www.revistaft.com.br

RevistaFT | ISSN 1678-0817 | v.30, n. 163, 2026 | DOI: 10.69849/ga7vc492

EduControle: sistema web para controle de entrada, saída e presença de alunos no Instituto Assistencial Dom Bosco

EduControle: a web system for student entry, exit and attendance control at Instituto Assistencial Dom Bosco

Rafael Sagan Souza[1]
Paulo Henrique Miler Mariano[2]
Nicolas Marcelo Belo Moreira[3]
Gabriel Christopher Dall'Pozzo[4]

RESUMO: Este relato descreve o desenvolvimento e a implantação do EduControle, sistema web de controle de entrada, saída e presença dos 180 alunos do Instituto Assistencial Dom Bosco, em Guarapuava, Paraná, ação de extensão de acadêmicos de Engenharia de Software. O controle era manual, o que gerava perda de informações e dificuldade de consulta ao histórico. Conduzido com Scrum, o sistema foi construído em React, TypeScript e Supabase, com controle de acesso por políticas de segurança no PostgreSQL, e validado pelos funcionários em piloto com uso assistido. O piloto ocorreu sem falhas críticas e evidenciou pendências de segurança, monitoramento e integridade na importação dos registros do relógio de ponto. Conclui-se que a arquitetura adotada foi adequada a uma instituição sem equipe própria de TI.

Palavras-chave: Extensão universitária; Controle de frequência; Engenharia de software; Sistemas de informação; Gestão educacional.

ABSTRACT: This report describes the development and deployment of EduControle, a web system for controlling the entry, exit and attendance of the 180 students of Instituto Assistencial Dom Bosco, in Guarapuava, Paraná, Brazil, as a university extension initiative by Software Engineering students. Control was manual, causing information loss and difficulty in consulting attendance history. Conducted with Scrum, the system was built with React, TypeScript and Supabase, with access control through security policies in PostgreSQL, and validated by staff in a pilot with assisted use. The pilot ran without critical failures and revealed pending security, monitoring and data integrity issues in the import of time clock records. The adopted architecture proved suitable for an institution without its own IT team.

[1] Acadêmico, Engenharia de Software, 8º período, Centro Universitário Campo Real.
[2] Acadêmico, Engenharia de Software, 8º período , Centro Universitário Campo Real.
[3] Acadêmico, Engenharia de Software, 8º período, Centro Universitário Campo Real.
[4] Especialista em Engenharia de Software, Professor Orientador, Centro Universitário Campo Real.

Keywords: University extension; Attendance control; Software engineering; Information systems; Educational management.

1 Introdução

A extensão universitária constitui, ao lado do ensino e da pesquisa, um dos pilares da educação superior brasileira. A Política Nacional de Extensão Universitária a define como um processo interdisciplinar, educativo, cultural, científico e político que promove a interação transformadora entre a universidade e outros setores da sociedade (FORPROEX, 2012). Com a Resolução CNE/CES nº 7, de 2018, essas atividades passaram a compor, no mínimo, 10% da carga horária curricular dos cursos de graduação (BRASIL, 2018b), o que tornou a extensão parte obrigatória da formação, e não apenas uma atividade complementar.

Na Engenharia de Software, a extensão tem um valor formativo particular: aproxima o estudante de problemas reais, com usuários reais, restrições de orçamento, prazos e limitações de infraestrutura que dificilmente aparecem em exercícios de sala de aula. Foi nesse contexto, no âmbito do programa institucional Engenharia Integral do Centro Universitário Campo Real, que se desenvolveu o projeto relatado neste artigo.

A entidade beneficiada foi o Instituto Assistencial Dom Bosco, de Guarapuava, Paraná, que atua há mais de 40 anos oferecendo cursos gratuitos a jovens e adolescentes e atende atualmente 180 alunos regularmente matriculados. Até o início do projeto, o registro de entrada, saída e frequência dos estudantes era feito de forma manual e pouco estruturada, o que gerava perda de informações, erros de registro e dificuldade para consultar o histórico de frequência. A instituição dispunha ainda de um relógio de ponto físico que permanecia ocioso pela ausência de profissional capacitado para operá-lo, já que o instituto não possui equipe própria de tecnologia da informação.

