AUTOMAÇÃO DE REGISTRO ACADÊMICO PARA OTIMIZAÇÃO PEDAGÓGICA

Amanda Leite de Souza Machado, Matheus Oliveira Marcondes, Yure Campos Camilo, Wagner dos Santos Clementino de Jesus.

Fundação Vale Paraibana de Ensino/Colégio Técnico "Antônio Teixeira Fernandes", Avenida Paraibuna, 75, Centro - 12245-020 - São José dos Campos-SP, Brasil, amandaleitedsm@gmail.com, matheusmarcondesunivap@gmail.com, ccamposcamilo@gmail.com, wagner@univap.br.

Resumo

O controle de frequência escolar é essencial para o acompanhamento acadêmico e para o cumprimento das exigências legais de permanência dos estudantes. Entretanto, registros manuais e conferências por planilhas podem consumir tempo pedagógico e gerar inconsistências. Este trabalho apresenta o desenvolvimento de um sistema desktop para automação do registro de assiduidade, integrando C#, WPF, arquitetura MVVM e banco de dados MySQL. A metodologia envolveu levantamento do problema, prototipagem, estruturação da base relacional, importação de arquivos CSV e geração de relatórios em HTML. Como resultado, o sistema permite cadastrar alunos e turmas, importar presenças, identificar ausências automaticamente e emitir relatórios de alunos, aulas e turmas. Conclui-se que a ferramenta contribui para a otimização do trabalho docente, a integridade dos dados e o acompanhamento mais ágil da frequência acadêmica.

Palavras-chave: Assiduidade. Automação escolar. C#. MySQL. Relatórios acadêmicos.

Área do Conhecimento: Ciências Exatas e da Terra - Ciência da Computação.

Introdução

A organização do ensino brasileiro exige acompanhamento contínuo da trajetória escolar, incluindo registros confiáveis de frequência. A Lei de Diretrizes e Bases da Educação Nacional (LDB) estabelece carga horária mínima anual e reforça a importância do controle da permanência dos estudantes. Nesse contexto, a assiduidade deixa de ser apenas uma rotina administrativa e passa a integrar o processo de gestão pedagógica, acompanhamento acadêmico e tomada de decisão escolar (Malta, 2017).

Apesar disso, ainda são comuns práticas baseadas em chamadas orais, diários físicos ou comparação manual entre listas de presença e planilhas de alunos matriculados. Esse processo é lento, propenso a erros e reduz o tempo que o professor poderia dedicar às atividades de ensino. O problema torna-se ainda mais relevante em turmas numerosas ou quando o docente precisa acompanhar várias classes no mesmo período letivo (Santos; Correia; Koehler, 2023).

Diante desse cenário, o presente trabalho tem como objetivo desenvolver um sistema desktop para automatizar a conferência de frequência acadêmica, cruzando dados de presença com listas de alunos cadastrados e gerando relatórios gerenciais. A proposta busca aumentar a confiabilidade dos registros, reduzir retrabalho e oferecer ao docente uma ferramenta prática para acompanhamento de ausências e histórico por turma.

Metodologia

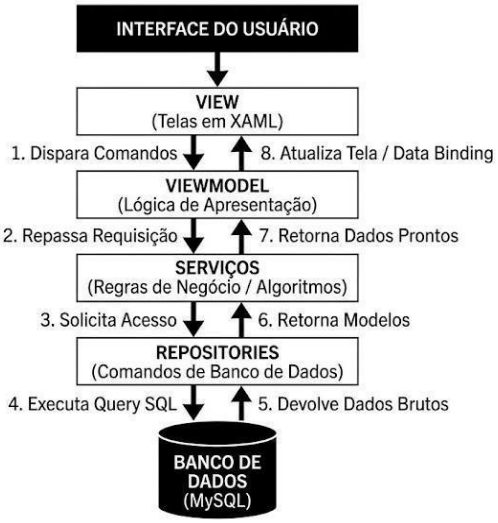
A presente pesquisa caracteriza-se como um estudo aplicado de desenvolvimento tecnológico, com foco na engenharia de software para otimização de processos acadêmicos, sendo desenvolvida a partir da identificação das necessidades relacionadas ao registro de alunos, turmas, aulas e assiduidade (Santos; Correia; Koehler, 2023), resultando na definição das funcionalidades de cadastro, importação, consulta e geração de relatórios para acompanhamento acadêmico.

