

ARTIGO DE PESQUISA/RESEARCH PAPER

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]

Antonieli Magalhães de Souza [Universidade Federal da Bahia | antonieli@ufba.br]

João Victor Leahy de Melo [Universidade Federal da Bahia | joao.leahy@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, mas sua gestão ainda enfrenta burocracia, falta de transparência e processos manuais ineficientes. Este trabalho apresenta o Sistema de Monitoria-IC, uma plataforma Web que automatiza o ciclo de vida dos projetos de monitoria na UFBA, da criação e assinatura digital pelos docentes até a aprovação administrativa e a geração de planilhas consolidadas para o Instituto e a PROGRAD. O sistema foi desenvolvido com Next.js 15, TypeScript, tRPC, PostgreSQL e MinIO, seguindo princípios de engenharia de software que favorecem escalabilidade e manutenibilidade. A arquitetura separa um Sistema de Processamento de Transações (SPT), para operações cotidianas, das funcionalidades gerenciais de análise e controle. A validação técnica por testes *end-to-end* demonstra a robustez da solução. Os resultados indicam eliminação de retrabalho administrativo, maior transparência por meio de histórico auditável e uma base sólida para a modernização da gestão acadêmica na universidade.

Abstract. Academic monitoring is a fundamental process in Brazilian universities, yet its management still suffers from bureaucracy, lack of transparency, and inefficient manual procedures. This work presents Sistema de Monitoria-IC, a web platform that automates the lifecycle of monitoring projects at UFBA, from project creation and digital signing by professors to administrative approval and the generation of consolidated spreadsheets for the Institute and PROGRAD. The system was built with Next.js 15, TypeScript, tRPC, PostgreSQL, and MinIO, following software engineering principles that favor scalability and maintainability. The architecture separates a Transaction Processing System (TPS), for daily operations, from managerial functionality for analysis and control. Technical validation through *end-to-end* tests demonstrates the robustness of the solution. Results indicate the elimination of administrative rework, greater transparency through auditable history, and 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.

Recebido/Received: 20 January 2026 • Aceito/Accepted: 17 July 2026 • Publicado/Published: 11 September 2026

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 importa 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 candi-

datos, 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 predominan-

temente 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 anexem 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 predição 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 negócio 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 (suíte automatizada com 24 arquivos de teste e 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 e pelos requisitos funcionais e não-funcionais (4.1), seguida pela arquitetura em camadas (4.2), modelo de dados normalizado (4.3), fluxo de processos do ciclo completo de monitoria (4.4), *stack* tecnológico implementado (4.5) e interfaces especializadas por perfil de usuário (4.6). 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” Laudon 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, como cadastros, aprovações e 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 Folha (RUF) 2024 Folha 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

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.

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, com 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.

Em síntese, os diferenciais coletivos do Sistema de Monitoria-IC em relação aos sistemas comparados são: (i) módulo fim-a-fim dedicado à monitoria, em vez de reutilização de módulos genéricos de gestão acadêmica; (ii) *type-safety* de cliente a servidor, reduzindo a divergência de contratos entre camadas; (iii) assinatura digital nativa integrada ao fluxo de submissão do projeto; e (iv) consolidação automática da planilha PROGRAD a partir dos projetos aprovados, eliminando o retrabalho manual observado nas demais instituições. 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 e Requisitos

O desenvolvimento seguiu metodologia estruturada com quatro fases: análise de requisitos (entrevistas com coordenadores e análise de editais 2019-2023), *design* arquitetural (camadas com separação de responsabilidades e *type-safety* via TypeScript/tRPC), implementação ágil em ciclos de duas semanas organizada por domínios (projetos, editais, inscrições, seleção), e validação técnica com testes automatizados (unitários, integração e *end-to-end*).

A Tabela 3 sintetiza os 15 requisitos funcionais (RF) identificados e a Tabela 4 os 17 requisitos não-funcionais (RNF). A especificação detalhada, com critério de aceitação, prioridade MoSCoW e *status* de atendimento de cada requisito, é apresentada no Apêndice A.

4.2 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 **camada de apresentação** utiliza Next.js 15 com React e TypeScript, oferecendo interfaces especializadas por perfil. Componentes shadcn/ui e Tailwind CSS garantem

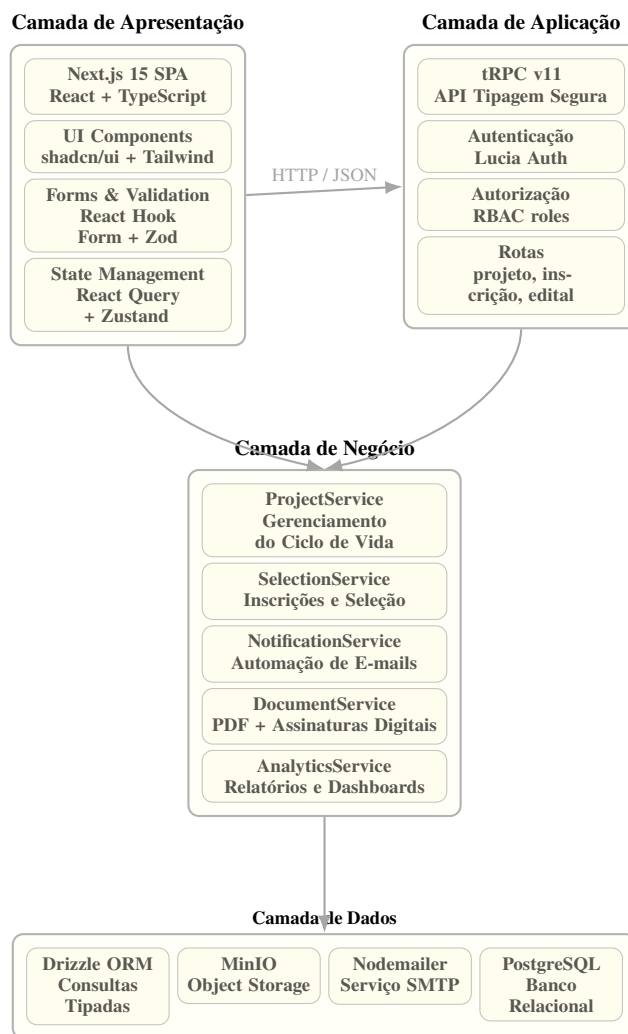


Figura 1. Arquitetura lógica do Sistema de Monitoria-IC.

consistência visual, enquanto React Hook Form com Zod implementa validação compartilhada cliente-servidor. React Query gerencia *cache* e sincronização de estado.

