

DESENVOLVIMENTO DE SOFTWARE ORIENTADO A OBJETOS PARA GESTÃO**ESCOLAR:** Um Estudo de Caso com Java e Ionic no Centro de Ensino ViriatoCorrêa¹**OBJECT-ORIENTED SOFTWARE DEVELOPMENT FOR SCHOOL****MANAGEMENT:** A Case Study with Java and Ionic at the Centro de Ensino Viriato

Corrêa

Ismael Marques Da Costa Neto²Mariana Stephany Oliveira Araújo³Enzo Santiago Ferreira⁴Grazielle Lima Pereira⁵Maria Eduarda Aires Marques⁶Josiel Ribeiro de Sousa⁷Heitor Wagner Vieira Pereira⁸Leonardo Henrique Silva Lago⁹**RESUMO**

Sistemas de gestão acadêmica genéricos frequentemente falham em atender demandas específicas escolares, como a identificação visual de discentes. Este artigo apresenta o desenvolvimento de um sistema customizado para o Centro de Ensino Educa Mais Viriato Corrêa, visando otimizar conselhos de classe e avaliações. A metodologia consistiu em um estudo de caso de desenvolvimento de software, aplicando conceitos de Orientação a Objetos com a linguagem Java (Spring Boot) no backend e o framework Ionic no frontend. Os resultados incluem a

¹ Short Paper apresentado à disciplina Programação Orientada a Objetos do Curso de Engenharia de Software do Centro Universitário Unidade de Ensino Superior Dom Bosco - UNDB.

² Graduando do 4º Período do Curso de Engenharia de Software do Centro Universitário Unidade de Ensino Superior Dom Bosco - UNDB. E-mail: 002-027188@aluno.undb.edu.br.

³ Graduanda do 4º Período do Curso de Engenharia de Software do Centro Universitário Unidade de Ensino Superior Dom Bosco - UNDB. E-mail: 002-026966@aluno.undb.edu.br.

⁴ Graduando do 4º Período do Curso de Engenharia de Software do Centro Universitário Unidade de Ensino Superior Dom Bosco - UNDB. E-mail: 002-027268@aluno.undb.edu.br.

⁵ Graduanda do 4º Período do Curso de Engenharia de Software do Centro Universitário Unidade de Ensino Superior Dom Bosco - UNDB. E-mail: 002-027185@aluno.undb.edu.br.

⁶ Graduanda do 4º Período do Curso de Engenharia de Software do Centro Universitário Unidade de Ensino Superior Dom Bosco - UNDB. E-mail: 002-026743@aluno.undb.edu.br.

⁷ Graduando do 4º Período do Curso de Engenharia de Software do Centro Universitário Unidade de Ensino Superior Dom Bosco - UNDB. E-mail: 002-027107@aluno.undb.edu.br.

⁸ Graduando do 4º Período do Curso de Engenharia de Software do Centro Universitário Unidade de Ensino Superior Dom Bosco - UNDB. E-mail: 002-027224@aluno.undb.edu.br.

⁹ Graduando do 4º Período do Curso de Engenharia de Software do Centro Universitário Unidade de Ensino Superior Dom Bosco - UNDB. E-mail: leonardo.lago@unbd.edu.br.

arquitetura do sistema e a implementação funcional dos módulos "Carômetro" e "Simulados". Conclui-se que a abordagem tecnológica adotada proporcionou uma solução robusta e móvel, validando o desenvolvimento customizado para a resolução de gargalos pedagógicos.

Palavras-chave: Desenvolvimento de Software. Gestão Escolar. Spring Boot. Ionic. Aplicação Híbrida.

ABSTRACT

Generic academic management systems often fail to meet school-specific demands, such as the visual identification of students. This article presents the development of a customized system for the Educa Mais Viriato Corrêa Teaching Center, aiming to optimize class councils and evaluations. The methodology consisted of a software development case study, applying Object-Oriented concepts with the Java language (Spring Boot) on the backend and the Ionic framework on the frontend. The results include the system architecture and the functional implementation of the "Carômetro" and "Simulados" modules. It is concluded that the adopted technological approach provided a robust and mobile solution, validating customized development as an effective way to resolve pedagogical bottlenecks.