O sistema desktop foi construído utilizando a linguagem C# sobre a plataforma *.NET Framework* 4.7.2, cuja escolha é fundamentada na sua facilidade nativa para a manipulação de arquivos de texto e streams de dados, funções essenciais para a leitura de planilhas e integração no sistema (Kunen, 2024). Para a construção da interface gráfica, empregou-se o *Windows Presentation Foundation* (WPF). Para o armazenamento e estruturação das informações, optou-se pelo banco de dados

relacional MySQL, o qual oferece segurança e eficiência no armazenamento das informações do sistema (Laudon; Laudon, 1999). O controle de versão do código foi realizado via Git, com hospedagem no GitHub. O fluxo de telas foi previamente validado utilizando etapas de identificação do problema, ideação, prototipação e refinamento das soluções com apoio do método *Design Sprint* (Knapp; Zeratsky; Kowitz, 2017), permitindo validar a usabilidade e os fluxos de uso antes da consolidação e codificação das interfaces finais.

Do ponto de vista da engenharia de software, a aplicação foi estruturada sob o padrão arquitetural Model-View-ViewModel (MVVM), garantindo o baixo acoplamento entre a interface do usuário desenvolvida em *Extensible Application Markup Language* (XAML), a lógica de apresentação (*ViewModels*) e o modelo de dados (Oliveira, 2012). Para otimizar a manutenção e a escalabilidade, o sistema foi subdividido em camadas adicionais, seguindo os princípios de código limpo com foco na clareza, legibilidade e no princípio da responsabilidade única de cada método e camada (Magalhães; Tiosso, 2019). A camada de Repositórios (como *AlunoRepository* e *TurmaRepository*) é responsável por isolar as consultas e comandos *Structured Query Language* (SQL). Já a camada de Serviços destina-se a centralizar e encapsular as regras de negócio e algoritmos de processamento (Manhoni, 2025). Essa divisão funcional encontra-se ilustrada na Figura 1.

Figura 1 - Fluxo lógico e arquitetura em camadas do sistema.



Fonte: Autores (2026).

O modelo relacional de dados foi normalizado para representar as principais entidades do domínio escolar: alunos, turmas, vínculo aluno-turma, aulas e ausências. O processamento do controle de frequência foi estruturado em um fluxo de três etapas principais: extração, validação e persistência. Inicialmente, o sistema utiliza rotinas de leitura de arquivos para extrair dados tabulares de planilhas no formato *Comma-Separated-Values* (CSV) (Rocha, 2021). Em seguida, a camada de serviços executa a validação e o cruzamento das informações: o algoritmo compara a lista de alunos presentes no arquivo importado com a lista completa de alunos vinculados àquela turma específica no banco de dados.

Por dedução lógica, os estudantes matriculados que não constam no arquivo CSV importado são classificados como faltosos. Os registros dessas ocorrências são então repassados aos repositórios e persistidos no MySQL, vinculados à data da aula e à turma correspondente, evitando redundâncias e preservando a integridade referencial.

Para a extração de resultados e acompanhamento pedagógico, desenvolveu-se um módulo de emissão de relatórios (Relatorio Service) que consome objetos de transferência de dados *Data Transfer Objects* (DTO). Este serviço estrutura os dados gerenciais dinamicamente em arquivos de marcação *HyperText Markup Language* (HTML). Essa abordagem foi escolhida por ser leve e interoperável, permitindo que os documentos (relatórios de alunos, turmas e ausentes por aula) sejam abertos diretamente no navegador padrão do sistema operacional, facilitando a visualização, a impressão e a