Esse cenário definiu o problema central do projeto: como informatizar o controle de presença de uma instituição assistencial sem equipe técnica própria, de modo que a solução fosse simples de operar, de baixo custo de manutenção e que aproveitasse os recursos já existentes. A ausência de equipe de TI não é um detalhe operacional, mas uma restrição que condicionou as principais decisões de arquitetura do sistema.

Diante disso, este artigo tem como objetivo relatar a experiência de concepção, desenvolvimento e implantação do EduControle, sistema web de controle de entrada, saída e

presença de alunos, discutindo as decisões técnicas adotadas, os resultados do piloto de implantação e as limitações identificadas. Como objetivos específicos, busca-se: descrever o contexto institucional que motivou a ação; apresentar as etapas metodológicas do projeto; analisar as principais decisões de arquitetura e seus *trade-offs*; relatar os resultados do piloto e a percepção dos usuários; e registrar as limitações e recomendações para a continuidade da solução. O texto está organizado em cinco seções: após esta introdução, a seção 2 apresenta a fundamentação teórica; a seção 3 descreve a metodologia; a seção 4 apresenta e discute os resultados; e a seção 5 traz as considerações finais.

2 Revisão da Literatura

2.1 Extensão universitária e formação em engenharia de software

A Política Nacional de Extensão Universitária estabelece cinco diretrizes para as ações extensionistas: interação dialógica; interdisciplinaridade e interprofissionalidade; indissociabilidade entre ensino, pesquisa e extensão; impacto na formação do estudante; e impacto e transformação social (FORPROEX, 2012). A interação dialógica é especialmente relevante em projetos de software, pois pressupõe que a solução seja construída com a comunidade, e não apenas entregue a ela. Na prática, isso se traduz em participação ativa da instituição no levantamento de requisitos, na validação das funcionalidades e na avaliação do produto final.

A Resolução CNE/CES nº 7/2018 reforça essa perspectiva ao estabelecer que atividades de extensão devem envolver diretamente as comunidades externas às instituições de ensino superior e estar vinculadas à formação do estudante (BRASIL, 2018b). Para um curso de Engenharia de Software, um projeto de extensão bem conduzido permite exercitar, em um contexto real, competências como engenharia de requisitos, projeto de arquitetura, gestão do projeto, implantação e transmissão de conhecimento ao usuário final.

2.2 Métodos ágeis, Scrum e produto mínimo viável

Sommerville (2018) aponta que os métodos ágeis são particularmente adequados ao desenvolvimento de sistemas de pequeno e médio porte, em contextos nos quais os requisitos tendem a mudar e há possibilidade de envolvimento próximo do cliente ao longo do processo.

Essas características correspondem ao cenário de um projeto de extensão, em que a equipe é pequena, o prazo é delimitado pelo calendário acadêmico e a compreensão do problema se forma à medida que a instituição tem contato com as versões funcionais do sistema.

O Scrum é um framework leve que organiza o trabalho em ciclos de duração fixa, de até um mês, denominados *sprints*, ao final dos quais se espera um incremento utilizável do produto (SCHWABER; SUTHERLAND, 2020). O Time Scrum é composto por um *Product Owner*, responsável por maximizar o valor do produto e gerenciar o *backlog*; um *Scrum Master*, responsável pela eficácia do time e pela adoção do framework; e os desenvolvedores, responsáveis pela construção do incremento (SCHWABER; SUTHERLAND, 2020).

Complementarmente, o conceito de produto mínimo viável, ou MVP (*Minimum Viable Product*), proposto por Ries (2012), designa a versão do produto que permite completar um ciclo de aprendizagem com o menor esforço possível. Em um projeto com prazo restrito, delimitar o MVP é uma decisão de escopo: define o que precisa estar funcionando para gerar valor à instituição e o que pode ser adiado sem comprometer esse valor.

