, de Machado [2023], propõe um sistema Web para gerenciar requerimentos acadêmicos de alunos do IC-UFBA. Em termos de escopo, o IC-Requests foca em um processo mais delimitado — solicitações específicas como trancamentos, declarações e ajustes de matrícula — enquanto o Sistema de Monitoria-IC cobre o ciclo de vida completo da monitoria acadêmica, desde a criação de projetos pelos docentes até a emissão de relatórios finais e consolidação para a PROGRAD. Metodologicamente, o IC-Requests dá ênfase à avaliação empírica com usuários, incluindo questionários de usabilidade com alunos, ao passo que o Monitoria-IC privilegia o detalhamento arquitetural e a robustez técnica, com testes automatizados cobrindo o fluxo transacional. As duas abordagens são complementares: o IC-Requests contribui com uma validação centrada no usuário, e o Sistema

de Monitoria-IC aprofunda a engenharia de software em um domínio mais amplo e complexo.

Outro projeto relevante é “AlertUFBA: um sistema Web de orientações sobre segurança e emergência no contexto universitário”, de Santos [2023], que desenvolve uma plataforma informativa para centralizar orientações em caso de emergências no campus. Embora o domínio seja distinto do da monitoria, há semelhanças metodológicas na condução de entrevistas e questionários com usuários para validar a utilidade do sistema. Em comparação, o AlertUFBA atua sobretudo como um canal de comunicação e prevenção, enquanto o Sistema de Monitoria-IC implementa um fluxo transacional estruturado, com múltiplos atores e requisitos formais de prestação de contas institucional. Assim, o Monitoria-IC posiciona-se mais próximo de um Sistema de Processamento de Transações (SPT) integrado a funcionalidades gerenciais, conforme discutido na Seção 2. Em síntese, ambos os trabalhos do IC-UFBA contribuem metodologicamente com avaliação de usabilidade e validação empírica, enquanto o Sistema de Monitoria-IC se diferencia pelo escopo mais abrangente (ciclo completo de monitoria versus processos específicos), pela arquitetura robusta em três camadas (Router-Service-Repository) e pela natureza transacional com múltiplos atores interdependentes, caracterizando um sistema de informação empresarial completo ao invés de soluções pontuais ou informativas.

3.3 Comparação com Sistemas Acadêmicos Existentes

Além dos trabalhos de TCC, diversas universidades brasileiras utilizam sistemas acadêmicos integrados, como JúpiterWeb (USP), SIGA (UFRJ), SIGAA (UnB), CAGR (UFSC) e SEI (UNIFESP), que oferecem módulos genéricos para gestão de disciplinas, matrículas e processos administrativos. A análise documental realizada para este trabalho, baseada em páginas institucionais e editais públicos, não identificou, entretanto, um módulo específico e completo de monitoria acadêmica que cubra de forma integrada todas as etapas descritas na Seção 4. A Tabela 2 resume, de forma sintética, o posicionamento do Sistema de Monitoria-IC em relação a esses sistemas genéricos, evidenciando as diferenças de escopo, suporte à monitoria e avaliação documentada.

À luz dessa comparação, o Sistema de Monitoria-IC distingue-se por ser um sistema específico para monitoria acadêmica, cobrindo o ciclo de vida completo do processo e integrando funcionalidades transacionais e gerenciais em uma única plataforma. A solução proposta não pretende substituir sistemas acadêmicos institucionais existentes, mas sim complementar a infraestrutura da UFBA com um módulo especializado, passível de integração futura com sistemas como SIAC ou SIGAA.

4 Sistema de Monitoria-IC

4.1 Metodologia

O desenvolvimento do Sistema de Monitoria-IC seguiu uma metodologia estruturada que combina práticas consolidadas de engenharia de software com adaptações ao contexto acadêmico específico da UFBA. O processo abrangeu quatro fases principais: análise de requisitos, design e arquitetura, implementação e validação técnica.

A identificação de requisitos foi conduzida através de entrevistas semiestruturadas com coordenadores do programa de monitoria do Instituto de Computação, professores com histórico de participação em projetos de monitoria e administradores responsáveis pela gestão do processo. Essas entrevistas, realizadas ao longo de dois meses, permitiram compreender as necessidades de cada perfil de usuário, identificar pontos de dor no processo manual existente e capturar expectativas quanto à automação. Complementarmente, realizou-se análise documental de editais de monitoria de semestres anteriores (2019 a 2023), normas institucionais da PROGRAD e relatórios anuais do programa para compreender o ciclo completo de monitoria, identificar pontos críticos de ineficiência e documentar as regras de negócio vigentes. Essa combinação de técnicas qualitativas (entrevistas) e análise documental proporcionou uma visão abrangente dos requisitos, resultando em um documento de especificação estruturado em funcionais (RF) e não-funcionais (RNF), conforme apresentado na Tabela 3.

Com os requisitos consolidados, foi projetada uma arquitetura em camadas que separa responsabilidades entre apresentação, aplicação, negócio e dados, conforme discutido em detalhes na Seção 4. O design arquitetural baseou-se em três princípios fundamentais: separação clara de responsabilidades entre as quatro camadas, garantindo baixo acoplamento e alta coesão; *type-safety* completa em toda a stack através de TypeScript e tRPC, eliminando classes inteiras de erros em tempo de desenvolvimento; e modularidade estrutural para facilitar manutenção, testes e evolução futura do sistema. O modelo de dados foi normalizado conforme as formas normais relacionais clássicas, mantendo integridade referencial entre todas as entidades e garantindo rastreabilidade completa de todas as transações através de timestamps e histórico de alterações. A escolha de tecnologias priorizou maturidade e adoção na indústria, qualidade da documentação e suporte da comunidade, e alinhamento direto com os requisitos não-funcionais de desempenho, escalabilidade e segurança especificados.

O desenvolvimento seguiu práticas ágeis em ciclos iterativos curtos (duas semanas), com entregas incrementais testadas e validadas ao final de cada iteração. A implementação foi organizada por domínios de negócio (projetos, editais, inscrições, seleção, relatórios), permitindo desenvolvimento paralelo de funcionalidades independentes e reduzindo significativamente o acoplamento entre módulos. Cada domínio foi desenvolvido seguindo o padrão de três camadas (router, service, repository), facilitando testes isolados e evolução incremental. Para assegurar qualidade e robustez do código produzido, foram adotadas práticas rigorosas de engenharia de software: revisão sistemática de código (*code review*) para todas as mudanças antes de integração; linting e formatação automática com Biome, garantindo consistência estilística; verificação estática de tipos em tempo de compilação com TypeScript em modo estrito; testes unitários focados em lógica de negócio crítica (seleção de candidatos, alocação de bolsas, validações); testes *end-to-end* (E2E) cobrindo fluxos completos de monitoria com Playwright; e pipeline de integração contínua no GitHub Actions executando automaticamente testes, build e validações a cada commit.

A validação técnica da solução concentrou-se em duas frentes complementares. Na primeira frente, testes automati-

zados foram implementados em três níveis: cenários E2E com Playwright, cobrindo os papéis de administrador, professor e estudante em fluxos completos de monitoria; testes unitários e de integração com Vitest, validando regras de negócio críticas (critérios de seleção, equivalências de disciplinas, limites de bolsas) e interações corretas entre as camadas de aplicação, negócio e dados; e testes específicos de segurança e autorização, verificando isolamento adequado entre papéis e proteção de rotas sensíveis. A utilização de um ORM tipado (Drizzle) facilitou significativamente a detecção precoce de inconsistências entre o modelo de dados e o código da aplicação. Na segunda frente, foi realizada revisão sistemática de conformidade: verificação da aderência aos 15 requisitos funcionais especificados, validação dos 14 requisitos não-funcionais (métricas de desempenho, políticas de segurança, requisitos de usabilidade) e análise de conformidade com normas institucionais de monitoria da UFBA. Os resultados dessa validação técnica são apresentados em detalhes na Seção 5. Cabe ressaltar que, até o momento da redação deste trabalho, não foi realizada uma avaliação empírica com usuários finais em ambiente de produção real. Essa etapa, fundamental para validar a usabilidade percebida e o impacto organizacional efetivo, é considerada prioritária nos trabalhos futuros, conforme discutido na Seção 6.

4.2 Requisitos

O Sistema de Monitoria-IC foi especificado a partir de requisitos funcionais e não-funcionais identificados através de entrevistas com coordenadores, professores e análise do processo manual existente. A Tabela 3 apresenta os principais requisitos do sistema, categorizados por tipo.

A Tabela 3 sintetiza os requisitos levantados durante a fase de especificação. Esses requisitos orientaram as decisões arquiteturais e tecnológicas tomadas no desenvolvimento do sistema, estabelecendo critérios claros de sucesso para funcionalidades, desempenho, segurança e manutenibilidade. Com base nessa especificação, o sistema foi projetado seguindo uma arquitetura em camadas que separa claramente as responsabilidades, conforme descrito a seguir.

4.3 Arquitetura

O Sistema de Monitoria-IC foi projetado seguindo uma arquitetura em camadas que separa claramente as responsabilidades e promove manutenibilidade e escalabilidade, conforme ilustrado na Figura 1. A arquitetura segue o padrão de três camadas com componentes especializados. Na implementação prática, as camadas de Apresentação (cliente React) e Aplicação (rotas tRPC) coexistem na mesma aplicação Next.js (padrão full-stack monolítico), enquanto a separação lógica em quatro camadas apresentada facilita a compreensão didática das responsabilidades de cada componente arquitetural:

A arquitetura em camadas adotada confere ao sistema benefícios diretos em termos de manutenibilidade, testabilidade e evolução incremental. A separação clara de responsabilidades permite que modificações em uma camada sejam realizadas com impacto mínimo nas demais, facilitando tanto correções pontuais quanto a introdução de novas funcionalidades sem comprometer a estabilidade do sistema em produção. A independência entre camadas viabiliza testes unitários isolados de cada componente (routers tRPC, serviços de negócio,

repositórios de dados), contribuindo para uma base de código mais robusta e confiável. Além disso, a modularização arquitetural favorece o desenvolvimento paralelo por equipes distintas: designers de interface podem atuar na camada de apresentação enquanto engenheiros de backend refinam serviços de negócio ou otimizam consultas na camada de dados, sem conflitos de responsabilidade ou acoplamento excessivo entre artefatos.

A comunicação entre camadas segue padrões bem estabelecidos que asseguram consistência e rastreabilidade das operações. O cliente React consome endpoints tRPC que, internamente, invocam serviços de negócio responsáveis por orquestrar chamadas a repositórios e a APIs externas (envio de e-mails via Nodemailer, armazenamento de arquivos no MinIO). Os dados trafegam sempre validados por esquemas Zod compartilhados entre cliente e servidor, eliminando divergências de tipo e prevenindo inconsistências. Erros originados em qualquer camada (validação de entrada, regras de negócio, falhas de persistência) são propagados de forma estruturada até o cliente, que apresenta mensagens adequadas ao contexto de uso. A arquitetura ainda contempla pontos de extensão planejados para integrações futuras, como autenticação via CAS (Central Authentication Service) da universidade, sincronização automática de dados acadêmicos com sistemas legados e exportação de métricas para ferramentas de business intelligence externas, garantindo que o sistema possa crescer em funcionalidade sem necessidade de reestruturação fundamental.