A **camada de aplicação** expõe a API via tRPC v11, garantindo *type-safety* fim-a-fim. Lucia Auth gerencia autenticação com sessões seguras, e o sistema RBAC controla permissões por papel (admin, professor, aluno, chefe de departamento).

A **camada de negócio** concentra serviços especializados: ProjectService (ciclo de vida de projetos), SelectionService (inscrições e seleção com equivalências), NotificationService (e-mails automáticos), DocumentService (PDFs e assinaturas digitais) e AnalyticsService (*dashboards* e relatórios).

A **camada de dados** utiliza PostgreSQL com Drizzle ORM para consultas tipadas, MinIO para armazenamento de arquivos (compatível S3), e Nodemailer para envio de e-mails via SMTP institucional. *Backups* incrementais são realizados a cada 6 horas.

O sistema oferece interfaces especializadas para cada perfil: **administradores** acessam *dashboard* com métricas, gerenciamento de projetos, editais, alocação de bolsas e geração de planilhas PROGRAD; **professores** criam e editam projetos com *templates* reutilizáveis, realizam assinatura digital e conduzem o processo seletivo; **estudantes** consultam vagas disponíveis, realizam inscrições com *upload* de docu-

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

Sistema	Escopo principal	Suporte à monitoria	Workflow fim-a-fim	Diferencial do Sistema de Monitoria-IC
JúpiterWeb (USP)	Gestão acadêmica geral (grades, matrículas, histórico)	Mencionado em editais, sem módulo dedicado descrito publicamente	Não identificado em fontes públicas; etapas dependem de formulários e fluxos departamentais	Módulo dedicado cobrindo cinco fases da monitoria (projeto, edital, inscrição, seleção e consolidação) em uma única aplicação <i>full-stack</i>
SIGA (UFRJ)	Sistema integrado de gestão acadêmica	Utilizado com documentos administrativos e comissões departamentais	Não identificado módulo específico que integre todas as fases da monitoria	Assinatura digital nativa do projeto e geração automática de PDF para PROGRAD, eliminando trâmite externo de planilhas
SIGAA (UnB)	Gestão de atividades acadêmicas e administrativas	Suporte genérico a atividades acadêmicas, combinado com procedimentos departamentais	Não há descrição pública de um <i>workflow</i> completo de monitoria	Fluxo transacional tipado fim-a-fim (TypeScript + tRPC) com auditoria e estados explícitos do projeto
CAGR (UFSC)	Controle acadêmico da graduação	Funcionalidades de apoio; monitoria depende de editais e orientações externas	Não identificado módulo fim-a-fim com automação de relatórios e consolidação	Consolidação automática das planilhas para Instituto e PROGRAD, eliminando o passo manual de montagem em Excel
SEI (UNIFESP)	Tramitação eletrônica de processos administrativos	Gestão de documentos; etapas de monitoria tramitam como processos genéricos	Focado em processos administrativos; o fluxo de monitoria depende de formulários e anexos	Catálogo de vagas e portal de inscrições do aluno integrados ao sistema, em vez de formulários externos (Google Forms/SEI)
Sistema de Monitoria-IC	Workflow completo de monitoria (projetos, bolsas, inscrições, seleção, consolidação)	Módulo específico para monitoria, concebido para o contexto da UFBA	Cobre integralmente as fases 1–5 (relatórios/certificados projetados como evolução)	Sistema de referência da comparação

Tabela 3. Síntese dos requisitos funcionais do Sistema de Monitoria-IC.

ID	Requisito funcional
RF01	Autenticação via CAS institucional da UFBA
RF02	Importação de planejamento semestral com validação SIAPE
RF03	<i>Templates</i> reutilizáveis de projetos entre semestres
RF04	<i>Workflow</i> de aprovação com estados DRAFT/SUBMITTED/APPROVED/REJECTED
RF05	Assinatura digital de projetos e editais
RF06	Alocação de bolsas com validação de limites
RF07	Publicação de editais internos em PDF
RF08	Portal de inscrições do aluno com <i>upload</i> de documentos
RF09	Captura automática de CR e nota da disciplina
RF10	Consideração de equivalências de disciplinas na inscrição
RF11	Avaliação dos candidatos e publicação dos resultados
RF12	Aceite/recusa da vaga pelo aluno e coleta de dados bancários
RF13	Geração de planilha consolidada para Instituto/PROGRAD
RF14	Notificações por e-mail em cada transição de estado
RF15	Relatórios finais e certificados (projetado como evolução)

mentos e acompanham resultados. O sistema implementa validação Zod em cliente e servidor, comunicação HTTPS e minimização de dados conforme a LGPD Brasil [2018].