2.3 Arquitetura de aplicações web e backend como serviço

Para Sommerville (2018), as decisões de projeto arquitetural influenciam diretamente atributos não funcionais do sistema, como desempenho, segurança, disponibilidade e manutenibilidade, razão pela qual devem considerar o contexto em que o software será operado. Pressman e Maxim (2016) destacam, no mesmo sentido, que a engenharia de requisitos estabelece a base sobre a qual o projeto e a construção serão realizados, de modo que restrições do ambiente devem ser identificadas antes das escolhas de tecnologia.

Uma alternativa arquitetural cada vez mais comum para sistemas de pequeno porte é o modelo de *backend* como serviço (BaaS, de *Backend as a Service*), no qual funcionalidades como banco de dados, autenticação e interface de acesso aos dados são fornecidas por uma plataforma gerenciada. Nesse modelo, a aplicação cliente, frequentemente uma aplicação de página única (SPA, de *Single Page Application*), comunica-se diretamente com o serviço, sem uma API própria intermediária. O ganho é a redução da infraestrutura a ser mantida; o custo é que regras de autorização, que em uma arquitetura tradicional ficariam em um servidor de aplicação, precisam ser garantidas na própria camada de dados.

No PostgreSQL, esse controle pode ser implementado por meio de políticas de segurança em nível de linha (RLS, de *Row Level Security*), que restringem, por usuário ou papel, quais linhas de uma tabela podem ser consultadas ou modificadas. Quando a RLS está habilitada em uma tabela e nenhuma política foi definida, o comportamento padrão é negar o acesso a todas as linhas (POSTGRESQL GLOBAL DEVELOPMENT GROUP, 2026). A documentação do Supabase recomenda habilitar a RLS em todas as tabelas expostas pela sua API, justamente porque a chave de acesso utilizada pelo cliente é pública e, sem as políticas adequadas, os dados ficariam acessíveis a qualquer pessoa que a obtivesse (SUPABASE, 2026).

2.4 Proteção de dados pessoais de adolescentes

Sistemas que registram presença e horários de alunos tratam dados pessoais e, no caso de instituições que atendem jovens e adolescentes, dados de menores de idade. A Lei Geral de Proteção de Dados Pessoais determina que o tratamento de dados de crianças e adolescentes seja realizado em seu melhor interesse (BRASIL, 2018a, art. 14) e que os agentes adotem medidas de segurança, técnicas e administrativas, aptas a proteger esses dados de acessos não autorizados (BRASIL, 2018a, art. 46). Esse enquadramento reforça a importância do controle de acesso por perfil e da gestão de credenciais, aspectos discutidos na seção 4.

3 Metodologia

Este trabalho caracteriza-se como um relato de experiência, de natureza aplicada e abordagem qualitativa, referente a um projeto de extensão executado entre 6 de março e 16 de setembro de 2026. A equipe foi composta por três acadêmicos do 8º período do curso de Engenharia de Software do Centro Universitário Campo Real, que assumiram os papéis de *Product Owner*, *Scrum Master* e desenvolvedor, sob orientação de um professor do curso.

O desenvolvimento seguiu o framework Scrum e foi organizado nas seguintes etapas: levantamento de requisitos junto à instituição, por meio de reuniões com a coordenação e com os funcionários envolvidos no controle de presença; definição do escopo do MVP; implementação incremental em *sprints*; validação por meio de testes manuais e de integração; treinamento presencial dos usuários, realizado entre 12 e 14 de agosto de 2026; piloto de implantação com uso assistido, entre 19 e 21 de agosto de 2026; e entrega formal à instituição, em 16 de setembro de 2026, registrada em termo de entrega e recebimento.