Do ponto de vista de escalabilidade, a arquitetura permite crescimento horizontal tanto na camada de aplicação quanto na de dados. O uso de containers Docker facilita a replicação de instâncias da aplicação atrás de um平衡ador de carga, enquanto o PostgreSQL suporta réplicas de leitura para distribuir consultas pesadas. O MinIO pode ser configurado em modo distribuído para alta disponibilidade de arquivos. Essa flexibilidade arquitetural garante que o sistema possa atender a demandas crescentes sem redesenho significativo. A separação entre lógica de negócio e persistência também facilita a substituição de componentes específicos: o banco de dados poderia ser migrado para soluções gerenciadas em nuvem (como Amazon RDS ou Google Cloud SQL) sem alterações no código da aplicação, e o armazenamento de objetos poderia transicionar para Amazon S3 ou Google Cloud Storage mantendo compatibilidade via protocolo S3.

Do ponto de vista operacional, o sistema foi projetado para facilitar monitoramento e diagnóstico em ambiente de produção. Os logs estruturados em formato JSON permitem integração com ferramentas de observabilidade como Elasticsearch, Grafana ou Datadog, possibilitando alertas automáticos em caso de degradação de performance ou erros recorrentes. As métricas de uso (requisições por segundo, tempo de resposta, taxa de erros) podem ser exportadas via endpoints dedicados, alimentando dashboards operacionais que auxiliam a equipe de infraestrutura na identificação proativa de gargalos. A estratégia de cache implementada no cliente (via React Query) reduz significativamente a carga sobre o servidor, enquanto índices otimizados no banco de dados garantem tempos de resposta consistentes mesmo sob picos de demanda característicos de períodos de inscrição.

A arquitetura contempla ainda mecanismos de resiliência

que minimizam o impacto de falhas parciais. Transações de banco de dados são utilizadas para garantir atomicidade em operações críticas, como a submissão de projetos e a alocação de bolsas, evitando estados inconsistentes mesmo em caso de interrupções inesperadas. O sistema implementa padrões de *retry* com backoff exponencial para chamadas a serviços externos (envio de e-mails, armazenamento de arquivos), de modo que falhas transitórias não resultem em perda de dados ou operações incompletas. Filas de processamento assíncrono permitem que tarefas demoradas (geração de relatórios, envio em lote de notificações) sejam executadas em segundo plano sem bloquear a interface do usuário, melhorando a experiência percebida e a responsividade geral do sistema.

O tratamento de erros segue uma estratégia hierárquica que propaga exceções de forma controlada através das camadas arquiteturais. Erros de validação detectados na camada de apresentação são exibidos imediatamente ao usuário com mensagens contextualizadas, enquanto falhas de regras de negócio na camada de serviços são transformadas em códigos de erro padronizados que o cliente interpreta para apresentar feedback apropriado. Exceções não previstas são capturadas por handlers globais que registram informações diagnósticas detalhadas (stack trace, contexto da requisição, estado do usuário) antes de retornar respostas genéricas que não expõem detalhes internos da implementação, preservando a segurança do sistema.

A gestão de configurações adota o princípio de separação entre código e ambiente, utilizando variáveis de ambiente para parâmetros que variam entre ambientes de desenvolvimento, homologação e produção. Credenciais sensíveis (chaves de API, senhas de banco de dados, tokens de autenticação) são injetadas em tempo de execução através de gerenciadores de segredos (*secrets managers*) ou arquivos de ambiente protegidos, nunca sendo armazenadas no repositório de código. Essa abordagem facilita a implantação em diferentes infraestruturas e permite que a mesma imagem Docker seja promovida entre ambientes sem necessidade de recompilação, seguindo práticas recomendadas de DevOps e integração contínua.

O versionamento da API segue convenções semânticas que permitem evolução gradual sem quebrar clientes existentes. Mudanças compatíveis (adição de campos opcionais, novos endpoints) são incorporadas transparentemente, enquanto alterações incompatíveis (remoção de campos, mudança de tipos) são sinalizadas através de mecanismos de depreciação que concedem período de transição aos consumidores. A documentação da API é gerada automaticamente a partir dos esquemas Zod e das definições tRPC, garantindo que a especificação técnica permaneça sincronizada com a implementação real e reduzindo o risco de divergências entre documentação e comportamento efetivo do sistema.

Tabela 1. Panorama sintético da gestão de monitoria em universidades públicas brasileiras, a partir de fontes públicas (sítios institucionais e editais).

Fonte: páginas oficiais dos sistemas Júpiter/USP Universidade de São Paulo [2024], SIGA/UFRJ Universidade Federal do Rio de Janeiro [2024], SIGAA/UnB Universidade de Brasília [2024], CAGR/UFSC Universidade Federal de Santa Catarina [2024] e SEI/UNIFESP Universidade Federal de São Paulo [2024], complementadas por editais e orientações públicas (2024–2025).

Universidade	Uso de sistema acadêmico	Evidências de ferramentas	Observações a partir das fontes públicas
USP	Sim	Júpiter + formulários complementares	Monitoria relacionada ao sistema acadêmico geral, com seleção apoiada em formulários e orientações departamentais.
Unicamp	Parcial	Páginas do PAD e editais locais	Documentação pública menciona critérios e programas (por exemplo, PAD), mas não detalha mecanismo único e integrado de inscrição para todos os cursos.
UFRGS	Sim	Portal do Aluno	Portal acadêmico institucional com funcionalidades para atividades acadêmicas; editais indicam processos complementares por unidade.
UFRJ	Sim	SIGA + documentos administrativos	Parte do cadastro ocorre no SIGA; editais descrevem etapas adicionais conduzidas por departamentos e comissões.
UFMG	Parcial	Editais e formulários online	Editais indicam uso de formulários eletrônicos simples para inscrição, com consolidações posteriores em planilhas e documentos administrativos.
UNESP	Parcial	Editais + documentação interna	Não foram localizados, nas fontes públicas, módulos específicos de monitoria; editais sugerem trâmites administrativos distribuídos entre unidades.
UFSC	Sim	CAGR	Sistema acadêmico com módulo genérico; as especificidades da monitoria são tratadas em editais e orientações complementares.
UnB	Sim	SIGAA	SIGAA oferece módulo para atividades acadêmicas; editais indicam que a monitoria utiliza esse módulo em combinação com procedimentos departamentais.
UNIFESP	Parcial	SEI + formulários	Processos administrativos tramitam via SEI, com inscrições tipicamente apoiadas em formulários e documentos anexos.
UFPR	Parcial	Editais + formulários online	Editais recentes mencionam uso de formulários (como Google/Microsoft Forms) e trâmites eletrônicos departamentais, sem referência a módulo dedicado de monitoria.

Tabela 2. Comparação sintética entre sistemas acadêmicos genéricos e o Sistema de Monitoria-IC.

Sistema	Escopo principal	Supporte à monitoria	Workflow de monitoria fim-a-fim	Avaliação com usuários (nas fontes públicas)
JúpiterWeb (USP)	Gestão acadêmica geral (grades, matrículas, histórico)	Mencionado em editais e orientações complementares, sem módulo dedicado descrito publicamente	Não identificado em fontes públicas; etapas de monitoria aparentam depender de formulários e fluxos departamentais	Não documentada explicitamente
SIGA (UFRJ)	Sistema integrado de gestão acadêmica	Utilizado em combinação com documentos administrativos e comissões departamentais para monitoria	Não identificado módulo específico que integre todas as fases da monitoria	Não documentada explicitamente
SIGAA (UnB)	Gestão de atividades acadêmicas e administrativas	Oferece suporte a atividades acadêmicas em geral; a monitoria é tratada em conjunto com procedimentos departamentais	Não há descrição pública de um workflow completo de monitoria como o proposto neste trabalho	Não documentada explicitamente
CAGR (UFSC)	Controle acadêmico da graduação	Possui funcionalidades de apoio; monitoria complementada por editais e orientações externas ao sistema	Não identificado módulo fim-a-fim de monitoria com automação de relatórios e consolidação	Não documentada explicitamente
SEI (UNIFESP)	Tramitação eletrônica de processos	Utilizado para gestão de documentos administrativos, incluindo etapas de monitoria	Focado em processos administrativos; o fluxo de monitoria depende de formulários e anexos geridos pelos departamentos	Não documentada explicitamente
Sistema de Monitoria-IC	Workflow completo de monitoria (projetos, bolsas, inscrições, seleção, consolidação)	Módulo específico para monitoria, concebido para o contexto da UFBA	Parcial: cobre integralmente as fases 1 a 5; relatórios/certificados foram especificados como evolução	Avaliação técnica por testes automatizados; avaliação com usuários prevista como trabalho futuro

Tabela 3. Requisitos funcionais e não-funcionais do Sistema de Monitoria-IC.

ID	Tipo	Descrição
Requisitos Funcionais		
RF01	Funcional	Importar planejamento semestral (Excel) com disciplinas e professores (SIAPE)
RF02	Funcional	Criar e reutilizar templates de projetos entre semestres
RF03	Funcional	Workflow de aprovação com estados (DRAFT, SUBMITTED, APPROVED, REJECTED)
RF04	Funcional	Assinatura digital de projetos por professores e chefe de departamento
RF05	Funcional	Alocar bolsas por projeto com validação de limites totais
RF06	Funcional	Gerar e publicar edital interno com assinatura digital
RF07	Funcional	Portal de inscrições para estudantes com upload de documentos
RF08	Funcional	Captura automática de CR e notas do histórico acadêmico
RF09	Funcional	Considerar equivalências entre disciplinas na seleção
RF10	Funcional	Interface para professores avaliarem e selecionarem candidatos
RF11	Funcional	Portal para aceite/rejeição de vagas e coleta de dados bancários
RF12	Funcional	Gerar planilhas consolidadas para PROGRAD com dados de monitores
RF13*	Funcional	Emitir relatórios finais e certificados de monitoria
RF14	Funcional	Notificações automáticas por e-mail em todas as etapas
RF15	Funcional	Dashboard administrativo com métricas e indicadores
*RF13 implementado: relatórios finais e certificados completos (código pronto, validação com usuários pendente).		
Requisitos Não-Funcionais		
RNF01	Desempenho	Tempo de carregamento inicial < 3s (First Contentful Paint)
RNF02	Desempenho	Latência de API < 100ms para 95% das requisições
RNF03	Escalabilidade	Suportar 1000+ usuários concorrentes sem degradação
RNF04	Segurança	Autenticação via Lucia Auth com sessões de 30 dias
RNF05	Segurança	Controle de acesso baseado em papéis (RBAC)
RNF06	Segurança	Validação de entrada com Zod em cliente e servidor
RNF07	Confiabilidade	Disponibilidade > 99% em horário comercial
RNF08	Confiabilidade	Backup incremental a cada 6 horas com retenção de 30 dias
RNF09	Usabilidade	Interface responsiva para desktop, tablet e mobile
RNF10	Usabilidade	Acessibilidade WCAG 2.1 nível AA
RNF11	Manutenibilidade	Type-safety completo com TypeScript em todo codebase
RNF12	Manutenibilidade	Cobertura de testes > 85% em lógica de negócio
RNF13	Interoperabilidade	API REST e tRPC para integração com sistemas externos
RNF14	Portabilidade	Containerização com Docker para deploy em qualquer ambiente