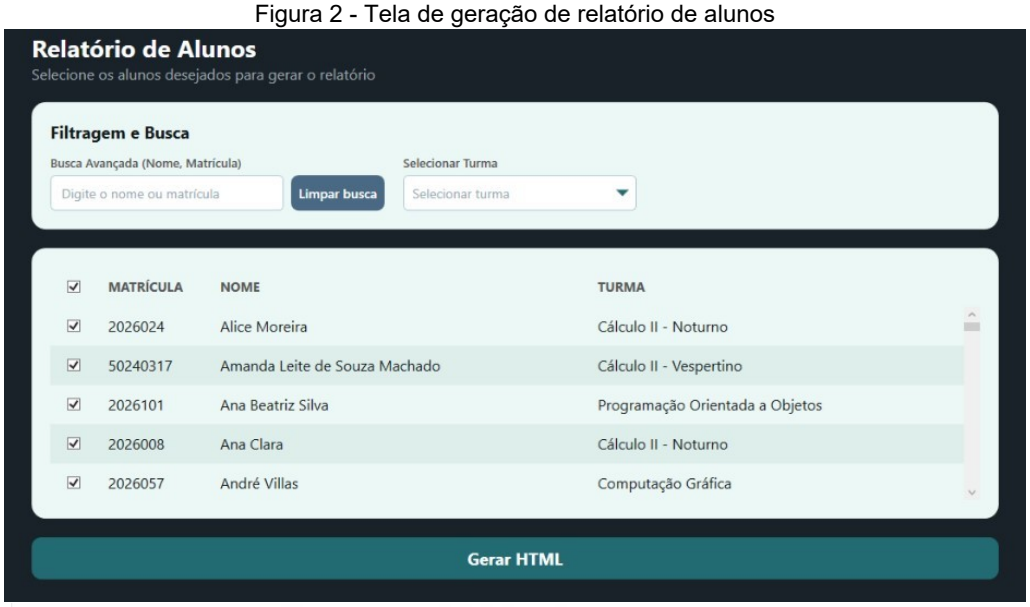
exportação nativa para formato *Portable Document Format* (PDF) sem a necessidade de instanciar bibliotecas pesadas na aplicação.

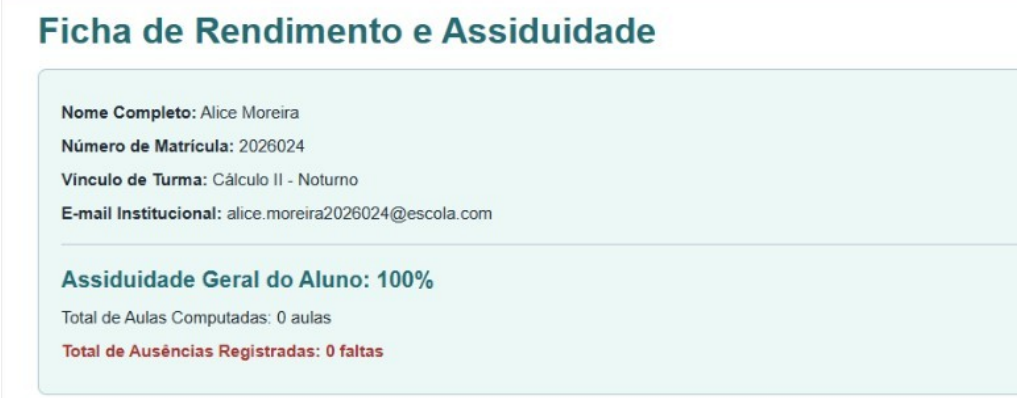
Resultados

O desenvolvimento resultou em um sistema funcional de gestão de assiduidade, capaz de centralizar cadastros, importar presenças e gerar relatórios. A Central de Relatórios – interface exibida ao clicar em 'Relatórios' no *sidemenu* – reúne três caminhos principais: relatórios de alunos, de aulas e de turmas.

Essa organização simplifica a navegação e permite que o usuário filtre informações por nome, matrícula, turma ou período, selecionando os registros que devem compor o documento final. Foram implementadas telas específicas para relatórios por aluno, aula e turma. Nos relatórios por aluno, o docente pode consultar a ficha de rendimento e assiduidade de um ou mais alunos selecionados. Nos relatórios por aula, é possível verificar ausentes daquela selecionada. Nos relatórios por turma, o sistema consolida o histórico de frequência, permitindo visão coletiva do comportamento da classe ao longo do período letivo. Tais telas mencionadas são representadas, respectivamente, pelas Figuras 2, 3 e 4.