O piloto foi estruturado em três dias. No primeiro, realizaram-se a publicação do *frontend*, a configuração do *backend* com dados de amostra e a validação das variáveis de ambiente e das políticas de acesso. No segundo, foram configuradas as estações de trabalho da secretaria e da recepção, sobre o hardware já existente na instituição, e executados testes de fumaça (*smoke tests*) das funcionalidades críticas: autenticação, cadastro de alunos, registro de entrada e saída e geração de relatórios. No terceiro, os funcionários utilizaram o sistema com suporte presencial e remoto da equipe, e foram coletadas percepções de uso.

O levantamento de requisitos foi realizado em cerca de cinco reuniões on-line com a coordenação e com os funcionários responsáveis pelo controle de presença. As observações obtidas nessas reuniões, no treinamento e no piloto foram registradas diretamente no quadro Scrum da equipe, na forma de novos itens de *backlog*, ajustes em itens existentes e anotações associadas a cada tarefa. Esse registro, somado à documentação de requisitos e de arquitetura, aos manuais do usuário e técnico e ao termo de entrega, constitui a base de dados deste relato. A análise foi qualitativa e consistiu em confrontar o que foi planejado com o que foi efetivamente entregue, identificando decisões técnicas, dificuldades e pendências.

4 Resultados e Discussão

4.1 Requisitos e funcionalidades entregues

O MVP do EduControle foi estruturado em torno de dez requisitos funcionais, sintetizados no Quadro 1. Os perfis de acesso definidos foram Secretária, Portaria e Funcionário, cada um com acesso apenas às telas compatíveis com sua função.

Quadro 1 – Requisitos funcionais do MVP do EduControle

ID	Requisito	Situação na entrega
RF01	Autenticar usuário e aplicar perfil de acesso	Implementado
RF02	Cadastrar turma	Implementado
RF03	Cadastrar aluno vinculado a uma turma	Implementado
RF04	Registrar entrada do aluno com data e hora automáticas	Implementado
RF05	Registrar saída do aluno com data e hora automáticas	Implementado; regra para saída sem entrada prévia pendente
RF06	Consultar histórico de entrada e saída	Implementado
RF07	Gerar relatório com filtros por período, aluno e turma	Implementado
RF08	Exportar relatório em CSV e PDF	Implementado
RF09	Consultar ajuda e perguntas frequentes no sistema	Implementado
RF10	Importar registros do relógio de ponto físico	Implementado por importação manual de arquivo

Fonte: Elaborado pelos autores (2026).

Uma decisão de projeto relevante foi impedir a edição manual do horário nos registros de entrada e saída: data e hora são gravadas pelo próprio sistema no momento do evento. Essa escolha protege a confiabilidade do histórico, já que um dos problemas do processo manual era justamente o registro incorreto de horários. Entre os requisitos não funcionais, destacam-se o controle de acesso por RLS, a compatibilidade com as duas últimas versões principais dos navegadores Chrome, Edge e Firefox, a dependência de conexão com a internet, sem modo offline, e a usabilidade voltada a operadores sem formação técnica.

Cabe registrar uma lição de processo: a especificação formal dos requisitos foi consolidada a partir do comportamento já implementado e validado no piloto, e não integralmente antes da implementação. Isso permitiu avançar rapidamente nas primeiras *sprints*, mas deixou dois pontos em aberto na entrega: a regra de negócio para o registro de uma saída sem entrada prévia no mesmo dia (RF05) e uma meta formal de tempo de resposta para o registro de presença, que não chegou a ser medida. Em consonância com Pressman e Maxim (2016), a experiência mostrou que requisitos não documentados no momento oportuno tendem a reaparecer como ambiguidades no final do projeto.