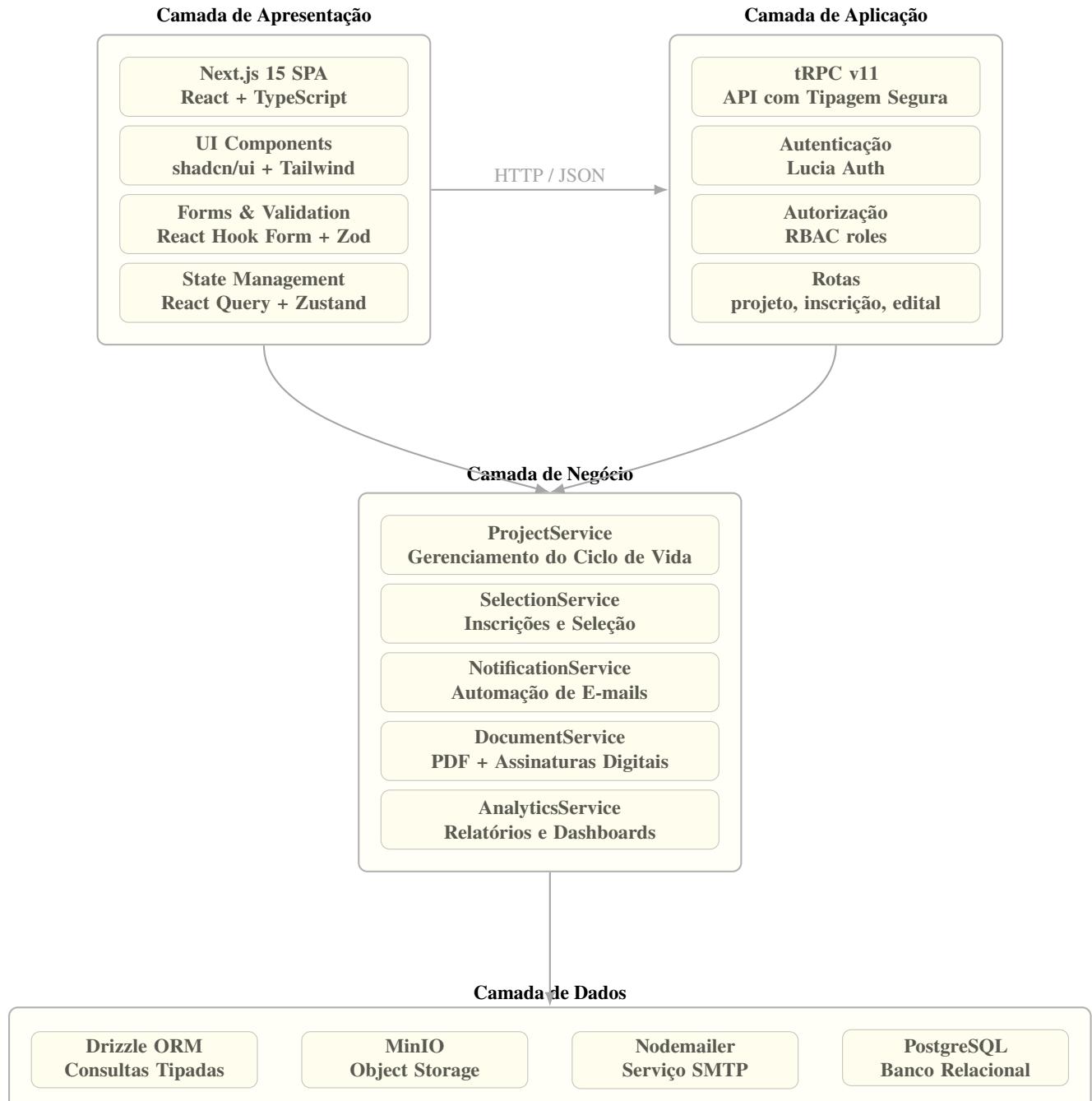


Figura 1. Arquitetura lógica do Sistema de Monitoria-IC.

A camada de apresentação foi implementada como uma Single Page Application (SPA) utilizando Next.js 15, React e TypeScript, proporcionando uma interface responsiva construída a partir de componentes shadcn/ui e Tailwind CSS. O roteamento é dinâmico e protegido por papéis, assegurando que cada perfil de usuário (administrador, professor ou estudante) tenha acesso apenas às funcionalidades pertinentes. O estado do cliente é coordenado por React Query, que gerencia cache e sincronização de dados remotos, e Zustand, que cuida do estado local da aplicação. Os formulários utilizam React Hook Form para gerenciamento eficiente de estado e validações, combinado com Zod para garantir consistência entre as validações de cliente e servidor.

O sistema oferece interfaces especializadas estruturadas em módulos funcionais para cada perfil de usuário. Para o perfil de administrador, são disponibilizados sete módulos principais que refletem as responsabilidades de coordenação do programa de monitoria. O módulo Dashboard consolida métricas operacionais e indicadores em tempo real, incluindo distribuição de projetos por status (rascunho, submetido, aprovado, rejeitado), número total de inscrições por período letivo e visualização rápida de prazos críticos. O módulo Projetos centraliza o gerenciamento completo do ciclo de vida de projetos de monitoria: visualização e aprovação de projetos submetidos pelos docentes, importação de planejamento semestral via planilha Excel com dados SIAPE (Sistema Integrado de Administração de Pessoal), e alocação de bolsas por projeto com validação automática de limites totais informados pela PROGRAD, impedindo excedentes. O módulo Editais permite criar, configurar e publicar editais internos do departamento, definindo datas de inscrição, pontos de prova e bibliografia, além de solicitar assinatura digital do chefe do departamento antes da publicação, com envio automatizado de notificações por e-mail. O módulo Documentos oferece gestão centralizada de arquivos, incluindo históricos acadêmicos, comprovantes de matrícula, documentos de identificação e PDFs gerados pelo sistema (projetos assinados, editais, atas), com suporte a versionamento e metadados para auditoria. O módulo Usuários implementa funcionalidades CRUD completas para administração de contas de professores e alunos, permitindo pesquisa avançada por nome, matrícula ou SIAPE, edição de perfis acadêmicos e envio de convites institucionais para novos docentes. O módulo Configurações abrange o cadastro e manutenção de cursos de graduação, departamentos acadêmicos, disciplinas com códigos oficiais, configuração de equivalências curriculares (fundamental para o processo seletivo que considera disciplinas equivalentes ao calcular notas), e parâmetros de integração SMTP para envio automatizado de notificações. Por fim, o módulo Sistema consolida funcionalidades analíticas e gerenciais: analytics com visualizações de tendências históricas (evolução de inscrições, distribuição de bolsas por departamento ao longo dos semestres), relatórios consolidados estruturados para PROGRAD (listagem de monitores ativos com dados de pagamento) e planilhas finais em formato Excel diferenciando bolsistas e voluntários, com links para PDFs dos projetos aprovados.

Para o perfil de professor, o sistema estrutura-se em quatro módulos que cobrem as responsabilidades docentes no programa de monitoria. O módulo Meus Projetos permite criar, editar e gerenciar projetos individuais ou coletivos (quando

múltiplos professores ministram uma mesma disciplina), com funcionalidade de reutilização de templates de semestres anteriores para agilizar a criação, edição de título, descrição, atividades e carga horária, e assinatura digital com visualização prévia do PDF antes da submissão administrativa. O módulo Processo Seletivo concentra todas as etapas relacionadas à seleção de monitores: gerenciar candidatos permite visualizar lista de alunos inscritos em cada projeto com informações acadêmicas (CR, nota na disciplina considerando equivalências configuradas), documentos anexados (histórico, comprovante de matrícula) e tipo de vaga pretendida (bolsista, voluntário ou ambos); avaliar candidatos oferece interface para atribuição de notas de prova e entrevista, com cálculo automático da nota final ponderada conforme critérios institucionais; selecionar monitores apresenta lista ranqueada de candidatos por nota final, permitindo escolha de bolsistas até o limite alocado pela administração e definição de voluntários até o limite configurado pelo próprio professor; e publicar resultados gera automaticamente atas de seleção em PDF e dispara notificações por e-mail para todos os candidatos, informando aprovação (bolsista ou voluntário), lista de espera ou não seleção. O módulo Documentos permite gerar e assinar atas de seleção (documento oficial registrando o processo seletivo, critérios utilizados, candidatos inscritos e resultado final) e termos de compromisso (documento assinado digitalmente pelo professor e pelo monitor, estabelecendo responsabilidades, carga horária e atividades previstas). Por fim, o módulo Gestão Acadêmica oferece configuração e acompanhamento de monitores voluntários adicionais além dos definidos no edital, permitindo que o professor amplie sua equipe de monitoria conforme demanda da disciplina, embora sem remuneração institucional.

Para o perfil de estudante, o sistema disponibiliza um módulo unificado de Monitoria com quatro funcionalidades principais que cobrem a jornada completa do aluno. A funcionalidade Inscrição em Monitoria permite que o estudante se inscreva em projetos disponíveis no período ativo de inscrições, informando preferência por vaga bolsista, voluntária ou ambas, com upload obrigatório de histórico acadêmico atualizado, comprovante de matrícula vigente e documento de identificação. A funcionalidade Vagas Disponíveis apresenta catálogo paginado de projetos com vagas abertas, com filtros por departamento acadêmico (DCC, DCI), tipo de vaga (bolsista, voluntário) e busca textual por título de disciplina ou nome do professor orientador, exibindo para cada projeto: nome da disciplina, código oficial, professores responsáveis, número de vagas bolsistas e voluntárias disponíveis, carga horária semanal e descrição das atividades previstas. A funcionalidade Resultados das Seleções exibe lista de todas as inscrições realizadas pelo aluno no período letivo atual, indicando para cada uma o status detalhado (selecionado bolsista, selecionado voluntário, lista de espera com posição, ou não selecionado), com resumo agregado por categoria no topo da página e notificações automáticas quando novos resultados são publicados pelos professores. Por fim, a funcionalidade Meu Status consolida informações de aceite ou rejeição de vagas oferecidas, com formulário para preenchimento de dados bancários quando o aluno aceita uma vaga bolsista (banco, agência, número da conta e dígito verificador), além de indicadores visuais de pendências que possam impedir a efetivação

da monitoria (dados bancários incompletos, documentação faltante ou termo de compromisso não assinado).

A camada de aplicação utiliza tRPC v11 para prover uma API completamente *type-safe* entre cliente e servidor, eliminando a necessidade de definições duplicadas de tipos e reduzindo significativamente a possibilidade de erros de integração. As procedures são organizadas por domínio (projeto, inscrição, edital, entre outros), facilitando a manutenção e evolução do sistema. A autenticação é realizada via Lucia Auth, um framework moderno e seguro que gerencia sessões de usuário com duração configurável de 30 dias. As validações ocorrem automaticamente em ambos os lados da comunicação através de esquemas Zod compartilhados, e o sistema oferece suporte a *subscriptions* para atualizações em tempo quase real, permitindo que múltiplos usuários visualizem mudanças de estado instantaneamente.