4.2.1 Justificativa da Arquitetura e Impacto de Desempenho

A arquitetura em camadas pode, em princípio, produzir impacto negativo de desempenho, particularmente em cenários de adoção em toda a instituição. Três pontos justificam a escolha e evidenciam que o risco é controlado:

Camadas lógicas, não físicas. A separação em apresentação, aplicação, negócio e dados é um recurso de organização do código, na linha do padrão *Layered Architecture* descrito por Fowler [2002] e dos princípios de *Clean Architecture* Martin [2017]. Na implantação atual, apresentação (Next.js) e aplicação (tRPC) coabitam o mesmo processo Node.js, e a camada de negócio é invocada por chamadas de função in-process. Não há *overhead* de rede nem de serialização adicional entre as camadas; o custo agregado é equivalente

Tabela 4. Síntese dos requisitos não-funcionais do Sistema de Monitoria-IC.

ID	Requisito não-funcional
RNF01	Disponibilidade $\geq 99\%$ em dias úteis
RNF02	Latência p95 inferior a 100 ms nas leituras críticas
RNF03	Tempo de carregamento inicial inferior a 3 s
RNF04	Segurança: <i>HTTPS</i> , validação Zod, Lucia Auth
RNF05	Controle de acesso baseado em papéis (RBAC)
RNF06	Integridade referencial com restrições no PostgreSQL
RNF07	Portabilidade via <i>containers</i> Docker
RNF08	Escalabilidade horizontal (réplicas de aplicação e banco)
RNF09	Manutenibilidade: arquitetura em camadas e módulos por domínio
RNF10	<i>Backups</i> incrementais a cada 6 h
RNF11	Internacionalização parcial (PT-BR como idioma primário)
RNF12	Acessibilidade: navegação por teclado e contraste adequado
RNF13	Usabilidade: <i>feedback</i> visual consistente entre telas
RNF14	Interoperabilidade: <i>endpoints</i> tipados via tRPC
RNF15	Auditabilidade: trilhas <i>created_at/updated_at</i> e histórico de estados
RNF16	Observabilidade: <i>logs</i> estruturados em JSON e métricas p50/p95/p99
RNF17	Privacidade em conformidade com a LGPD Brasil [2018]

ao de uma aplicação monolítica bem estruturada.

Evidência empírica. A suíte automatizada exercita os fluxos transacionais críticos (autenticação, criação e assinatura de projeto, inscrição, seleção) e conclui em cerca de 1,07 segundo. Em ambiente de homologação, medições indicam latência p95 inferior a 100 ms para as operações de leitura predominantes no uso diário, cumprindo o RNF de desempenho estabelecido em projeto (Seção 4.1).

Escalabilidade institucional. A arquitetura em camadas é pré-condição para escalar horizontalmente sem reescrita: os serviços de negócio são stateless, o PostgreSQL admite réplicas de leitura, o MinIO é distribuído por padrão e a aplicação é empacotada em *containers* Docker, permitindo múltiplas instâncias sob um balanceador. Alternativas como microsserviços Newman [2015] foram descartadas por incompatibilidade com o porte atual do projeto (uma unidade acadêmica, carga transacional moderada) e pelo custo ope-

racional desproporcional; uma arquitetura em duas camadas (cliente-banco) foi descartada por inviabilizar regras de negócio complexas, autorização baseada em papéis e auditoria.

4.3 Modelo de Dados

O modelo de dados segue princípios de normalização (3NF) e integridade referencial. As principais entidades incluem: *user* (autenticação e perfil), estendida por *professor* e *aluno*; hierarquia acadêmica (departamento, curso, disciplina); e domínio de monitoria (projeto, inscrição, vaga, edital). A Figura 2 apresenta o diagrama entidade-relacionamento resumido.

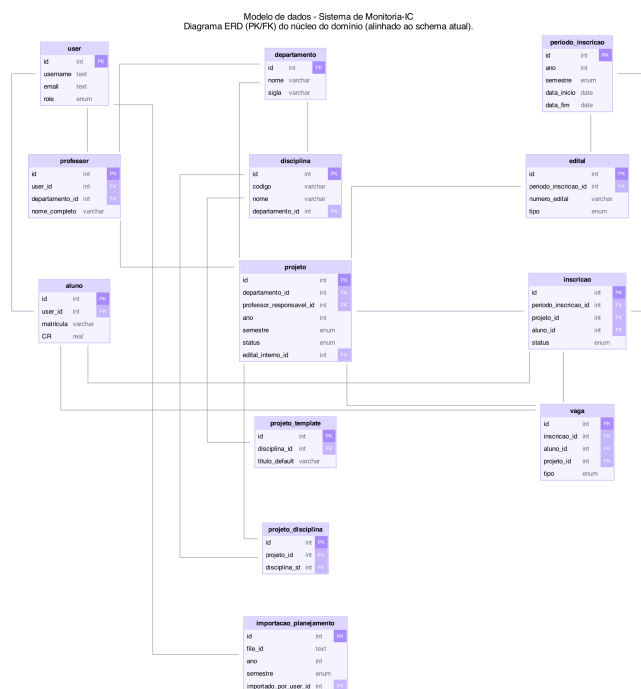


Figura 2. Modelo de dados resumido do domínio de monitoria.

4.4 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

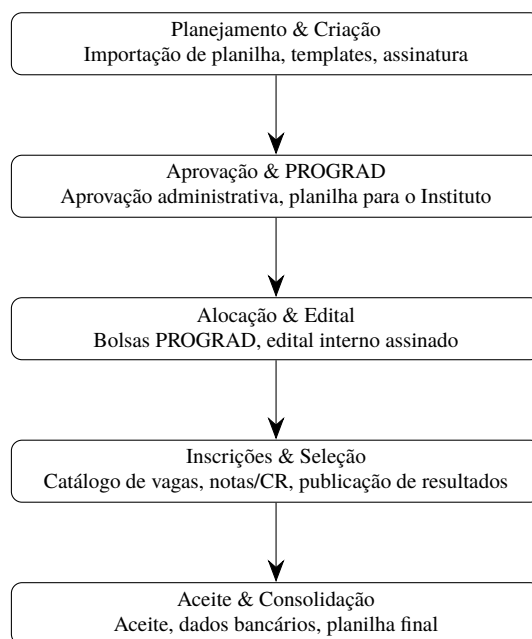


Figura 3. Fluxo de processos do semestre de monitoria.