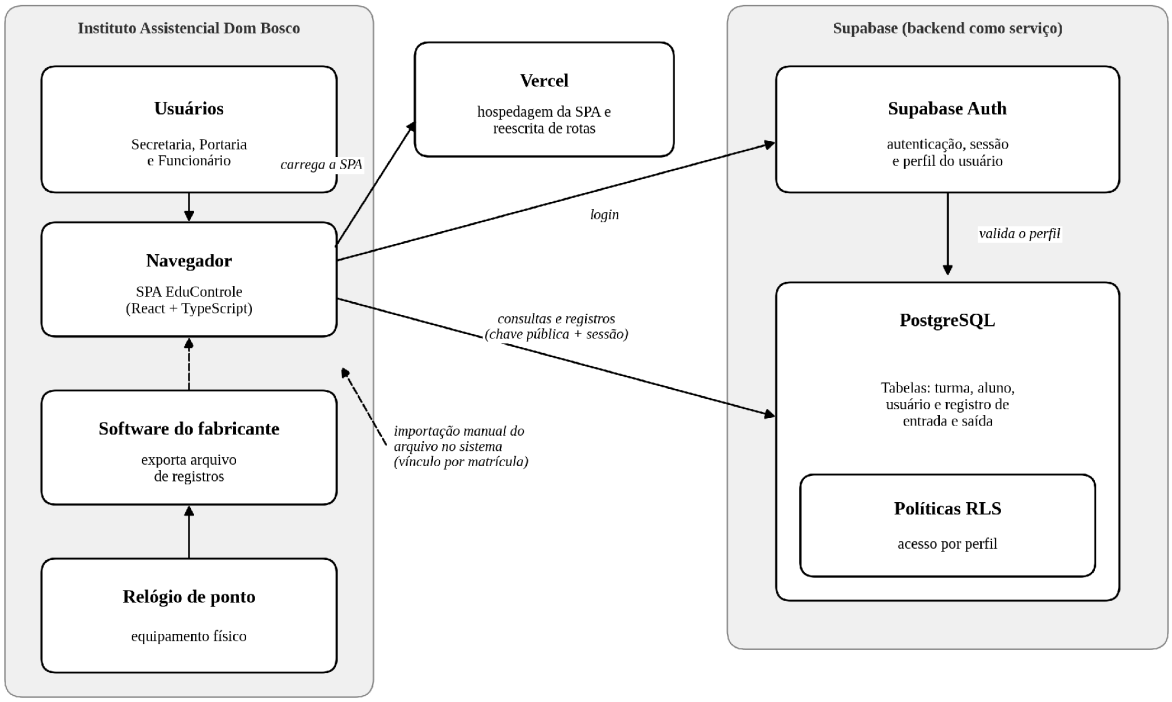
4.2 Arquitetura e decisões técnicas

O EduControle foi construído como uma SPA em React 18 e TypeScript, com Vite como ferramenta de build. A interface utiliza a biblioteca de componentes shadcn/ui com Tailwind CSS; o estado assíncrono e o cache das consultas são gerenciados pelo TanStack React Query; os

formulários são controlados pelo React Hook Form, com validação de esquema pela biblioteca Zod; e a navegação entre telas é feita pelo React Router v7. O *backend* é o Supabase, que fornece um banco PostgreSQL gerenciado e um módulo de autenticação, e o *frontend* é publicado na plataforma Vercel. A Figura 1 apresenta os componentes da solução e suas interações.

A principal decisão arquitetural foi dispensar uma API própria intermediária, fazendo o *frontend* comunicar-se diretamente com o Supabase. A alternativa, uma API desenvolvida, por exemplo, em Node.js, ofereceria maior controle sobre as regras de negócio e desacoplaria a interface do banco de dados, mas exigiria manter um servidor em execução, monitorar sua disponibilidade e implementar uma camada própria de autenticação e autorização. Considerando que a instituição não possui equipe de TI e que o prazo era limitado ao calendário acadêmico, o modelo BaaS mostrou-se mais adequado, alinhado ao que Sommerville (2018) aponta sobre a influência do contexto operacional nas escolhas de arquitetura.

Figura 1 - Arquitetura do EduControle e fluxo de importação do relógio de ponto



Fonte: Elaborado pelos autores (2026).