A camada de negócio concentra a lógica central do sistema em serviços especializados e desacoplados. O ProjectService gerencia o ciclo de vida completo de projetos de monitoria, desde a criação com base em templates até a consolidação final para PROGRAD. O SelectionService processa as inscrições de estudantes e executa a lógica de seleção, considerando critérios como coeficiente de rendimento, nota na disciplina e equivalências curriculares. O NotificationService orquestra o envio de notificações por e-mail em todas as etapas do processo, garantindo que professores, estudantes e administradores sejam informados sobre mudanças de estado relevantes. O DocumentService é responsável pela geração de documentos PDF (projetos, editais, relatórios) e pelo gerenciamento de assinaturas digitais, tanto dos professores quanto do chefe do departamento. Por fim, o AnalyticsService consolida dados transacionais em métricas e indicadores apresentados nos dashboards administrativos, fornecendo visões agregadas do processo de monitoria para apoiar a tomada de decisões institucionais.

A camada de dados utiliza PostgreSQL como sistema gerenciador de banco de dados relacional, combinado com Drizzle ORM para garantir consultas tipadas e seguras. O esquema de dados foi normalizado seguindo as formas normais clássicas, mantendo integridade referencial entre todas as entidades do domínio. As migrações são versionadas e reversíveis, permitindo evolução controlada do modelo de dados ao longo do tempo. Índices foram criados estrategicamente nas colunas mais consultadas para otimizar o desempenho de queries frequentes, especialmente aquelas relacionadas à listagem de projetos por período, busca de inscrições por estudante e geração de relatórios consolidados. Para armazenamento de arquivos (históricos acadêmicos, documentos comprobatórios, PDFs gerados), o sistema utiliza MinIO, uma solução compatível com o protocolo S3 que oferece escalabilidade horizontal e facilita futuras migrações para serviços de nuvem. Rotinas automatizadas de backup incremental são executadas a cada seis horas, com retenção de 30 dias, assegurando resiliência e possibilidade de recuperação em caso de falhas.

Além dos aspectos funcionais e arquiteturais, o Sistema de Monitoria-IC foi concebido considerando requisitos de segurança da informação e privacidade de dados, em consonância com a Lei Geral de Proteção de Dados (LGPD). A autenticação é realizada por meio de Lucia Auth, com

sessões protegidas e gerenciamento explícito de papéis (administração, professor, estudante), implementando um modelo de controle de acesso baseado em papéis (RBAC, do inglês *Role-Based Access Control*) que restringe o acesso a dados e funcionalidades de acordo com o perfil do usuário. O uso de um ORM tipado (Drizzle) contribui para mitigar riscos de injeção de SQL, enquanto a comunicação cliente-servidor é projetada para operar sobre HTTPS em ambiente de produção, assegurando a confidencialidade dos dados em trânsito. Do ponto de vista de privacidade, o sistema adota o princípio da minimização de dados, armazenando apenas informações pessoais estritamente necessárias para a gestão da monitoria (por exemplo, identificação do aluno, histórico acadêmico relevante e dados bancários de bolsistas). Políticas de retenção de dados e de acesso a relatórios podem ser configuradas para atender às diretrizes institucionais, e o histórico de operações mantém trilhas de auditoria para fins de transparência e responsabilização. A interface Web foi desenvolvida com responsividade, visando funcionamento adequado em diferentes tamanhos de tela (desktop, tablet e dispositivos móveis) e utilizando componentes que suportam navegação por teclado. Embora uma avaliação formal de acessibilidade ainda não tenha sido conduzida, o sistema segue boas práticas básicas de design de interface (uso consistente de hierarquia visual, contrastes adequados e feedback claro de ações), abrindo caminho para evoluções futuras em direção à conformidade com as diretrizes WCAG 2.1 em nível institucional.

4.4 Modelo de Dados

O modelo de dados foi projetado para capturar todas as entidades e relacionamentos do domínio de monitoria, seguindo princípios de normalização e integridade referencial. O esquema relacional adota a terceira forma normal (3NF), eliminando redundâncias e garantindo que cada atributo dependa exclusivamente da chave primária de sua tabela. Chaves estrangeiras com restrições de integridade referencial asseguram consistência entre entidades relacionadas, impedindo operações que resultariam em registros órfãos ou referências inválidas. O modelo contempla ainda índices compostos estratégicamente posicionados nas colunas mais frequentemente consultadas, otimizando operações de busca e agregação que sustentam os dashboards e relatórios do sistema.

A modelagem foi orientada pelos fluxos de negócios identificados durante o levantamento de requisitos, priorizando a rastreabilidade de operações e a auditabilidade de mudanças de estado. Cada entidade principal mantém campos de metadados temporais (`created_at`, `updated_at`) que permitem reconstruir o histórico de modificações. Relacionamentos muitos-para-muitos foram decompostos em tabelas associativas que capturam atributos específicos da associação (por exemplo, a relação entre projeto e disciplina inclui informações sobre o papel de cada disciplina no projeto). A Figura 2 apresenta as principais entidades e seus relacionamentos:

Modelo de dados - Sistema de Monitoria-IC
Diagrama ERD (PK/FK) do núcleo do domínio (alinhado ao schema atual).

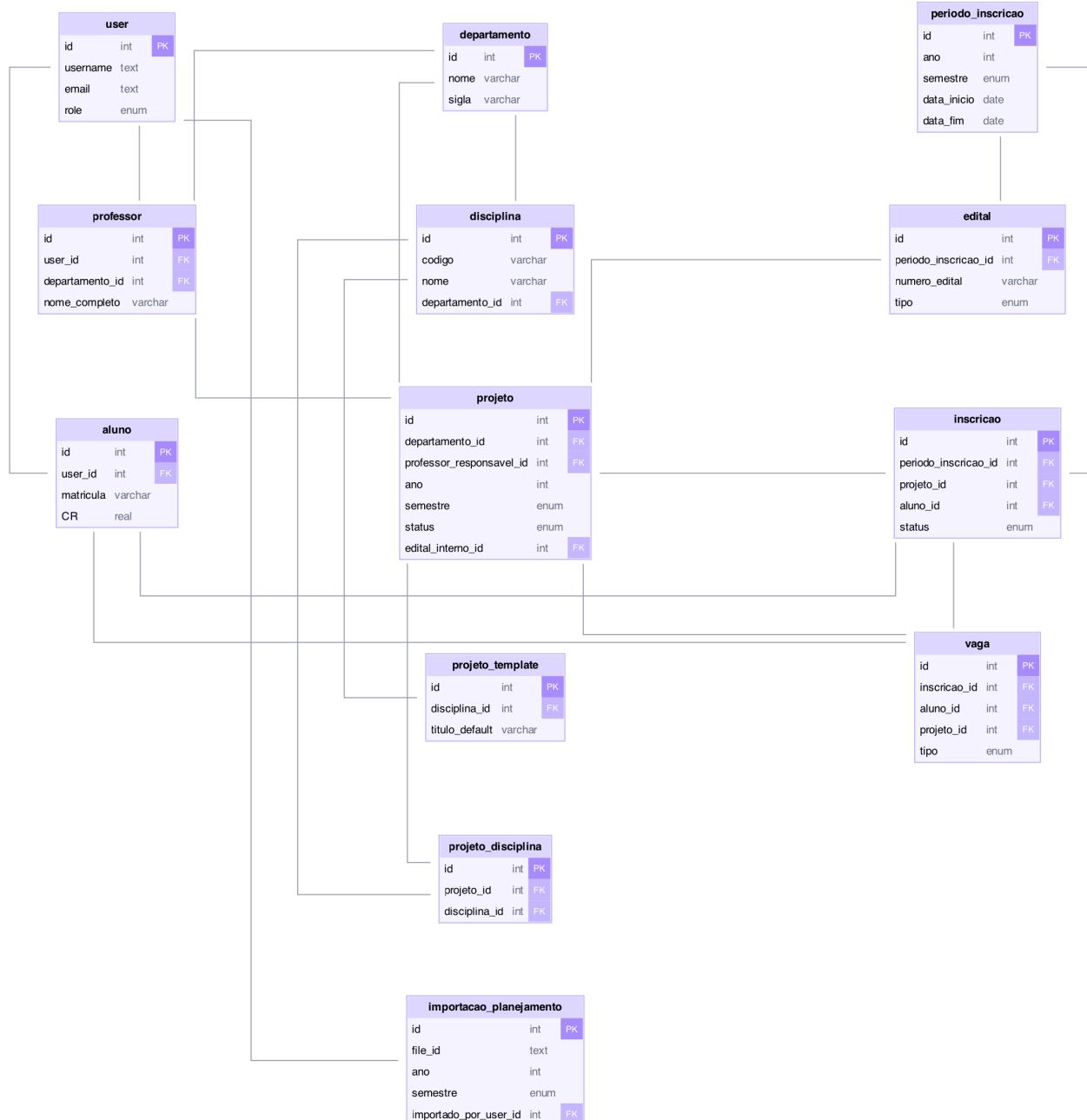


Figura 2. Modelo de dados resumido do domínio de monitoria.

O dashboard administrativo consolida indicadores operacionais (projetos por status, períodos de inscrição, volume de candidaturas) e oferece navegação rápida para tarefas críticas do papel *admin*, conforme a estrutura do menu lateral.

No núcleo de usuários, a entidade *user* concentra autenticação e perfil básico, estendida por perfis específicos de professor (dados acadêmicos) e aluno (matrícula, curso e CR), com papéis representados por *role* (*admin, professor, student*). No eixo acadêmico, departamento, curso e disciplina descrevem a hierarquia institucional, os cursos vinculados e as disciplinas (com códigos e equivalências). No domínio da monitoria, *periodo_inscricao* e *edital* organizam publicações por semestre; projeto evolui por estados bem definidos (DRAFT, SUBMITTED, APPROVED/REJECTED) e se relaciona a *projeto_template* para reuso; *inscricao* registra candidaturas e notas; vaga modela posições de monitor (BOLSISTA, VOLUNTARIO); e *importacao_planejamento* registra o histórico de importações semestrais (com arquivo, totalizações e usuário responsável).

4.5 Fluxo de Processos

O fluxo de monitoria é orquestrado em cinco etapas contínuas que estruturam o semestre, conforme ilustrado na Figura 3. Neste trabalho, a etapa enfatizada (e adotada como foco da validação do sistema) é o fluxo de *Planejamento & Criação* até *Aprovação & PROGRAD*, pois concentra o maior impacto na redução de retrabalho e na padronização dos documentos encaminhados ao Instituto. Na prática, esse fluxo se materializa em módulos específicos da aplicação: (i) a administração acessa *Importar Planejamento* para enviar a planilha semestral; o sistema valida o arquivo, cria projetos iniciais por disciplina e dispara notificações automáticas aos professores, sinalizando que há projetos pendentes de revisão e assinatura; (ii) ao acessar o dashboard, o professor visualiza os projetos em estado de pendência e, caso não exista template prévio, é conduzido a criar um template padrão da disciplina (título, descrição, carga horária e atividades), permitindo reuso nos próximos semestres; (iii) em seguida, o professor revisa o projeto do semestre, gera o PDF e realiza a assinatura digital no módulo *Assinatura de Documentos*; (iv) a submissão assinada torna o projeto elegível para revisão administrativa em *Gerenciar Projetos*, onde o administrador pode aprovar ou rejeitar com feedback; e (v) quando aprovado, o projeto passa a compor automaticamente a planilha consolidada em *Planilha para Instituto*, disponível para download e envio por e-mail, incluindo referências e anexos aos PDFs dos projetos aprovados, facilitando o trâmite com o Instituto/PROGRAD.