Keywords: Software Development. School Management. Spring Boot. Ionic. Hybrid Application.

1 INTRODUÇÃO

O contexto educacional contemporâneo demanda ferramentas que agilizem processos administrativos e pedagógicos, garantindo eficiência na gestão da informação. No Centro de Ensino Educa Mais Viriato Corrêa, identificou-se um gargalo significativo nesses processos: a dificuldade dos docentes em associar visualmente os alunos aos seus dados acadêmicos durante os conselhos de classe, devido à ausência de um "carômetro" digital, somada à sobrecarga manual na criação e correção de avaliações e simulados. Essa carência de ferramentas adequadas gera inconsistências e consome tempo precioso do corpo docente, evidenciando a necessidade de uma solução tecnológica que integre essas funcionalidades de forma acessível e móvel.

Para atender a demandas específicas como essa, a literatura aponta a importância de soluções de software flexíveis e bem arquitetadas. Manoppo e Istiono (2023) destacam que o desenvolvimento de sistemas customizados, utilizando arquiteturas modulares, permite superar as limitações de compatibilidade e rigidez

frequentemente encontradas em sistemas legados ou genéricos. No tocante à implementação, a escolha das tecnologias é crucial. Para o backend, frameworks como o Spring Boot são amplamente recomendados por simplificar a construção de aplicações prontas para produção, oferecendo robustez e reduzindo o tempo de configuração inicial (Jani, 2020). Além disso, o uso de Java com Spring Boot proporciona agilidade e segurança no desenvolvimento de APIs REST, facilitando a integração entre sistemas (Calça; Carlson Filho, 2022).

Já para a interface do usuário (frontend), a necessidade de acesso via dispositivos móveis justifica a adoção de uma abordagem de desenvolvimento híbrido. Segundo Silva e Sotto (2017), essa modalidade permite criar aplicativos para múltiplas plataformas (Android e iOS) com um único código-base, reduzindo custos e tempo de desenvolvimento. O framework Ionic, especificamente, destaca-se nesse cenário por permitir o uso de tecnologias web padrão (HTML, CSS e JavaScript) para construir interfaces com aparência e desempenho nativos, facilitando a manutenção e a escalabilidade do projeto (Chaudhary, 2018).

Diante desse cenário, o objetivo geral deste trabalho é apresentar o desenvolvimento do Sistema de Gestão Acadêmica EduNext, uma solução customizada para o Centro de Ensino Educa Mais Viriato Corrêa. Os objetivos específicos incluem: a) levantar os requisitos funcionais junto à coordenação da escola; b) modelar e implementar os módulos de Carômetro e Simulados utilizando uma arquitetura moderna; e c) aplicar na prática os conceitos de Programação Orientada a Objetos através das tecnologias Spring Boot e Ionic. O restante deste artigo está organizado nas seções de Metodologia, onde se detalha o processo de desenvolvimento; Resultados e Discussão, que apresenta o sistema implementado; e Considerações Finais.

2 METODOLOGIA

Este trabalho caracteriza-se como uma pesquisa aplicada, de natureza qualitativa e exploratória, estruturada como um estudo de caso de desenvolvimento de software. O objetivo foi a construção de uma solução tecnológica customizada para problemas reais de gestão acadêmica.

O estudo foi realizado no Centro de Ensino Educa Mais Viriato Corrêa. A seleção da amostra de participantes para o levantamento de requisitos foi intencional, composta pela coordenação pedagógica e pelo corpo docente da

instituição, que atuaram como stakeholders principais na definição das demandas e necessidades do sistema, especificamente a identificação visual de alunos e a gestão de avaliações.

Para a obtenção dos dados, a técnica utilizada foi a entrevista semiestruturada com o representante da instituição. A partir das informações coletadas, procedeu-se à etapa de modelagem e especificação do sistema. Os requisitos foram traduzidos em Histórias de Usuário para guiar o desenvolvimento ágil. Os instrumentos utilizados nesta fase incluíram a Linguagem de Modelagem Unificada (UML) para a criação de diagramas de classes, visando estruturar as entidades do banco de dados, e a ferramenta Swagger para a documentação e design da API, garantindo a padronização dos contratos de comunicação entre as camadas do sistema.