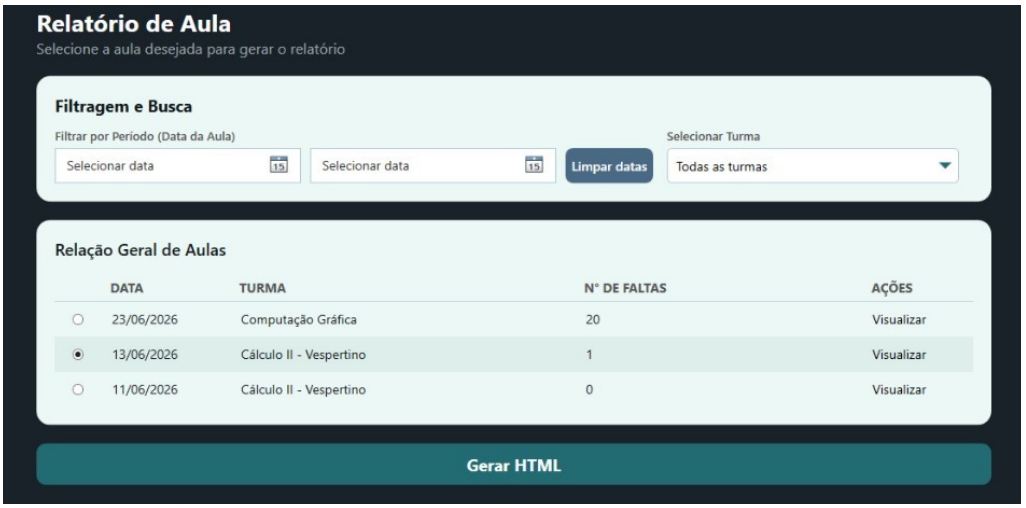
Figura 2 - Tela de geração de relatório de alunos





Fonte: Autores (2026).

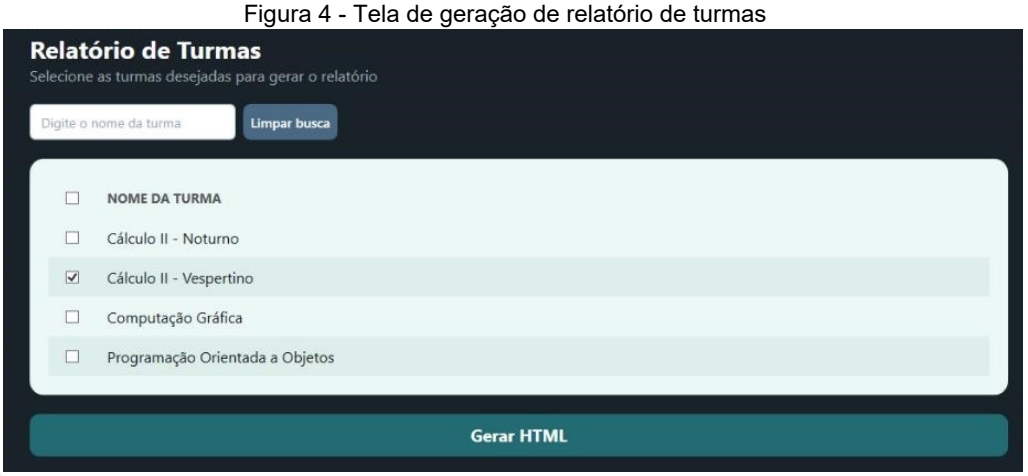
Figura 3 - Tela de geração de relatório de aula





Fonte: Autores (2026).

Figura 4 - Tela de geração de relatório de turmas





Fonte: Autores (2026).

Discussão

Os resultados indicam que a informatização do controle de frequência contribui para transformar dados brutos, antes dispersos, em informações estratégicas úteis para a gestão acadêmica. Essa transformação de registros operacionais em indicadores de apoio à decisão reflete os princípios fundamentais dos Sistemas de Informação Gerenciais (Velloso, 1999). A arquitetura em camadas também fortalece a manutenção do sistema, pois isola adequadamente consultas SQL, regras de negócio e a interface gráfica.