As demais etapas (*Alocação & Edital*, *Inscrições & Seleção* e *Aceite & Consolidação*) complementam o ciclo e já estão contempladas no sistema, permitindo a configuração de vagas, publicação de edital interno, inscrição de estudantes, seleção e consolidação final de dados bancários e planilhas para órgãos responsáveis.¹

Em *Gerenciar Projetos*, o administrador acompanha sub-

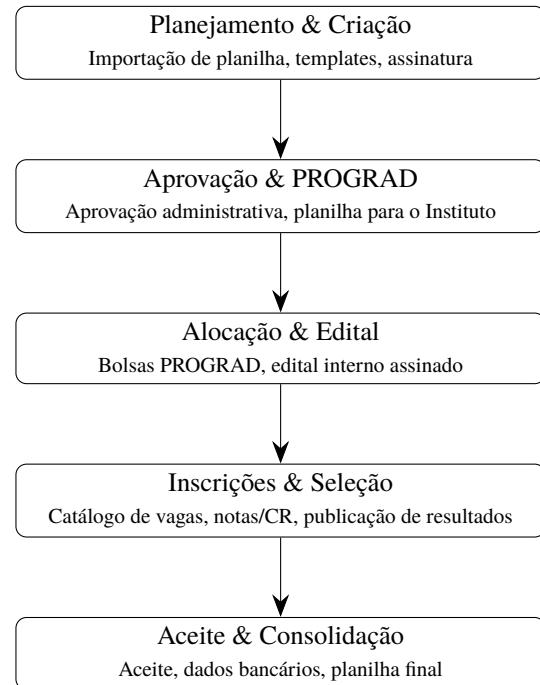


Figura 3. Fluxo de processos do semestre de monitoria.

missões, aprovações e gera a planilha consolidada para o Instituto/PROGRAD, garantindo padronização e trilhas de auditoria.

4.6 Implementação e Tecnologias

O sistema foi implementado utilizando um conjunto coeso de tecnologias modernas e consolidadas, selecionadas para maximizar produtividade de desenvolvimento, segurança e manutenibilidade. No *frontend*, adotou-se Next.js 15.1.4 com App Router, um framework React de produção que oferece renderização híbrida (SSR e CSR), otimizações automáticas de desempenho e roteamento baseado em sistema de arquivos. TypeScript 5.x foi utilizado em todo o código-fonte para garantir *type safety* e detectar erros em tempo de desenvolvimento. A interface visual foi construída com Tailwind CSS, um framework utilitário que proporciona consistência e rapidez na estilização, complementado por shaden/ui, uma coleção de componentes acessíveis e personalizáveis. Os formulários são implementados com React Hook Form, que minimiza re-renderizações desnecessárias e oferece validação eficiente, combinado com Zod para esquemas de validação compartilhados entre cliente e servidor. A sincronização de estado remoto e o gerenciamento de cache são tratados por React Query (TanStack Query), que automatiza requisições, invalidações e atualizações otimistas.

No *backend*, a API foi construída com tRPC v11, uma solução inovadora que elimina a necessidade de definições duplicadas de tipos entre cliente e servidor, garantindo que toda comunicação seja completamente tipada em tempo de compilação. A autenticação e gerenciamento de sessões são realizados por Lucia Auth, um framework moderno que oferece flexibilidade e segurança superiores a soluções legadas. A persistência de dados utiliza Drizzle ORM em conjunto com PostgreSQL, proporcionando consultas SQL tipadas e migrações versionadas de forma declarativa. Para armazenamento de arquivos binários (históricos acadêmicos, documentos comprobatórios, PDFs gerados), o sistema emprega

¹O recorte deste trabalho segue o planejamento visível na Figura 3, com foco nas fases implementadas até a geração da planilha consolidada para o Instituto/PROGRAD. Esse fluxo foi revisado passo a passo em conjunto com o Prof. Rubisley de Paula Lemes, assegurando aderência ao procedimento esperado no contexto do Instituto de Computação.

MinIO, uma solução compatível com o protocolo S3 da Amazon que facilita futuras migrações para serviços de nuvem. O envio automatizado de notificações por e-mail é realizado via Nodemailer, configurado para utilizar o servidor SMTP institucional da UFBA.

A infraestrutura de *DevOps* foi projetada para garantir qualidade e confiabilidade através de práticas modernas de integração e entrega contínuas. A solução completa é containerizada com Docker, incluindo o banco de dados PostgreSQL, o servidor MinIO e a aplicação Node.js, facilitando implantação consistente em diferentes ambientes (desenvolvimento, homologação e produção). Uma pipeline automatizada no GitHub Actions executa validações a cada commit: linting e formatação de código com Biome, verificação estática de tipos com TypeScript, execução de testes unitários com Vitest para validar a lógica de negócio, execução de testes end-to-end com Playwright para validar fluxos completos do usuário, e build de produção para detectar erros de compilação antecipadamente. Essa abordagem garante que apenas código validado e livre de regressões seja promovido para ambientes de produção.

A organização da lógica de negócio segue uma estrutura modular baseada em routers tRPC especializados por domínio. O router principal (`appRouter`) agrupa routers específicos para autenticação e usuários (`auth`, `me`, `user`, `apiKey`), entidades acadêmicas (`departamento`, `curso`, `discipline`), gestão de monitoria (`projeto`, `projetoTemplates`, `edital`, `inscricao`, `selecao`, `vagas`), funcionalidades administrativas (`importProjects`, `scholarshipAllocation`, `analytics`, `relatorios`) e infraestrutura transversal (`file`, `signature`, `notificacoes`). Essa separação facilita a compreensão do código, permite testes isolados de cada módulo e viabiliza o desenvolvimento paralelo por múltiplos desenvolvedores sem conflitos significativos.

4.7 Interfaces do Sistema

O sistema oferece interfaces especializadas para cada perfil de usuário, conforme exemplificado nas Figuras 4 a 15. A página inicial (Figura 4) apresenta a proposta do sistema, acesso ao login institucional e comunicação de períodos de inscrição quando ativos, funcionando como ponto de entrada para estudantes, professores e administradores. A tela de login (Figura 5) valida credenciais locais ou redireciona ao CAS quando habilitado, estabelecendo a sessão do usuário e direcionando ao dashboard apropriado.

O dashboard administrativo (Figura 6) consolida indicadores operacionais (projetos por status, períodos de inscrição, volume de candidaturas) e oferece navegação rápida para tarefas críticas do papel *admin*. A interface de assinatura de projetos (Figura 7) permite que o professor visualize o PDF gerado, use a assinatura padrão do perfil ou desenhe uma nova assinatura antes de submeter para análise administrativa.

Para estudantes, a tela de inscrição (Figura 8) oferece filtros por departamento e tipo de vaga, busca por título/professor e exibe avisos sobre o período ativo; quando o período está fechado, os componentes permanecem acessíveis para consulta e planejamento. Após criar um projeto, o professor pode editá-lo através da interface de edição (Figura 9), que permite ajustar informações do plano de monitoria, atividades

The screenshot shows the main landing page of the system. It features three main sections: 'Para Professores' (For Professors) with tasks like creating and managing projects, selecting candidates, defining volunteer positions, and digitizing documents; 'Para Alunos' (For Students) with tasks like visualizing available projects, registering in projects, monitoring candidate status, and accepting or rejecting offers; and 'Para Administradores' (For Administrators) with tasks like approving or rejecting projects, defining periods, defining volunteer positions, and managing users and reports. Below these is a 'Fluxo do Processo' (Process Flow) diagram with five numbered steps: 1. Criação e Submissão de Projetos (Professores criam propostas de monitoria definindo objetivos, disciplinas vinculadas e professores participantes. Após finalização, submetem para análise administrativa); 2. Análise Administrativa e Criação do Edital (Administradores analisam projetos submetidos, aprovam ou rejeitam propostas, definem quantidade de vagas com bolsa e criam edital oficial do período de inscrição); 3. Abertura de Inscrições para Alunos (Durante o período definido no edital, alunos visualizam projetos aprovados e se candidatam às vagas de monitoria (bolsistas ou voluntárias)); 4. Processo Seletivo e Definição de Vagas (Professores analisam candidaturas, realizam processo seletivo, definem vagas voluntárias adicionais e registram resultados através da ata de seleção); 5. Aceite e Início das Atividades (Alunos selecionados aceitam ou recusam ofertas (máximo 1 bolsa por semestre), assinam termos de compromisso e iniciam as atividades de monitoria).

Figura 4. Página inicial pública do sistema.

This is a screenshot of the local login page. It has fields for 'E-mail' (with placeholder 'seu_email@example.com') and 'Senha' (password field). Below these are buttons for 'Entrar' (Login) and 'Esqueci minha senha' (Forgot my password). At the bottom, there are links for 'Não possui conta?' (Does not have an account?) with 'Cadastrar-se' (Register) and '← Voltar ao inicio' (← Back to start).

Figura 5. Tela de login local do sistema.

This screenshot shows the administrative dashboard ('Dashboard'). On the left is a sidebar with navigation links for 'Dashboard', 'Projetos' (with sub-links for Importar Planilhas, Gerenciar Projetos, Templates de Projeto, Edital, Inscrições, Seleção, Relatórios, Usuários, Configurações, and Sessões), 'Notificações', and 'Perfil'. The main area has tabs for 'Projetos', 'Professores', and 'Alunos'. It displays metrics: 'Ressarcidos' (0 projects in progress), 'Em Análise' (1 project pending admin approval), 'Aprovados' (1 project approved for submission), and 'Rejeitados' (0 rejected). Below this is a table for 'Componente curricular' (MATA37) with columns for Status (Approved), Ano (2025), Semestre (2º), Voluntários (2), and Inscrições (1). Navigation buttons 'Anterior', '1 de 1', and 'Próximo' are at the bottom right.

Figura 6. Dashboard administrativo com métricas.

previstas e configurações de vagas antes da submissão.

As funcionalidades administrativas incluem o gerenciamento de projetos (Figura 10), onde o administrador acompanha submissões e aprovações e gera a planilha consolidada para o Instituto/PROGRAD; a gestão de editais (Figura 11), que permite configurar datas, pontos de prova e bibliografia, solicitar assinatura digital do chefe do departamento e publicar com notificações automáticas; e a alocação de bolsas

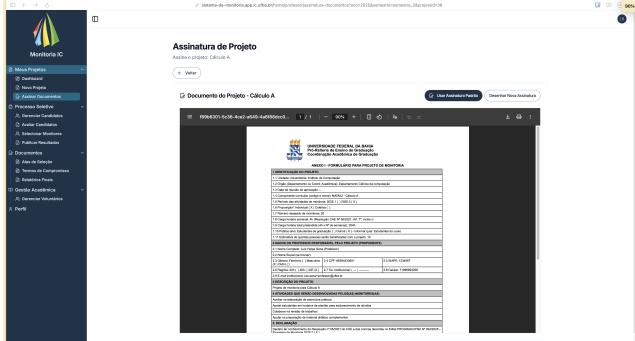


Figura 7. Assinatura de projeto pelo professor com prévia do PDF.



Figura 11. Gestão de editais com status e ações.