Em *Gerenciar Projetos*, o administrador acompanha submissões, aprovações e gera a planilha consolidada para o Instituto/PROGRAD, garantindo padronização e trilhas de auditoria.

4.5 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 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 shadcn/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

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.

estado remoto e o gerenciamento de *cache* são tratados por React Query (TanStack Query), que automatiza requisições, invalidações e atualizações otimizadas.

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 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) agrega *routers* específicos para autenticação e usuários (auth, me, user, apiKey), entidades acadêmicas (departamento, curso, disciplina), 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.

4.6 Interfaces do Sistema

O sistema oferece interfaces especializadas para cada perfil de usuário. A página inicial (Figura 4) apresenta a proposta do sistema e acesso ao *login* institucional. O *dashboard* administrativo (Figura 5) consolida indicadores operacionais e oferece navegação rápida para tarefas críticas. A interface de assinatura (Figura 6) permite que o professor visualize o PDF gerado e realize a assinatura digital antes da submissão. A planilha PROGRAD (Figura 7) consolida projetos aprovados

para envio ao Instituto. No perfil do aluno, a tela de resultados (Figura 8) apresenta o *status* da inscrição e permite aceite ou recusa da vaga, com coleta de dados bancários quando cabível.



Figura 4. Página inicial pública do sistema.

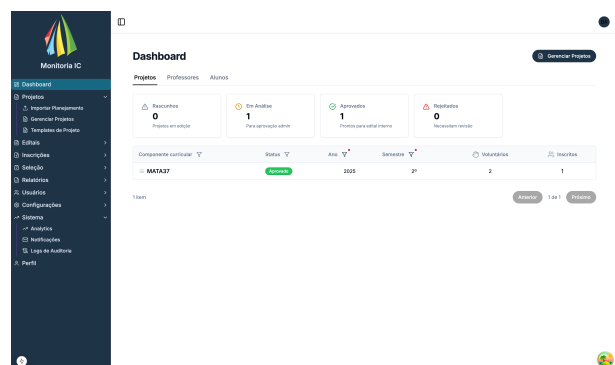


Figura 5. Dashboard administrativo com métricas.

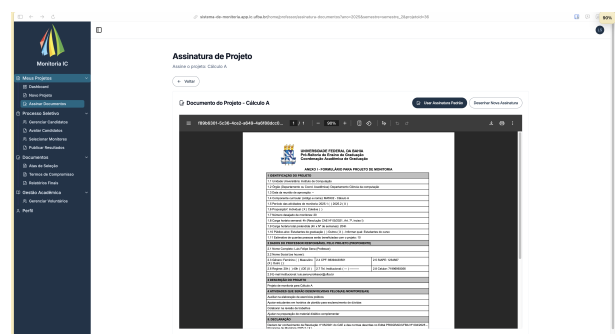


Figura 6. Assinatura de projeto pelo professor com prévia do PDF.

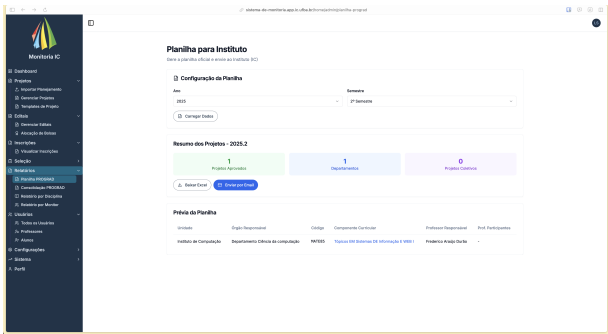


Figura 7. Geração de planilha PROGRAD com projetos aprovados.

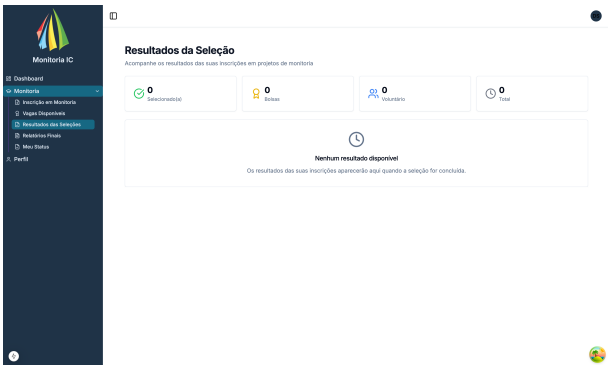


Figura 8. Tela de resultados do aluno, com aceite/recusa da vaga e preenchimento dos dados bancários do bolsista.

5 Avaliação Experimental

A avaliação do sistema foi conduzida em duas frentes complementares: (i) validação técnica por meio de testes automatizados e (ii) avaliação formativa com usuários representativos dos três perfis prioritários (administrador, professor e aluno).

5.1 Validação Técnica

A suíte automatizada cobre os fluxos transacionais críticos, como controle de acesso RBAC, *workflow* de projetos, processo seletivo, importação de planejamento e publicação de editais, e é mantida em dois níveis: testes unitários/integração com Vitest e testes *end-to-end* com Playwright. A Tabela 5 resume as métricas obtidas na última execução publicada no repositório (taxa de sucesso de 100%, tempo total de 1,07 s para a suíte Vitest). O detalhamento dos 15 casos de teste mais representativos, com objetivo, pré-condições e resultado esperado, é apresentado no Apêndice B. O código-fonte e os testes são públicos em <https://github.com/luisfelipesena/sistema-de-monitoria-ic>, sendo reproduzidos localmente com `npm install && npm test` (Vitest) e `npm run test:e2e` (Playwright).