A determinação da estratégia de implementação baseou-se em critérios de eficiência e portabilidade. Para o desenvolvimento do backend, utilizou-se a linguagem Java com o framework Spring Boot. A escolha se justifica pela capacidade do framework de simplificar a configuração inicial e acelerar o desenvolvimento de APIs REST robustas e escaláveis, conforme apontam Jani (2020) e Calça e Carlson Filho (2022). Para a persistência dos dados, foi adotado o Sistema Gerenciador de Banco de Dados (SGBD) PostgreSQL, devido à sua confiabilidade e natureza open source.

No que tange ao frontend (aplicação cliente), foram utilizados os equipamentos de desenvolvimento padrão e o framework Ionic com React. Esta escolha atendeu ao requisito de acessibilidade móvel, permitindo o desenvolvimento de uma aplicação híbrida compatível com Android, iOS e Web a partir de uma única base de código, o que reduz custos e a curva de aprendizado da equipe (Silva; Sotto, 2017; Chaudhary, 2018).

O desenvolvimento seguiu uma abordagem ágil e iterativa. Os dados coletados (requisitos) foram classificados em módulos prioritários: "Carômetro" (prioridade alta) e "Simulados". O tratamento dos dados consistiu na implementação da lógica de negócios e na validação por meio de testes funcionais. Os resultados desta pesquisa, apresentados na seção seguinte, consistem na descrição da arquitetura do sistema desenvolvido, ilustrada por diagramas, documentação da API e telas da aplicação funcional.

3 RESULTADOS E DISCUSSÃO

O principal resultado deste trabalho é o desenvolvimento do sistema EduNext, uma solução de gestão acadêmica customizada que integra uma API robusta a uma interface móvel híbrida. A seguir, apresentam-se os artefatos de software desenvolvidos e a discussão sobre sua pertinência técnica e pedagógica.

3.1 ENGENHARIA DE REQUISITOS E ARQUITETURA DO SISTEMA

O desenvolvimento iniciou-se pela definição das Histórias de Usuário, derivadas do levantamento de requisitos junto ao Centro de Ensino, o que guiou a modelagem das funcionalidades prioritárias. A Figura 1 apresenta o Diagrama de Classes do sistema, evidenciando as relações fundamentais entre as entidades Professor, Turma, Aluno e Simulado, essenciais para o funcionamento dos módulos.