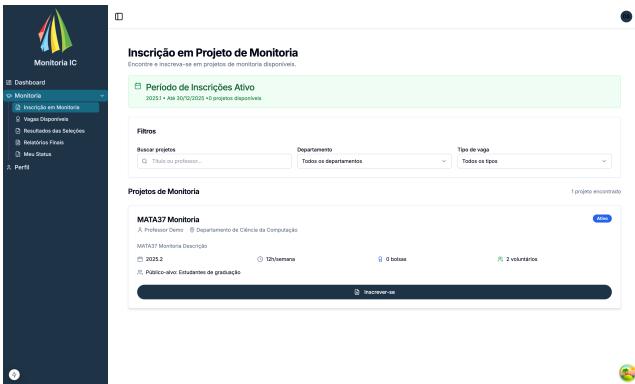


Figura 8. Inscrição do aluno em projetos de monitoria.

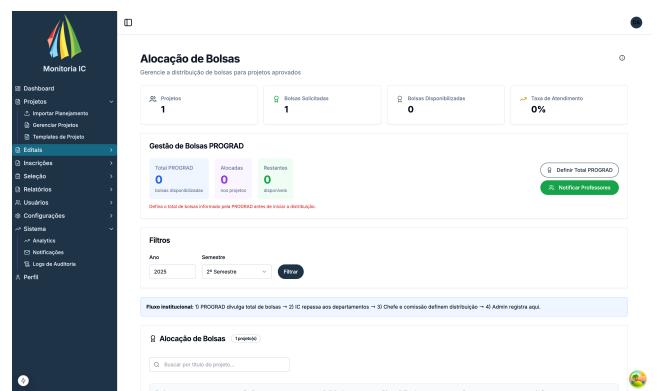


Figura 12. Interface de alocação de bolsas.

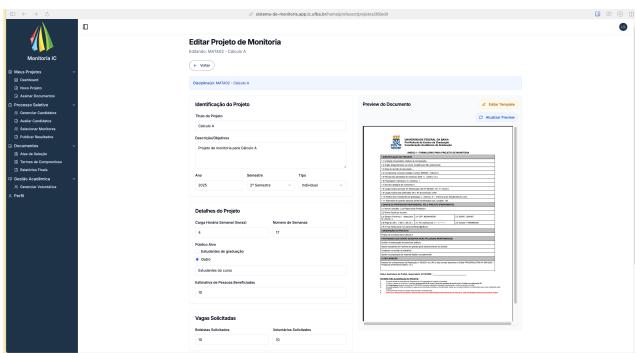


Figura 9. Interface de edição de projeto de monitoria pelo professor.

(Figura 12), que impede excedentes em relação ao total de bolsas concedidas pela PROGRAD.

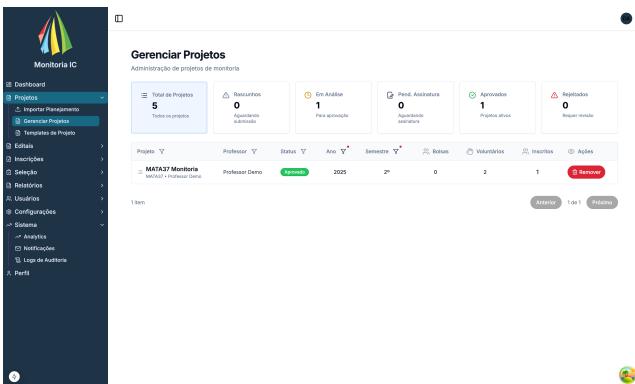


Figura 10. Gerenciamento de projetos por semestre.

Completam as interfaces administrativas a gestão de usuários (Figura 13), onde a administração pesquisa, filtra e gerencia contas mantendo coerência com regras institucionais; a importação de planejamento (Figura 14), que processa pla-

nilhas do planejamento e emite relatório de inconsistências; e a tela de departamentos (Figura 15), que permite manter a estrutura acadêmica essencial para vínculos de disciplinas e projetos. A interface de análise de projetos (Figura 16) permite ao administrador revisar projetos submetidos pelos professores, visualizando detalhes do plano de monitoria, informações da disciplina e dados do professor responsável antes de aprovar ou solicitar revisões. Por fim, a geração da planilha para a PROGRAD (Figura 17) consolidada todos os projetos aprovados em formato exportável para envio ao Instituto.

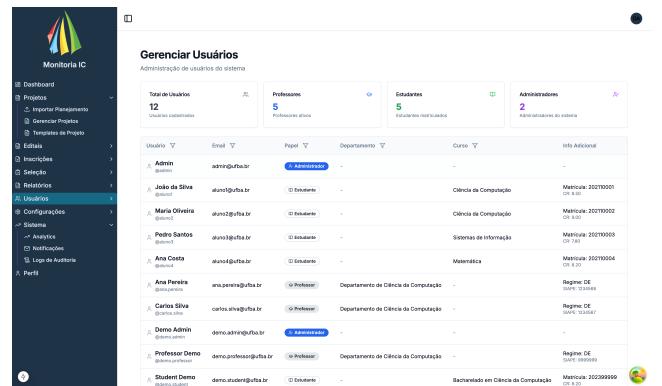


Figura 13. Administração de usuários.

5 Avaliação Experimental

O objetivo desta avaliação experimental é validar a efetividade do Sistema de Monitoria-IC em dois eixos complementares: (i) eficiência operacional, mensurando ganhos potenciais em tempo, redução de erros e automação de tarefas em relação ao processo manual anterior; e (ii) qualidade técnica da imple-

Figura 14. Importação de planejamento.

Figura 15. Gestão de departamentos.

Figura 16. Interface de análise de projetos submetidos pelo administrador.

mentação, analisando robustez, cobertura de testes e aderência da arquitetura às boas práticas de sistemas de informação. Nesta seção, descrevem-se a metodologia adotada, os resultados técnicos obtidos e as principais limitações e ameaças à validade.

5.1 Metodologia

A avaliação foi conduzida em ambiente de desenvolvimento e teste, utilizando dados de exemplo que reproduzem cenários

Figura 17. Geração de planilha PROGRAD com projetos aprovados.

reais de monitoria do Instituto de Computação. A metodologia adotada combinou três frentes principais:

1. Testes end-to-end (E2E): foram especificados cenários automatizados que percorrem o fluxo completo de monitoria, desde a criação e submissão de projetos pelos docentes até a seleção de monitores, aceite de vagas e consolidação final para a PROGRAD. Esses cenários incluem variações de papéis (administração, professores, estudantes), estados de projetos (rascunho, submetido, aprovado) e tipos de vaga (bolsista, voluntário).
2. Testes unitários e de integração: serviços centrais da camada de negócio (gestão de projetos, inscrições, seleção, relatórios) foram exercitados por meio de testes unitários e de integração, verificando regras de negócios críticas, validações de dados e interações com a camada de dados. A utilização de um ORM tipado (Drizzle) facilitou a detecção de inconsistências entre modelo de dados e código.
3. Métricas de execução e logs de aplicação: durante a execução dos testes automatizados, foram coletadas métricas de tempo de resposta das principais rotas e registros de log para monitorar possíveis falhas, exceções não tratadas ou degradação de desempenho. Essas informações subsidiaram a análise de robustez e escalabilidade da solução em cenários típicos de uso.

Não foi realizado, até o momento, um piloto formal em produção com usuários finais (docentes, estudantes e administradores). A avaliação com pessoas usuárias é, portanto, considerada trabalho futuro, a ser conduzida em ambiente real de monitoria, conforme discutido na Seção 6.

5.2 Resultados Técnicos

Os resultados obtidos até esta etapa de avaliação concentram-se na dimensão técnica do sistema. A Tabela 4 sintetiza alguns indicadores relevantes dos testes automatizados executados.

A execução dos testes unitários e de integração demonstrou robustez na implementação das regras de negócio e validações de autorização. Os 55 testes automatizados, distribuídos em 15 arquivos de teste organizados por domínio (departamento, disciplina, projeto, inscrição, edital, seleção, usuário, arquivo, termos, analytics, PDF, templates, onboarding, API keys), cobrem fluxos críticos do sistema: controle de acesso baseado em papéis (verificando que administradores, professores e estudantes acessam apenas funcionalidades pertinentes), workflow de aprovação de projetos (transições de estado DRAFT → SUBMITTED → APPROVED/REJECTED), pro-

Tabela 4. Síntese de resultados técnicos da avaliação preliminar.

Indicador	Valor Medido
Testes unitários	55 testes em 15 arquivos, cobrindo routers tRPC (autenticação, projetos, inscrições, seleção, editais, arquivos, analytics, relatórios)
Taxa de sucesso	100% (55/55 testes aprovados sem falhas ou regressões)
Tempo execução	1.07s para a suíte completa (73ms em testes, 173ms em setup, 992ms em transformação TypeScript)
Inicialização	Servidor Next.js iniciando em 1.34s em ambiente de desenvolvimento
Cobertura funcional	Validação de regras críticas: controle de acesso RBAC, workflow de projetos, processo seletivo, importação de planejamento, publicação de editais
Estabilidade	Zero erros críticos não tratados; todas as exceções de negócio corretamente transformadas em TRPCError

cesso seletivo com cálculo automático de notas finais ponderadas, importação de planejamento com validação de consistência de dados SIAPE, publicação de editais com validação de assinatura digital obrigatória, e gerenciamento de inscrições considerando equivalências curriculares configuradas. A taxa de sucesso de 100% e o tempo de execução reduzido (1.07s para toda a suíte) evidenciam a estabilidade da solução e a eficácia das práticas de desenvolvimento adotadas, incluindo validação estática de tipos via TypeScript, uso de ORM tipado (Drizzle) para prevenir inconsistências entre modelo de dados e código, e pipeline de integração contínua executando testes automaticamente a cada commit.

As métricas de desempenho coletadas em ambiente de desenvolvimento local indicam tempo de inicialização do servidor Next.js de 1.34 segundos, demonstrando otimização adequada do processo de build e carregamento de dependências. A arquitetura adotada, com separação clara entre camadas de apresentação, aplicação, negócio e dados, contribui para a manutenibilidade e testabilidade do sistema, permitindo validação isolada de cada componente e facilitando a identificação de regressões. A utilização de tRPC v11 para comunicação type-safe entre cliente e servidor elimina categorias inteiras de erros de integração, conforme evidenciado pela ausência de falhas relacionadas a incompatibilidades de tipos ou contratos de API nos testes executados.

Cabe ressaltar que esta avaliação concentra-se em aspectos técnicos e funcionais do sistema, validando a corretude da implementação em relação aos requisitos especificados na Tabela 3. Uma avaliação complementar com usuários finais (professores, estudantes e administradores) em ambiente real de monitoria, incluindo métricas de usabilidade percebida, satisfação, curva de aprendizado e impacto efetivo na redução de tempo gasto em tarefas administrativas, permanece como trabalho futuro prioritário, conforme discutido na Seção 6.

5.3 Avaliação com Usuários (Estudo Qualitativo)

Além da validação técnica descrita acima, foi conduzida uma avaliação qualitativa exploratória com dois participantes representando os perfis principais do sistema: um administrador (coordenador de monitoria) e um professor. A escolha por abordagem qualitativa, em detrimento de questionários quantitativos com escala Likert, justifica-se pelo número reduzido de participantes disponíveis ($N=2$), o que inviabiliza análises estatísticas significativas. Estudos qualitativos com amostras pequenas são apropriados para identificar problemas de usabilidade e obter insights profundos sobre a experiência do usuário Nielsen [1994], Lazar *et al.* [2017].