Ao debater os métodos de automação de chamadas, nota-se que a substituição de registros manuais pela importação automatizada de CSV favorece a agilidade docente (Rocha, 2021). Diferentemente de soluções recentes que exigem integração com hardware específico — como controle via tecnologia *Near Field Communication* (NFC) (Kunen, 2024) ou projetos de automatização por dispositivos físicos (Santos; Correia; Koehler, 2023) —, a abordagem deste trabalho foca estritamente no *software desktop*. Isso reduz barreiras de custo e infraestrutura, apresentando maior aderência operacional em escolas com recursos limitados. No que tange à exportação de dados, a geração de relatórios em HTML mostrou-se uma solução leve e compatível com diferentes ambientes computacionais. Embora existam alternativas robustas no ecossistema .NET para a criação direta de relatórios, como a biblioteca de geração de PDF *QuestPDF* (Mauri, 2024), a escolha pelo HTML eliminou a dependência de pacotes de terceiros de alta complexidade. Assim, o relatório pode ser aberto diretamente no navegador, impresso ou convertido nativamente, o que é especialmente útil em contextos escolares que demandam soluções simples e auditáveis (Silva, 2008).

Por fim, a centralização dos dados em base relacional favorece a continuidade do acompanhamento. A proposta demonstra que o sistema não substitui a análise pedagógica do professor, mas qualifica o seu acesso à informação. Ao entregar dados consolidados de forma rápida e permitir o resgate do histórico de turmas, a aplicação oferece suporte ágil para identificar estudantes com padrões recorrentes de ausência. Dessa forma, reduz-se a distância temporal entre o registro administrativo burocrático e a intervenção educacional preventiva.

Conclusão

Conclui-se que o sistema proposto foi estruturado com o objetivo de automatizar o registro de frequência acadêmica, apresentando um forte potencial para promover maior rapidez, integridade dos dados e apoio à tomada de decisão pedagógica. A solução desenvolvida viabiliza, em nível de software, funcionalidades como o cadastro de turmas e alunos, a importação de arquivos de presença, a dedução automática de ausências e a geração de relatórios de acompanhamento. Contudo, para confirmar sua eficiência definitiva na prática docente, faz-se necessária sua validação em um ambiente escolar real. Dessa forma, como continuidade, recomenda-se a execução de testes aplicados com diferentes formatos de planilhas, bem como o aprimoramento de indicadores visuais de risco de evasão e a integração de notificações para o acompanhamento preventivo de estudantes com baixa assiduidade.

Referências

KNAPP, Jake; ZERATSKY, John; KOWITZ, Braden. Sprint: o método usado no Google para testar e aplicar novas ideias em apenas cinco dias. Rio de Janeiro: Intrínseca, 2017.

KUNEN, G. M. Um modelo de automatização e controle de frequência em sala de aula integrado a um diário escolar utilizando tecnologia NFC. 2024.

LAUDON, K. C.; LAUDON, J. P. Sistemas de informação gerenciais. 4. ed. Rio de Janeiro: LTC, 1999.

MAGALHÃES, Phelipe Alex; TIOSSO, Fernando. CÓDIGO LIMPO: padrões e técnicas no desenvolvimento de software. Revista Interface Tecnológica, v. 16, n. 1, p. 197-207, 2019.
MALTA, D. A. S. A obrigatoriedade da educação infantil e a construção das práticas pedagógicas na pré-escola da Rede Municipal de Ensino de Franca/SP. 2017.

MANHONI, João Gabriel. Implementação de um CRUD integrado: PostgreSQL com funções de banco de dados e aplicação em C#/. NET.

MAURI, H. Criando relatórios com o QuestPDF no .NET. Henrique Mauri Blog, 2024. Disponível em: https://henriquemauri.net/. Acesso em: 06 abr. 2026.

OLIVEIRA, G. M. Desenvolvimento de software: separando interface das regras de negócio. 2012. Trabalho de Conclusão de Curso (Graduação em Ciência da Computação) - Centro Universitário FACVEST, Lages, 2012.

SANTOS, E. E.; CORREIA, Y. A.; KOEHLER, M. D. CardCall: projeto para automatização de chamadas escolares. Anais da Feira de Iniciação Científica e Extensão (FICE) Campus Camboriú, 2023.

SILVA, Maurício Samy. Criando sites com HTML: sites de alta qualidade com HTML e CSS. Novatec Editora, 2008.

ROCHA, Rafael. Integração semântica de dados tabulares em CSV: proposta de arcabouço comparativo de ferramentas. 2021.

VELLOSO, F. de C. Informática: conceitos básicos. Rio de Janeiro: Campus, 1999.