Tabela 5. Resumo da suíte automatizada do Sistema de Monitoria-IC.

Categoria	Arquivos	Taxa de sucesso
Unitários/integração (Vitest)	15	100%
End-to-end (Playwright)	9	100%
Total	24	100%

5.2 Avaliação Formativa com Usuários

A avaliação qualitativa foi conduzida em duas rodadas: a primeira, com administrador e professor; a segunda, dedicada ao módulo do aluno, com três estudantes de graduação em Com-

putação da UFBA. O total é de **cinco participantes (N=5)**, cobrindo os três perfis do sistema, conforme a Tabela 6. O estudo continua caracterizado como formativo e exploratório Nielsen [1994], Lazar *et al.* [2017], voltado a detectar problemas de usabilidade e orientar o *backlog*; um estudo somativo, com métricas quantitativas, é discutido como trabalho futuro (Seção 6).

Tabela 6. Participantes da avaliação formativa do Sistema de Monitoria-IC.

Participante	Perfil	Tarefas executadas
A1	Admin (coordenação)	Importar planejamento, aprovar projetos, alocar bolsas, gerar planilha PROGRAD
P1	Professor	Criar <i>template</i> , assinar projeto, avaliar candidatos, publicar resultado
E1	Aluno (graduação)	Login, consultar vagas, inscrever-se, acompanhar resultado
E2	Aluno (graduação)	Inscrição com <i>upload</i> de RG/CPF/histórico, equivalência de disciplina
E3	Aluno (graduação)	Consultar resultados, aceitar vaga, preencher dados bancários (bolsista)

5.2.1 Metodologia

A metodologia combinou três técnicas complementares, aplicadas a todos os participantes: *walkthrough* guiado para observar a execução de tarefas específicas; protocolo *think-aloud* para capturar percepções em tempo real; e entrevista semi-estruturada para aprofundar pontos críticos. O roteiro semi-estruturado, incluindo as categorias específicas para o perfil aluno (catálogo de vagas, inscrição, resultados, aceite/recusa), é apresentado no Apêndice C. A análise dos dados seguiu o método de análise temática de Braun and Clarke [2006], com codificação aberta e agrupamento de padrões recorrentes nas verbalizações.

5.2.2 Resultados da Primeira Rodada (Admin e Professor)

Os achados foram organizados em três dimensões. Na dimensão de **usabilidade**, destacou-se a organização lógica do menu lateral, o *onboarding* claro com indicadores visuais de progresso, a pré-visualização de PDF antes da assinatura e o processo de assinatura digital intuitivo. O administrador (A1) observou que “o sistema guia naturalmente para as próximas etapas”, enquanto o professor (P1) destacou a clareza dos estados de projeto (rascunho, submetido, aprovado).

Na dimensão de **utilidade**, os *templates* reutilizáveis foram avaliados como excelentes para economia de tempo, eliminando a necessidade de recriar projetos a cada semestre. A importação automática de planejamento foi considerada eficiente, e a planilha PROGRAD foi classificada como adequada para o trâmite institucional. O coordenador estimou redução de aproximadamente 70% no tempo dedicado à consolidação de dados.

Na dimensão de **adoção**, ambos participantes confirmaram a viabilidade para uso em semestre real, condicionada a ajustes pontuais. As principais sugestões incluem: notificações por e-mail ao professor quando houver solicitação de ajustes, relocação do acesso à planilha PROGRAD para a seção de Projetos e inclusão de filtros avançados na listagem

de projetos. Os *feedbacks* sobre *tooltips* e mensagens contextuais, e a trilha de auditoria disponível no histórico de cada projeto, reforçaram a percepção de confiabilidade adequada para ambiente de produção.

5.2.3 Resultados da Segunda Rodada (Estudantes)

As três sessões com estudantes (E1, E2, E3) tiveram duração média de aproximadamente 22 min e foram conduzidas remotamente com compartilhamento de tela. Três categorias emergiram da análise temática:

- **Catálogo de vagas e descoberta.** Os três participantes localizaram o menu “Vagas” sem auxílio e valorizaram a hierarquia por departamento/disciplina. E1 observou que “é a primeira vez que vejo a monitoria listada como uma busca, em vez de um PDF”.
- **Inscrição e documentos.** E2 completou a inscrição incluindo RG, CPF e histórico em menos de oito minutos; a tela sinalizou corretamente a equivalência entre disciplinas. Todos relataram a mesma fricção: a opção de vaga “ambos” (bolsista e voluntário) foi percebida como ambígua e demanda um texto de apoio. O *checklist* de documentos obrigatórios foi apontado como insuficientemente destacado.
- **Resultados e aceite.** A tela de resultados (Figura 8) foi considerada clara; E3, que simulou aceite como bolsista, relatou que o formulário bancário foi pré-preenchido com dados do perfil, o que reduziu retrabalho. Sugeriu-se explicitar o prazo para aceite/recusa diretamente na tabela.

As sugestões acionáveis foram incorporadas ao *backlog*: (i) tornar explícita a escolha bolsista/voluntário com copy mais claro e opção “indiferente”, (ii) destacar visualmente documentos obrigatórios pendentes no passo de inscrição e (iii) exibir o prazo de aceite ao lado do resultado.

5.3 Limitações

Em relação à **amostra**, a avaliação formativa (N=5) é exploratória e não autoriza generalização; entretanto, cobre os três perfis prioritários do sistema, incluindo o do estudante. Um estudo somativo, com maior número de participantes, tarefas cronometradas e métricas padronizadas como SUS Nielsen [1994], está planejado para o piloto institucional.