O foco da avaliação compreendeu o fluxo enfatizado neste trabalho: importação de planejamento → criação/reuso de templates → assinatura de projetos → aprovação administrativa → geração de planilha para o Instituto/PROGRAD.

5.3.1 Metodologia

A metodologia adotada combinou três técnicas complementares de coleta de dados:

1. **Walkthrough guiado:** cada participante realizou tarefas predefinidas correspondentes ao seu perfil, seguindo um roteiro estruturado que cobria as funcionalidades principais do sistema. Durante a execução, foram registrados checkpoints de sucesso/falha, tempo gasto e dificuldades observadas.
2. **Protocolo Think-Aloud:** os participantes foram instruídos a verbalizar seus pensamentos durante a interação com o sistema, expressando expectativas, dúvidas, surpresas e frustrações. Essa técnica permite capturar a experiência do usuário em tempo real, identificando problemas de usabilidade que não seriam evidentes apenas pela observação.
3. **Entrevista semiestruturada:** após o walkthrough, foi conduzida uma entrevista de aproximadamente 15 minutos com cada participante, explorando percepções gerais, pontos positivos e negativos, comparação com o processo manual atual e sugestões de melhoria.

As sessões foram gravadas (tela e áudio) mediante consentimento dos participantes, totalizando aproximadamente 1 hora por participante. A análise dos dados seguiu a técnica de análise temática Braun and Clarke [2006], identificando padrões recorrentes nas verbalizações e respostas.

A avaliação contou com dois participantes representando os perfis principais do sistema: o Prof. Rubisley (coordenador de monitoria, perfil administrador) e o Prof. Frederico (perfil professor). A Tabela 5 apresenta os participantes e suas atribuições no estudo. O administrador realizou tarefas de importação de planejamento, verificação de notificações automáticas, aprovação de projetos e geração de planilha PROGRAD. O professor executou o fluxo completo de primeiro acesso (recuperação de senha e onboarding com SIAPE, CPF e assinatura digital), visualização do projeto criado pela importação, criação de template reutilizável, edição de projeto do semestre (bolsas, voluntários, atividades), pré-visualização do PDF e assinatura digital com submissão.

O walkthrough foi estruturado em três sessões sequenciais simulando o fluxo real de uso: na primeira sessão, o

Tabela 5. Participantes da avaliação qualitativa.

Participante	Perfil	Tarefas Realizadas
Prof. Rubisley	Admin	Importar planejamento, verificar notificações, aprovar projeto, gerar planilha PROGRAD
Prof. Frederico	Professor	Primeiro acesso (senha), onboarding, criar template, editar projeto, assinar e submeter

administrador realizou login, importou a planilha de planejamento do DCC, verificou os professores notificados automaticamente e visualizou os projetos criados com status pendente de assinatura. Na segunda sessão, o professor executou o primeiro acesso via “Esqueci minha senha”, completou o onboarding, visualizou o projeto criado pela importação, criou/editou template padrão para a disciplina, editou o projeto do semestre e finalizou com pré-visualização do PDF, assinatura digital e submissão. Na terceira sessão, o administrador retornou para visualizar o projeto submetido, analisar os detalhes, aprovar o projeto e gerar a planilha consolidada para o Instituto/PROGRAD.

Após cada walkthrough, foi conduzida uma entrevista semiestruturada de aproximadamente 15 minutos, explorando a experiência geral do participante, comparação com o processo manual atual, estimativa de economia de tempo, barreiras percebidas para adoção em semestre real e sugestões de melhoria. As perguntas foram organizadas por categoria de análise (experiência geral, funcionalidades específicas por perfil, comparação, economia de tempo, adoção e sugestões), conforme apresentado na Tabela 6. Esta abordagem seguiu as recomendações de Lazar et al. Lazar et al. [2017] para condução de estudos qualitativos em IHC, permitindo explorar em profundidade as percepções e necessidades dos usuários enquanto mantém foco nas dimensões relevantes para validação do sistema. A estrutura do roteiro foi desenhada para capturar tanto aspectos funcionais quanto percepções subjetivas, possibilitando uma avaliação holística da experiência de uso do sistema por diferentes perfis de usuários acadêmicos.

5.3.2 Resultados da Avaliação Qualitativa

A análise temática das verbalizações e entrevistas, seguindo o método de Braun e Clarke Braun and Clarke [2006], identificou três categorias principais: usabilidade, utilidade e adoção. A Figura 18 apresenta o mapa temático resultante da análise, sintetizando os principais achados por categoria.

Perfil Administrador:

- Pontos positivos identificados: O participante destacou a organização lógica do menu lateral, que segue a ordem natural do fluxo de trabalho (projetos, editais, inscrições, seleção, relatórios). A importação de planejamento foi considerada eficiente, com feedback visual claro sobre o número de projetos criados e professores notificados. A planilha PROGRAD gerada foi avaliada como adequada ao formato institucional esperado.
- Dificuldades encontradas: Não foram identificadas dificuldades críticas. O participante executou todas as tarefas do roteiro sem interrupções ou erros significativos.

- Sugestões de melhoria: Foi sugerida a inclusão de um botão para notificar o professor quando o projeto necessita de ajustes, permitindo comunicação direta via e-mail com campo para mensagem personalizada. Também foi proposta a relocação da funcionalidade de geração de planilha PROGRAD para dentro da seção de Projetos, mantendo cada planilha na etapa correspondente do fluxo.

Perfil Professor:

- Pontos positivos identificados: A funcionalidade de templates reutilizáveis foi avaliada de forma muito positiva, sendo descrita pelo participante como excelente para economia de tempo em semestres futuros. O processo de onboarding (primeiro acesso) foi considerado claro e o fluxo de assinatura digital intuitivo, com a pré-visualização do PDF antes da assinatura sendo destacada como útil.
- Dificuldades encontradas: Houve uma confusão inicial no processo de redefinição de senha, que foi resolvida durante a própria sessão. O participante preencheu incorretamente um campo do formulário (público-alvo), demonstrando um cenário realista de uso que justifica a necessidade da funcionalidade de notificação sugerida pelo administrador.
- Sugestões de melhoria: Foi sugerido que o projeto permaneça editável após o envio ao administrador, sendo bloqueado apenas após a aprovação final, facilitando correções quando solicitadas.

Análise Cruzada:

- Padrões comuns entre os perfis: Ambos os participantes validaram o fluxo principal do sistema, desde a importação até a geração da planilha PROGRAD. A avaliação geral foi positiva, com reconhecimento explícito da contribuição do sistema para a modernização do processo de monitoria. Os participantes concordaram que o sistema está pronto para uso em um semestre real e que continuará evoluindo com contribuições de outros desenvolvedores.
- Divergências de percepção: A principal divergência refere-se à localização da funcionalidade de planilha PROGRAD: enquanto o sistema atual a posiciona na seção de Relatórios, o administrador manifestou preferência por mantê-la dentro da seção de Projetos, argumentando que cada etapa deve conter suas próprias planilhas de consolidação.

Validação dos Objetivos Específicos:

A Tabela 7 apresenta a validação dos objetivos específicos com base nas evidências coletadas durante as sessões de avaliação.

Intenção de Adoção:

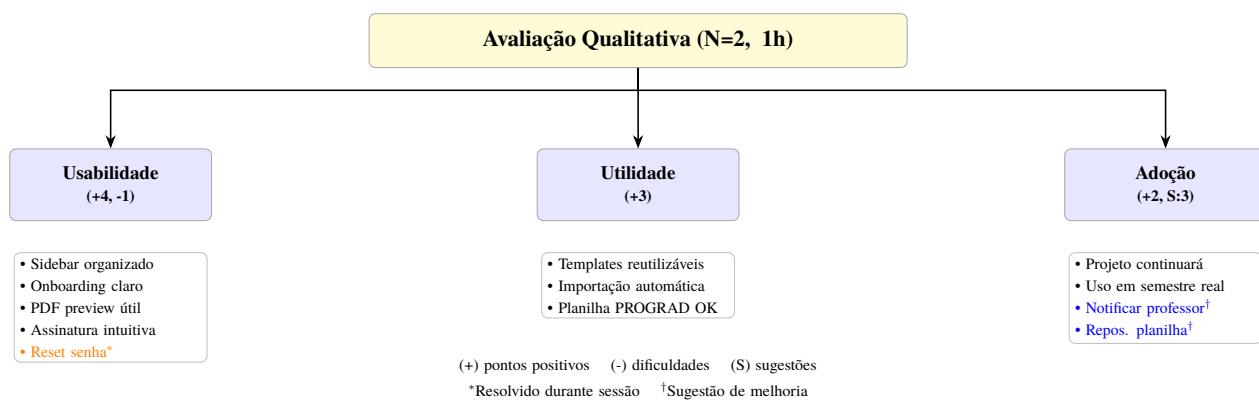
Ambos os participantes confirmaram a viabilidade do sistema para uso em um semestre real. O administrador destacou que o projeto é contínuo e evoluirá com contribuições futuras de outros desenvolvedores. As barreiras identificadas para adoção plena incluem a necessidade de implementar a funcionalidade de notificação ao professor para solicitação de ajustes e pequenos ajustes de nomenclatura e posicionamento de funcionalidades na interface.

Tabela 6. Roteiro de entrevista semiestruturada por perfil de usuário.

Categoría	Perfil	Pergunta
Experiência geral	Ambos	Descreva sua experiência geral com o sistema. O que mais chamou sua atenção (positivo ou negativo)?
Importação	Admin	A importação da planilha funcionou como esperava? Os erros (se houve) foram bem explicados?
Notificações	Admin	Conseguiu verificar quais professores foram notificados? O feedback do sistema foi suficiente?
Primeiro acesso	Professor	O processo de primeiro acesso (recuperação de senha, onboarding) foi tranquilo? O que poderia melhorar?
Template	Professor	A criação de template foi útil? Usaria em semestres futuros para economizar tempo?
Assinatura digital	Professor	O processo de assinatura digital foi intuitivo? Confia que a assinatura foi registrada corretamente?
PDF gerado	Professor	O PDF gerado atendeu suas expectativas? A pré-visualização foi útil antes de assinar?
Aprovação	Admin	O fluxo de aprovação/rejeição está intuitivo? As informações são suficientes para tomar a decisão?
Planilha PROGRAD	Admin	A planilha gerada atende às necessidades reais? Precisaria de ajustes manuais antes de enviar?
Comparação	Ambos	Comparando com o processo manual atual (planilhas, emails): o que ficou mais fácil? O que ainda parece complicado?
Economia de tempo	Ambos	Quanto tempo você estima que economizaria por semestre usando o sistema?
Adoção	Ambos	Conseguiria usar esse sistema num semestre real? Quais barreiras precisariam ser resolvidas?
Sugestões	Ambos	O que mudaria ou adicionaria ao sistema? Alguma funcionalidade crítica está faltando?

Tabela 7. Validação dos objetivos específicos com base na avaliação qualitativa.