Essa escolha, contudo, desloca a responsabilidade pela segurança para as políticas de RLS. Como a chave usada pelo navegador é pública (SUPABASE, 2026), a autorização por perfil depende integralmente de as políticas estarem corretas e completas. Na prática, isso impõe uma

regra de manutenção: qualquer nova tabela ou coluna sensível deve ser criada acompanhada da política correspondente. O comportamento de negação por padrão do PostgreSQL (POSTGRESQL GLOBAL DEVELOPMENT GROUP, 2026) atenua o risco de exposição acidental em tabelas com RLS habilitada, mas não protege tabelas em que ela não tenha sido ativada.

Um problema técnico encontrado na publicação ilustra as particularidades das SPAs. Como o roteamento é feito no navegador, o acesso direto a uma rota interna, ou a atualização da página fora da rota inicial, fazia o servidor de hospedagem procurar um arquivo inexistente e retornar ao erro 404. A solução foi configurar uma regra de reescrita no arquivo de configuração da Vercel, direcionando todas as rotas ao arquivo inicial da aplicação para que o próprio roteador decida qual tela exibir. A regra foi documentada no manual técnico, pois precisa ser mantida em qualquer nova publicação ou migração de hospedagem.

4.3 Integração com o relógio de ponto

A integração com o relógio de ponto físico foi o principal desafio técnico do projeto. A proposta inicial previa uma integração automática e em tempo real entre o equipamento e o sistema em nuvem. Ao longo do desenvolvimento, porém, a equipe constatou que essa abordagem dependia de fatores que não eram de seu conhecimento prévio, relacionados ao equipamento e ao software do fabricante, cujo estudo e implementação não caberiam no prazo do projeto.

A solução entregue foi um processo de importação manual: os registros são exportados pelo software do fabricante, conferidos e importados no EduControle, que associa cada evento de entrada ou saída ao aluno correspondente por meio do número de matrícula. As matrículas dos 180 alunos foram atribuídas manualmente, em ordem alfabética, de 1 a 180, e essa numeração precisa coincidir nos dois sistemas. Após cada importação, recomenda-se validar uma amostra dos registros no histórico ou nos relatórios.

Essa abordagem resolveu o problema imediato e colocou em uso um equipamento que estava ocioso, mas apresenta uma fragilidade que precisa ser explicitada: a correspondência depende de uma numeração mantida manualmente em dois sistemas independentes. Um aluno cadastrado fora da ordem original, ou uma reorganização da lista no equipamento, pode associar registros ao aluno errado sem que o sistema indique qualquer erro. Trata-se de uma falha

silenciosa, um tipo de falha particularmente difícil de detectar, pois o dado incorreto aparece nos relatórios com a mesma aparência de um dado correto. Por isso, a equipe recomendou a criação de um validador de importação que aponte divergências de matrícula e, enquanto ele não existir, a designação formal de um responsável, na instituição, pela conferência a cada importação.

4.4 Implantação piloto e percepção dos usuários

Às 16 tarefas planejadas para a fase de implantação e treinamento foram concluídas, e o piloto de implantação transcorreu sem registro de falhas críticas. O sistema foi utilizado por funcionários da secretaria, da portaria e por um professor coordenador, e o hardware já existente na instituição atendeu aos requisitos, sem necessidade de aquisição de equipamentos, o que é relevante para uma entidade com recursos limitados.

O contato com os usuários gerou ajustes concretos no produto. Durante o treinamento, os próprios funcionários sinalizaram a necessidade de filtrar informações por turma, o que levou à inclusão da entidade Turma no modelo de dados. Durante os testes, foram apontados dois pontos de melhoria ainda não implementados na entrega: a busca de alunos por nome parcial não exibe a turma ou a matrícula ao lado do nome, o que pode causar confusão entre alunos com nomes parecidos; e a tela de autenticação, considerada funcional, mas pouco alinhada à identidade visual do instituto. Esses retornos ilustram a diretriz de interação dialógica (FORPROEX, 2012): parte relevante dos requisitos só se tornou visível quando os usuários operaram o sistema.