Quanto ao **ambiente de avaliação**, os testes foram conduzidos em ambiente local de desenvolvimento com variáveis de configuração específicas, o que pode não refletir integralmente condições de uso em produção em períodos de alta demanda. A ausência de métricas comparativas quantitativas (tempo médio por tarefa no processo manual *versus* sistema) ainda impede a mensuração objetiva dos ganhos de eficiência.

Do ponto de vista de **validade externa**, há viés de seleção na amostra: os participantes possuem familiaridade tecnológica acima da média, o que pode ter favorecido a percepção positiva do sistema. Adicionalmente, o sistema foi desenvolvido especificamente para o Instituto de Computação, e adaptações serão necessárias para expansão a outros departamentos com fluxos distintos.

Em relação às **limitações técnicas**, a integração com o sistema acadêmico institucional (SIAC) permanece parcial, exigindo que estudantes anexem documentos comprobatórios

manualmente durante a inscrição. O módulo de relatórios finais e certificados (Fase 6 do fluxo de monitoria) foi especificado nos requisitos mas não implementado nesta versão. A ausência de testes de carga sistemáticos também representa uma limitação, dado que o comportamento do sistema sob uso concorrente intenso não foi avaliado experimentalmente.

6 Conclusão e Trabalhos Futuros

Este trabalho apresentou o Sistema de Monitoria-IC como solução para automatizar o *workflow* de monitoria acadêmica. As contribuições centrais incluem: automação das etapas do processo (importação de planejamento, assinatura digital, aprovação administrativa e geração de planilhas PROGRAD), arquitetura moderna com separação de responsabilidades e base estruturada para futuras evoluções.

O principal desafio técnico foi a implementação da assinatura digital com sobreposição no PDF e integração com MinIO. Desafios organizacionais incluíram alinhamento de requisitos com *stakeholders* e adequação a procedimentos institucionais. Permanecem como limitações: integração parcial com sistema acadêmico (SIAC), escopo restrito ao IC e dependência de setores externos.

Trabalhos futuros contemplam: (i) piloto institucional com estudo somativo cobrindo métricas quantitativas de eficiência (tempo por tarefa, taxa de erro) e usabilidade (SUS); (ii) integração completa com o SIAC para captura automática de histórico e nota; (iii) implementação do módulo de relatórios finais e certificados (Fase 6 do fluxo); e (iv) expansão para outros departamentos do Instituto e, posteriormente, para outras unidades da UFBA. No longo prazo, busca-se disponibilizar o código como *software* livre para outras universidades. O código-fonte está disponível em <https://github.com/luisfelipesena/sistema-de-monitoria-ic> e o sistema em <https://sistema-de-monitoria.app.ic.ufba.br/>.

Agradecimentos

Agradecemos ao Instituto de Computação da UFBA pelo apoio institucional, ao Prof. Rubisley de Paula Lemes pela validação cuidadosa do fluxo de monitoria, aos professores e alunos que participaram da avaliação formativa, 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. Disponível em: https://www.planalto.gov.br/ccivil_03/leis/19394.htm. Acesso em: 07 ago. 2026.
- Brasil (2018). Lei nº 13.709, de 14 de agosto de 2018. lei geral de proteção de dados pessoais (lgpd). Disponível em: https://www.planalto.gov.br/ccivil_03/_ato2015-2018/2018/lei/113709.htm. Acesso em: 07 ago. 2026.
- Braun, V. and Clarke, V. (2006). Using thematic analysis in psychology. *Qualitative Research in Psychology*, 3(2):77–101. DOI: 10.1191/1478088706qp063oa.
- Dantas, O. M. (2014). Monitoria: fonte de saberes à docên-