Objetivo Específico	Status	Evidência
Digitalizar ciclo completo de projetos	✓ Validado	Fluxo completo executado com sucesso: importação de planejamento, criação de template, assinatura digital, aprovação administrativa e geração de planilha PROGRAD
Automatizar processo seletivo	✓ Parcial	Automação de notificações por e-mail e geração de PDFs confirmada; sugerida melhoria para notificação de ajustes em projetos
Sistematizar alocação de bolsas	✓ Validado	Validação de limites funcionando corretamente; planilha PROGRAD gerada no formato institucional esperado
Eliminar trabalhos manuais repetitivos	✓ Validado	Templates reutilizáveis avaliados como excelentes para economia de tempo em semestres futuros; alto impacto na redução de tarefas repetitivas
Fornecer base analítica para decisões	✓ Parcial	Dashboard funcional com métricas operacionais; analytics avançado identificado como trabalho futuro

**Figura 18.** Mapa temático da avaliação qualitativa com usuários.

5.4 Limitações e Ameaças à Validade

Esta avaliação apresenta limitações importantes que devem ser consideradas na interpretação dos resultados:

- **Amostra reduzida:** a avaliação qualitativa foi conduzida com apenas dois participantes ($N=2$), o que limita a generalização dos resultados. Embora estudos qualitativos com amostras pequenas sejam válidos para identificar problemas de usabilidade Nielsen [1994], uma avaliação mais abrangente com maior número de usuários seria desejável para confirmar os padrões identificados.
- **Ausência do perfil estudante:** a avaliação focou nos perfis de administrador e professor, não incluindo estudantes. O fluxo de inscrição, seleção e aceite de vagas pelos alunos não foi avaliado empiricamente com usuários reais.
- **Ambiente controlado:** os experimentos foram realizados em ambiente de desenvolvimento/staging, com dados de exemplo. A extração para contextos de maior escala (por exemplo, uso institucional ampliado para toda a UFBA) deve ser feita com cautela.
- **Ausência de comparação quantitativa:** não foram coletadas métricas comparativas diretas (tempo gasto, número de erros) entre o processo manual e o processo automatizado. As estimativas de economia de tempo baseiam-se nas percepções dos participantes.
- **Viés de seleção:** os participantes são professores do Instituto de Computação, com familiaridade com sistemas digitais acima da média. Usuários com menor experiência tecnológica podem ter experiências diferentes.
- **Evolução contínua do sistema:** o Sistema de Monitoria-IC foi concebido como software em evolução. Alterações futuras podem alterar o comportamento observado, exigindo reavaliação periódica.

Em síntese, a avaliação experimental combinou validação técnica (55 testes automatizados, 100% aprovação) com avaliação qualitativa exploratória, fornecendo evidências preliminares de usabilidade e utilidade. A validação em ambiente real de produção, com maior número de usuários e métricas quantitativas de impacto, permanece como trabalho futuro prioritário.

6 Conclusão e Trabalhos Futuros

Este trabalho apresentou o desenvolvimento do Sistema de Monitoria-IC como solução específica para o *workflow* de monitoria. As contribuições centrais abrangem a automação das etapas principais do processo (da importação de planejamento à aprovação administrativa e geração de planilhas para envio ao Instituto/PROGRAD), a adoção de uma arquitetura moderna e escalável que separa SPT e funcionalidades gerenciais e a disponibilização de uma base estruturada que viabiliza futuras pesquisas e implantações em ambiente institucional.

A adoção plena do sistema tem o potencial de produzir efeitos mensuráveis no cotidiano administrativo, reduzindo o tempo gasto em tarefas repetitivas, eliminando retrabalho e padronizando procedimentos entre departamentos. Na dimensão estratégica, a consolidação de dados permite análises históricas e de tendências, qualificando a tomada de decisão e abrindo espaço para políticas institucionais orientadas por evi-

dências. Do ponto de vista pedagógico, a maior facilidade de participação tende a ampliar o alcance do programa, a transparência reforça a confiança de todos os atores e a economia de tempo contribui para liberar docentes para atividades-fim.

Do ponto de vista técnico, o requisito de maior complexidade foi a implementação da assinatura digital com sobreposição no PDF e armazenamento no MinIO, envolvendo manipulação de coordenadas precisas no documento e integração assíncrona com o serviço de arquivos. No âmbito organizacional, enfrentaram-se desafios relacionados à infraestrutura da UFBA (disponibilidade de servidores, configuração de rede, permissões de acesso) e ao alinhamento contínuo de requisitos com os stakeholders, especialmente na definição de formatos de planilhas e fluxos de aprovação compatíveis com procedimentos institucionais existentes.

Apesar dos resultados positivos, permanecem desafios: a integração com o sistema acadêmico ainda é parcial para captação automática de CR e histórico; o escopo atual está restrito ao Instituto de Computação, exigindo adaptações para expansão; parte do fluxo depende de setores externos (PROGRAD, NUMOP), o que mantém etapas manuais; e a ausência de um aplicativo móvel nativo pode limitar o acesso em determinados contextos.

O plano de evolução contempla três horizontes temporais. Num primeiro momento, previsto para os próximos seis meses, propõe-se a realização de um piloto institucional no próprio Instituto de Computação, utilizando o Sistema de Monitoria-IC em paralelo ao processo manual em ao menos um semestre letivo completo. Nessa etapa, serão coletadas métricas quantitativas (tempo de execução de tarefas, número de erros administrativos, volume de retrabalho) e dados qualitativos de satisfação por meio de questionários e entrevistas com professores, estudantes e gestores, a fim de complementar a avaliação técnica aqui apresentada com evidências empíricas de usabilidade e impacto organizacional. O código base de todas as funcionalidades está completo, incluindo o módulo de certificados digitais. Além do piloto, prioriza-se a conclusão da integração com o sistema acadêmico (SIAC) e o desenvolvimento do aplicativo móvel (React Native).

Num segundo momento, em torno de um ano, pretende-se consolidar a implantação no Instituto e avaliar a viabilidade de expansão para outros departamentos da UFBA, com adaptações de regras de negócio quando necessário. Essa fase inclui o aprofundamento de integrações com sistemas acadêmicos institucionais (como SIAC ou eventuais migrações para o SIGAA), a evolução do módulo de relatórios e analytics (incluindo sistema de recomendação para correspondência entre aluno e projeto, e analytics avançado com técnicas de aprendizado de máquina), integrações com plataformas de ensino (como Moodle) e a documentação de procedimentos operacionais para suporte e manutenção do sistema, definindo claramente responsabilidades de desenvolvimento, operação e atendimento a usuários.

No horizonte de longo prazo, busca-se generalizar o uso do Sistema de Monitoria-IC para outras unidades acadêmicas e, potencialmente, para outras universidades públicas brasileiras, com a disponibilização do código como software livre sob licença adequada. Pretende-se também criar um repositório compartilhado de templates entre instituições, permitindo que universidades compartilhem boas práticas e modelos de proje-

tos de monitoria. Em paralelo, estudos longitudinais poderão investigar o impacto da automação da monitoria sobre indicadores acadêmicos (participação de estudantes, desempenho em disciplinas, distribuição de bolsas), bem como a relação entre transparência processual e percepção de equidade por parte da comunidade. A consolidação dessas evidências pode orientar políticas institucionais de modernização administrativa baseadas em dados.

O Sistema de Monitoria-IC representa um avanço significativo na modernização da gestão acadêmica universitária. Ao digitalizar e automatizar processos tradicionalmente manuais, o sistema não apenas aumenta a eficiência operacional, mas também estabelece uma base sólida para a transformação digital contínua das universidades brasileiras. A experiência de desenvolvimento e implantação deste sistema demonstra que é possível criar soluções tecnológicas específicas para problemas acadêmicos complexos, utilizando tecnologias modernas e práticas de engenharia de software consolidadas. O sucesso inicial na UFBA sugere alto potencial de replicação em outras instituições que enfrentam desafios similares. Espera-se que este trabalho inspire outras iniciativas de modernização administrativa no ambiente universitário e contribua para o avanço da discussão sobre transformação digital na educação superior brasileira. O código fonte e documentação estarão disponíveis publicamente após a conclusão das funcionalidades principais, permitindo que outras instituições adaptem e evoluam a solução conforme suas necessidades específicas.

Agradecimentos

Agradeço ao Instituto de Computação da UFBA pelo apoio institucional, aos professores e alunos que participaram dos testes piloto, e à equipe de TI que viabilizou a infraestrutura necessária.

Referências

- Brasil (1996). Lei nº 9.394, de 20 de dezembro de 1996. estabelece as diretrizes e bases da educação nacional.
- Braun, V. and Clarke, V. (2006). Using thematic analysis in psychology. *Qualitative Research in Psychology*, 3(2):77–101. DOI: 10.1191/147808706qp063oa.
- Dantas, O. M. (2014). Monitoria: fonte de saberes à docência superior. *Revista Brasileira de Estudos Pedagógicos*, 95(241):567–589.
- Davenport, T. H. (1993). *Process Innovation: Reengineering Work through Information Technology*. Harvard Business School Press.
- de S.Paulo, F. (2024). Ranking universitário folha (ruf) 2024.
- Frison, L. M. B. (2016). Monitoria: uma modalidade de ensino que potencializa a aprendizagem colaborativa e autorregulada. *Pro-Posições*, 27(1):133–153.
- Hammer, M. and Champy, J. (1993). *Reengineering the Corporation: A Manifesto for Business Revolution*. Harper Business.
- Laudon, K. C. and Laudon, J. P. (2011). *Management Information Systems: Managing the Digital Firm*. Pearson, 12th edition.
- Lazar, J., Feng, J. H., and Hochheiser, H. (2017). *Research Methods in Human-Computer Interaction*. Morgan Kaufmann, Cambridge, MA, 2nd edition.
- Machado, M. B. (2023). Ic-requests: um sistema sensível ao contexto para suporte a solicitações acadêmicas. Trabalho de conclusão de curso (graduação em ciência da computação), Universidade Federal da Bahia, Salvador, BA.
- Natário, E. G. and Santos, A. A. A. d. (2010). Programa de monitores para o ensino superior. *Estudos de Psicologia*, 27(3):355–364.
- Nielsen, J. (1994). *Usability Engineering*. Morgan Kaufmann, San Francisco, CA.
- Santos, A. I. C. (2023). Alertufba: um sistema web de orientações sobre segurança e emergência no contexto universitário. Trabalho de conclusão de curso (graduação em sistemas de informação), Universidade Federal da Bahia, Salvador, BA.
- UNESCO (2023). Global Education Monitoring Report 2023: Technology in Education – A Tool on Whose Terms? Technical report, UNESCO, Paris. Relatório global sobre tecnologia na educação e transformação digital.
- Universidade de Brasília (2024). Sigaa – sistema integrado de gestão de atividades acadêmicas.
- Universidade de São Paulo (2024). Júpiterweb – sistema académico de graduação.
- Universidade Federal de São Paulo (2024). Sei – sistema eletrônico de informações (unifesp).
- Universidade Federal de Santa Catarina (2024). Cagr – controle acadêmico de graduação.
- Universidade Federal do Rio de Janeiro (2024). Siga – sistema integrado de gestão académica.
- World Bank (2022). GovTech Maturity Index, 2022 Update: Trends in Public Sector Digital Transformation. Technical report, World Bank, Washington, DC.