Também foi definido um plano de contingência simples e compatível com a realidade da instituição: em caso de indisponibilidade confirmada por mais de 15 minutos, retorna-se temporariamente ao registro em papel, procedimento já conhecido pelos funcionários, até o restabelecimento do serviço.

4.5 Limitações e pendências

Os resultados devem ser lidos à luz de algumas limitações. O piloto foi realizado em ambiente de homologação, com acompanhamento manual da equipe, e a entrega ocorreu antes da migração para operação definitiva. Ficaram registradas no termo de entrega, como pré-requisitos para essa migração, a substituição das credenciais de teste por credenciais individuais com política mínima de senha, a integração de uma ferramenta de captura de erros no *frontend* e a execução de ao menos um teste completo de restauração de backup. Considerando que o sistema

trata dados de adolescentes, a gestão de credenciais e o controle de acesso não são apenas boas práticas, mas medidas exigidas pela LGPD (BRASIL, 2018a, art. 46).

Além disso, o projeto não realizou mensuração quantitativa do impacto, como a comparação do tempo gasto ou da quantidade de erros de registro antes e depois da implantação. Os ganhos relatados, como maior organização e facilidade de consulta ao histórico, baseiam-se na percepção dos usuários durante o piloto e na substituição do processo manual, e não em indicadores medidos. Essa avaliação é recomendada para uma etapa futura, após um período de uso em operação definitiva.

4.6 Contribuições sociais e formativas

Do ponto de vista social, o sistema atende diretamente os 180 alunos do instituto, que passam a ter entrada e saída registradas de forma padronizada, e a equipe administrativa responsável pela gestão da presença. Indiretamente, alcança as famílias desses alunos, que passam a contar com um registro mais confiável da rotina dos filhos na instituição, e contribui para fortalecer a gestão de uma entidade que oferece formação gratuita à comunidade de Guarapuava há mais de quatro décadas.

Do ponto de vista formativo, o projeto exigiu da equipe decisões que raramente aparecem em atividades acadêmicas convencionais: escolher uma arquitetura em função de quem vai mantê-la, renegociar o escopo de uma integração diante de restrições técnicas reais, documentar o sistema para um público não técnico e registrar com transparência as pendências de uma entrega. Essas experiências correspondem à diretriz de impacto na formação do estudante prevista na política nacional de extensão (FORPROEX, 2012).

4.7 Orientações para replicação

A experiência pode subsidiar projetos semelhantes em outras instituições sociais sem equipe de TI. A primeira orientação é escolher a arquitetura em função de quem vai mantê-la: em contextos como o do Instituto Dom Bosco, uma plataforma gerenciada reduz a dependência de manutenção de servidores, mas exige que a equipe domine o modelo de segurança dessa plataforma. No caso de soluções com RLS, as políticas de acesso devem ser tratadas como parte do esquema do banco de dados, versionadas e revisadas junto com cada nova tabela, e não como uma configuração aplicada uma única vez.

A segunda orientação é evitar que a integridade dos dados dependa de identificadores mantidos manualmente em sistemas independentes. Quando essa dependência for inevitável, como ocorreu na integração com o relógio de ponto, recomenda-se prever desde o início um mecanismo de validação na importação e um responsável institucional pela conferência. A terceira orientação é reservar tempo, no cronograma, para treinamento, piloto com usuários reais e documentação voltada a um público não técnico, etapas nas quais surgiram ajustes que não haviam aparecido durante o desenvolvimento.

Por fim, a replicação do EduControle em outro contexto exigiria adaptar perfis de acesso, regras de registro e formas de integração aos processos da instituição atendida. A experiência deve ser entendida, portanto, como referência de processo e de decisões técnicas, e não como solução pronta para qualquer organização.