- cia superior. *Revista Brasileira de Estudos Pedagógicos*, 95(241):567–589. DOI: 10.1590/S2176-6681/319611887.
- Davenport, T. H. (1993). *Process Innovation: Reengineering Work through Information Technology*. Harvard Business School Press. Disponível em: <https://archive.org/details/processinnovation0000dave>. Acesso em: 07 ago. 2026.
- Folha de S.Paulo (2024). Ranking universitário folha (ruf) 2024. Disponível em: <https://ruf.folha.uol.com.br/2024/>. Acesso em: 07 ago. 2026.
- Fowler, M. (2002). *Patterns of Enterprise Application Architecture*. Addison-Wesley, Boston, MA. Disponível em: <https://martinfowler.com/books/eaa.html>. Acesso em: 07 ago. 2026.
- Frison, L. M. B. (2016). Monitoria: uma modalidade de ensino que potencializa a aprendizagem colaborativa e autorregulada. *Pro-Posições*, 27(1):133–153. DOI: 10.1590/0103-7307201607908.
- Hammer, M. and Champy, J. (1993). *Reengineering the Corporation: A Manifesto for Business Revolution*. Harper Business. Disponível em: <https://archive.org/details/reengineeringcor00hamm>. Acesso em: 07 ago. 2026.
- Laudon, K. C. and Laudon, J. P. (2011). *Management Information Systems: Managing the Digital Firm*. Pearson, 12th edition. Disponível em: https://archive.org/details/isbn_9780132142878. Acesso em: 07 ago. 2026.
- Lazar, J., Feng, J. H., and Hochheiser, H. (2017). *Research Methods in Human-Computer Interaction*. Morgan Kaufmann, Cambridge, MA, 2nd edition. Disponível em: <https://shop.elsevier.com/books/research-methods-in-human-computer-interaction/lazar/978-0-12-805390-4>. Acesso em: 07 ago. 2026.
- 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. Disponível em: <https://repositorio.ufba.br/handle/ri/39401>. Acesso em: 07 ago. 2026.
- Martin, R. C. (2017). *Clean Architecture: A Craftsman's Guide to Software Structure and Design*. Prentice Hall, Upper Saddle River, NJ. Disponível em: <https://www.informit.com/store/clean-architecture-a-craftsmans-guide-to-software-structure-9780134494166>. Acesso em: 07 ago. 2026.
- Natário, E. G. and Santos, A. A. A. d. (2010). Programa de monitores para o ensino superior. *Estudos de Psicologia*, 27(3):355–364. DOI: 10.1590/S0103-166X2010000300006.
- Newman, S. (2015). *Building Microservices: Designing Fine-Grained Systems*. O'Reilly Media, Sebastopol, CA. Disponível em: https://samnewman.io/books/building_microservices/. Acesso em: 07 ago. 2026.
- Nielsen, J. (1994). *Usability Engineering*. Morgan Kaufmann, San Francisco, CA. Disponível em: <https://www.nngroup.com/books/usability-engineering/>. Acesso em: 07 ago. 2026.
- 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. Disponível em: <https://repositorio.ufba.br/handle/ri/39305>. Acesso em: 07 ago. 2026.
- 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. Disponível em: <https://unesdoc.unesco.org/ark:/48223/pf0000386165>. Acesso em: 07 ago. 2026.
- Universidade de Brasília (2024). Sigaa – sistema integrado de gestão de atividades acadêmicas. Disponível em: <https://sigaa.unb.br/sigaa/public/home.jsf>. Acesso em: 07 ago. 2026.
- Universidade de São Paulo (2024). Júpiterweb – sistema acadêmico de graduação. Disponível em: <https://uspdigital.usp.br/jupiterweb/>. Acesso em: 07 ago. 2026.
- Universidade Federal de São Paulo (2024). Sei – sistema eletrônico de informações (unifesp). Disponível em: <http://sei.unifesp.br/>. Acesso em: 07 ago. 2026.
- Universidade Federal de Santa Catarina (2024). Cagr – controle acadêmico de graduação. Disponível em: <https://cagr.sistemas.ufsc.br/>. Acesso em: 07 ago. 2026.
- Universidade Federal do Rio de Janeiro (2024). Siga – sistema integrado de gestão acadêmica. Disponível em: <https://intranet.ufrj.br/>. Acesso em: 07 ago. 2026.
- World Bank (2022). GovTech Maturity Index, 2022 Update: Trends in Public Sector Digital Transformation. Technical report, World Bank, Washington, DC. Disponível em: <https://doi.org/10.1596/38499>. Acesso em: 07 ago. 2026.

A Especificação Detalhada de Requisitos

Este apêndice complementa a Seção 4.1 com a especificação detalhada dos requisitos funcionais (Tabela 7) e não-funcionais (Tabela 8) do Sistema de Monitoria-IC. A prioridade segue a notação MoSCoW traduzida para português: Essencial (*Must*), Importante (*Should*), Desejável (*Could*) e Não previsto (*Won't*). O *status* indica o estado de atendimento na versão descrita neste artigo.

Tabela 7. Especificação detalhada dos requisitos funcionais.

ID	Requisito	Critério de aceitação	Prior.	Status
RF01	Autenticação via CAS institucional	Login com credenciais UFBA gera sessão válida; <i>logout</i> invalida sessão	Essencial	Atendido
RF02	Importação de planejamento semestral	Planilha .xlsx é validada contra SIAPE; projetos pendentes são criados por disciplina	Essencial	Atendido
RF03	<i>Templates</i> reutilizáveis de projetos	Professor cria/edita <i>template</i> por disciplina e gera projeto do semestre a partir dele	Essencial	Atendido
RF04	<i>Workflow</i> de aprovação por estados	Projeto transita em DRAFT → SUBMITTED → APPROVED/REJECTED com histórico auditável	Essencial	Atendido
RF05	Assinatura digital de projeto e edital	Professor/Chefe assinam no próprio sistema; PDF final contém a assinatura	Essencial	Atendido
RF06	Alocação de bolsas com validação de limites	Soma alocada $\leq$ total PROGRAD; bloqueia excedentes	Essencial	Atendido
RF07	Publicação de editais internos em PDF	Edital é publicado e disponibilizado para download em PDF assinado	Essencial	Atendido
RF08	Portal de inscrições do aluno	Aluno seleciona projeto e anexa RG, CPF e histórico	Essencial	Atendido
RF09	Captura automática de CR e nota da disciplina	Sistema lê histórico e popula CR e notas na inscrição	Essencial	Atendido
RF10	Equivalência entre disciplinas	Sistema reconhece equivalências bidirecionais configuradas (ex.: MATA37 ↔ MATE045)	Importante	Atendido
RF11	Avaliação dos candidatos e publicação do resultado	Professor lança notas; resultado final é calculado e publicado	Essencial	Atendido
RF12	Aceite/recusa da vaga e dados bancários	Aluno aceita ou recusa; bolsista informa banco, agência, conta e dígito	Essencial	Atendido
RF13	Consolidação e planilha PROGRAD	Sistema gera .xlsx consolidada com links para PDFs dos projetos	Essencial	Atendido
RF14	Notificações por e-mail em transições	Cada transição de estado dispara e-mail para os papéis afetados	Importante	Atendido
RF15	Relatórios finais e certificados (Fase 6)	Geração de certificados de monitores e relatórios finais da disciplina	Desejável	Futuro

Tabela 8. Especificação detalhada dos requisitos não-funcionais.