5 Considerações Finais

Este artigo relatou a experiência de desenvolvimento e implantação do EduControle, sistema web de controle de entrada, saída e presença de alunos desenvolvido como projeto de extensão para o Instituto Assistencial Dom Bosco. O objetivo de substituir o controle manual por uma solução informatizada, simples de operar e de baixo custo de manutenção foi atingido no escopo do MVP: todas as funcionalidades foram entregues, com a integração ao relógio de ponto adaptado para importação manual, e utilizadas pelos funcionários no piloto, sem falhas críticas.

A principal conclusão técnica é que o modelo de *backend* como serviço, com controle de acesso por RLS, mostrou-se adequado a uma instituição sem equipe de TI, por reduzir a infraestrutura a ser mantida. Essa vantagem, porém, tem como contrapartida a concentração da segurança nas políticas do banco de dados, o que exige disciplina na evolução do sistema. Da mesma forma, a integração com o relógio de ponto por importação manual viabilizou o uso de um equipamento antes ocioso, mas depende de um processo de conferência que precisa ter um responsável definido na instituição.

Como trabalhos futuros, recomendam-se: a definição da regra de negócio para saídas sem entrada prévia; a medição formal do tempo de resposta do registro de presença; a criação de um validador automático de matrículas na importação; a inclusão de turma ou matrícula na busca de alunos; a implementação de monitoramento de erros; a substituição das credenciais de teste; e a avaliação quantitativa do impacto após um período de operação definitiva. Por fim, a experiência

reforça que, em projetos de extensão, a qualidade da entrega depende tanto do software produzido quanto da clareza com que suas limitações são comunicadas à comunidade atendida.

Referências

BRASIL. Lei nº 13.709, de 14 de agosto de 2018. **Lei Geral de Proteção de Dados Pessoais (LGPD)**. Brasília, DF: Presidência da República, 2018a. Disponível em: https://www.planalto.gov.br/ccivil_03/_ato2015-2018/2018/lei/l13709.htm. Acesso em: 26 out. 2026.

BRASIL. Ministério da Educação. Conselho Nacional de Educação. Câmara de Educação Superior. Resolução nº 7, de 18 de dezembro de 2018. Estabelece as Diretrizes para a Extensão na Educação Superior Brasileira e regimenta o disposto na Meta 12.7 da Lei nº 13.005/2014, que aprova o Plano Nacional de Educação – PNE 2014-2024 e dá outras providências. **Diário Oficial da União**: seção 1, Brasília, DF, p. 49-50, 19 dez. 2018b.

FÓRUM DE PRÓ-REITORES DAS INSTITUIÇÕES PÚBLICAS DE EDUCAÇÃO SUPERIOR BRASILEIRAS (FORPROEX). **Política Nacional de Extensão Universitária**. Manaus: FORPROEX, 2012.

POSTGRESQL GLOBAL DEVELOPMENT GROUP. Row security policies. In: POSTGRESQL GLOBAL DEVELOPMENT GROUP. **PostgreSQL documentation**. [S. l.], 2026. Disponível em: https://www.postgresql.org/docs/current/ddl-rowsecurity.html. Acesso em: 26 out. 2026.

PRESSMAN, Roger S.; MAXIM, Bruce R. **Engenharia de software**: uma abordagem profissional. 8. ed. Porto Alegre: AMGH, 2016.

RIES, Eric. **A startup enxuta**: como os empreendedores atuais utilizam a inovação contínua para criar empresas extremamente bem-sucedidas. São Paulo: Leya, 2012.

SCHWABER, Ken; SUTHERLAND, Jeff. **The Scrum Guide**: the definitive guide to Scrum: the rules of the game. [S. l.]: Scrum.org, 2020. Disponível em: https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-US.pdf. Acesso em: 26 out. 2026.

SOMMERVILLE, Ian. **Engenharia de software**. 10. ed. São Paulo: Pearson Education do Brasil, 2018.

SUPABASE. **Row Level Security**. [S. l.], 2026. Disponível em: https://supabase.com/docs/guides/database/postgres/row-level-security. Acesso em: 26 out. 2026.