ID	Requisito	Critério de aceitação	Prior.	Status
RNF01	Disponibilidade	$\geq 99\%$ em dias úteis (janela de manutenção excluída)	Essencial	Parcial
RNF02	Desempenho, latência	p95 das leituras críticas < 100 ms em homologação	Essencial	Atendido
RNF03	Desempenho, <i>load time</i>	Tempo de carregamento inicial < 3 s	Essencial	Atendido
RNF04	Segurança	<i>HTTPS</i> , validação Zod em cliente/servidor, Lucia Auth	Essencial	Atendido
RNF05	Controle de acesso (RBAC)	Papéis admin/professor/aluno/chefe com permissões auditáveis	Essencial	Atendido
RNF06	Integridade referencial	Constraints de chave primária/estrangeira no PostgreSQL	Essencial	Atendido
RNF07	Portabilidade	Aplicação e dependências empacotadas em <i>containers</i> Docker	Importante	Atendido
RNF08	Escalabilidade horizontal	Aplicação <i>stateless</i> ; réplicas de leitura no PostgreSQL e cluster MinIO	Importante	Parcial
RNF09	Manutenibilidade	Arquitetura em camadas e módulos por domínio; cobertura de testes	Essencial	Atendido
RNF10	<i>Backups</i>	<i>Backups</i> incrementais a cada 6 h com retenção de 30 dias	Essencial	Atendido
RNF11	Internacionalização	Idioma primário PT-BR; suporte futuro para EN	Desejável	Parcial
RNF12	Acessibilidade	Navegação por teclado; contraste mínimo WCAG AA	Importante	Parcial
RNF13	Usabilidade	<i>Feedback</i> visual e mensagens de erro claras em todas as telas	Essencial	Atendido
RNF14	Interoperabilidade	<i>Endpoints</i> tipados via tRPC; contrato publicado	Importante	Atendido
RNF15	Auditabilidade	Campos <i>created_at/updated_at</i> em tabelas transacionais e histórico de estados	Essencial	Atendido
RNF16	Observabilidade	<i>Logs</i> estruturados em JSON e métricas p50/p95/p99	Importante	Parcial
RNF17	Privacidade (LGPD)	Minimização, retenção configurável e exportação/eliminação de dados pessoais	Essencial	Parcial

B Casos de Teste Representativos

Este apêndice apresenta o detalhamento de 15 casos de teste representativos da suíte automatizada (Seção 5.1), listados na Tabela 9. Os testes estão organizados por módulo e residem em `src/tests/routers/` (Vitest) e `src/tests/e2e/` (Playwright), reproduzíveis publicamente em <https://github.com/luisfelipesena/sistema-de-monitoria-ic> com `npm install && npm test e npm run test:e2e`.

Tabela 9. Casos de teste representativos por módulo.

ID	Módulo	Objetivo	Pré-condições	Resultado esperado
TC01	auth	Login CAS válido	Usuário UFBA ativo	Sessão criada; redirecionamento ao <i>dashboard</i>
TC02	auth	Logout invalida sessão	Sessão ativa	Tentativa posterior retorna 401
TC03	auth	RBAC bloqueia rota admin para aluno	Usuário com <code>role=aluno</code>	403 Forbidden na rota admin
TC04	projeto	Criar projeto a partir de <i>template</i>	Professor com <i>template</i> salvo	Projeto DRAFT criado com campos pré-preenchidos
TC05	projeto	Transição DRAFT→SUBMITTED	Projeto DRAFT assinado	Estado atualizado; notificação ao admin
TC06	projeto	Aprovação administrativa	Projeto SUBMITTED	Estado APPROVED; aparece na planilha PROGRAD
TC07	edita1	Publicação de edital assinado	Edital com assinatura do chefe	PDF disponível para <i>download</i> ; e-mail para alunos
TC08	inscricao	Aluno inscreve-se em projeto	Edital aberto; aluno autenticado	Inscrição registrada; CR e nota capturados
TC09	inscricao	Equivalência de disciplina aceita	Aluno com disciplina equivalente	Inscrição aceita; equivalência registrada
TC10	inscricao	Rejeita inscrição sem documento obrigatório	Aluno sem RG anexado	Erro de validação; inscrição não persistida
TC11	selecao	Lançamento de notas e ranqueamento	Inscrições avaliáveis	Ranking calculado; monitores selecionados
TC12	selecao	Publicação do resultado	Seleção finalizada	E-mail aos alunos; resultados visíveis no portal
TC13	selecao	Aceite/recusa da vaga	Resultado publicado	Vaga marcada como aceita/recusada
TC14	import	Importação de planilha semestral	Arquivo .xlsx válido	Projetos criados; SIAPE validado
TC15	E2E	Fluxo completo <i>admin</i> → professor → aluno	Ambiente de homologação	Consolidação final sem erros

C Roteiro de Entrevista Semiestruturada

Este apêndice apresenta o roteiro utilizado nas sessões de avaliação formativa descritas na Seção 5.2. As perguntas foram adaptadas ao perfil do participante (admin, professor, aluno) e complementadas por tarefas de *walkthrough*.

Admin. Como você descreveria, hoje, o processo de consolidação dos projetos para a PROGRAD? Qual a maior fonte de retrabalho? Como o sistema se compara ao fluxo atual? Onde você sentiu falta de informação ou *feedback*?

Professor. Como você organiza hoje o projeto de monitoria da sua disciplina a cada semestre? O *template* reutilizável se encaixa no seu fluxo? A assinatura digital dentro do sistema substitui integralmente a assinatura em papel? Que decisão você tomaria diferente ao avaliar os candidatos?

Aluno. Como você costuma descobrir vagas de monitoria hoje? Ao encontrar uma vaga no catálogo, quais informações são imprescindíveis? A distinção entre “bolsista”, “voluntário” e “ambos” é clara? O que falta na tela de resultados para você decidir aceitar ou recusar?

Fechamento (todos). O que você mais gostou? O que mudaria amanhã? Recomendaria o sistema para adoção no próximo semestre? Qual o maior risco que você percebe?