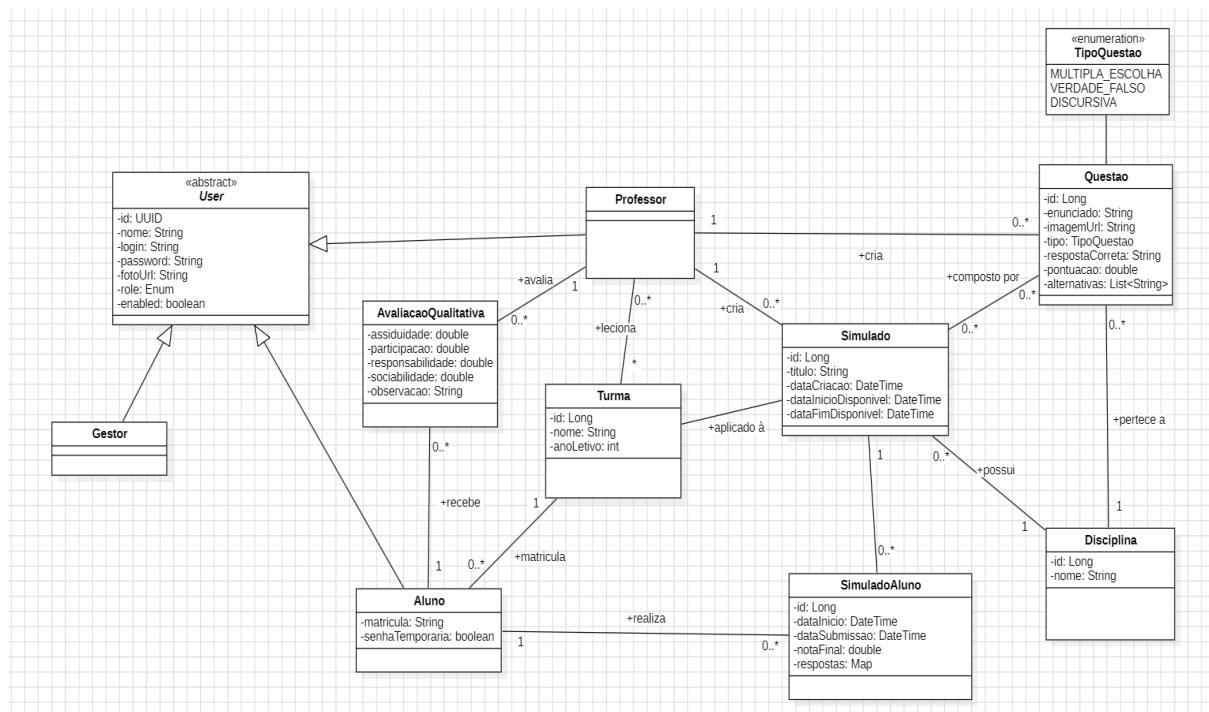


Figura 1: Diagrama de Classe das entidades principais. Fonte: Os autores (2025).

A implementação desta estrutura no backend utilizou a linguagem Java com o framework Spring Boot. A escolha por esta tecnologia permitiu a criação de uma API REST segura e escalável. A comunicação entre o servidor e os clientes foi padronizada e documentada através da ferramenta Swagger, conforme demonstrado na Figura 2.

The screenshot displays a Swagger UI interface for a REST API. It is organized into three main sections: **aluno-controller**, **file-upload-controller**, and **simulado-controller**. Each section lists various HTTP methods (GET, POST, PUT, DELETE) and their corresponding URLs. The URLs are color-coded: blue for GET, orange for PUT, red for DELETE, green for POST, and light blue for other methods like /alunos/{id}. Each entry includes a lock icon and a dropdown arrow.

- aluno-controller:**
 - GET /alunos/{id}
 - PUT /alunos/{id}
 - DELETE /alunos/{id}
 - PUT /alunos/{id}/resetar-senha
 - PUT /alunos/alterar-senha
 - GET /alunos
 - POST /alunos
 - GET /alunos/{alunoId}/avaliacoes
- file-upload-controller:**
 - POST /upload/photo
- simulado-controller:**
 - POST /simulados/{simuladoId}/submeter
 - POST /simulados/gerar-aleatorio
 - POST /simulados/criar
 - GET /simulados/{simuladoId}/resumo
 - GET /simulados/{simuladoId}/resultados
 - GET /simulados/{simuladoId}/fazer
 - GET /simulados/{simuladoId}/alunos
 - GET /simulados/professor/meus-simulados
 - GET /simulados/disponiveis

Figura 2: Exemplo da documentação da API REST via Swagger. Fonte: Os autores (2025).

Para garantir a confiabilidade da aplicação, foram implementados testes unitários e de integração. Esses testes validaram as regras de negócio críticas, como o cálculo de notas e a vinculação de alunos às turmas, assegurando que alterações no código não introduzirem regressões. Estes resultados, somados à documentação via Swagger, corroboram os estudos de Jani (2020) e Calça e Carlson Filho (2022), que destacam o Spring Boot como uma ferramenta que acelera o desenvolvimento de aplicações robustas e testáveis.

3.2 INTERFACE MÓVEL E O MÓDULO CARÔMETRO

Para atender à demanda de mobilidade dos docentes, o frontend foi desenvolvido utilizando o framework Ionic. As Figuras 3 e 4 ilustram a interface do módulo "Carômetro" e do módulo "Simulados" respectivamente, onde o professor pode visualizar a lista de alunos de uma turma, com suas respectivas fotos e dados

acadêmicos, diretamente pelo smartphone e criar simulados por turma, disciplina e personalizar disponibilidades e questões, respectivamente.

The figure consists of two side-by-side screenshots of a mobile application. The left screenshot shows a student profile for 'Ismael Neto'. At the top, there's a back arrow labeled 'VOLTAR' and the name 'Ismael Neto'. Below the name is a portrait photo of a young man with dark hair and a beard, wearing a light blue shirt. Below the photo, the name 'Ismael Neto' is displayed again, followed by 'Turma: 1º Ano A - Matutino' and 'Matrícula: 001'. Below this, there's a section titled 'Desempenho Geral (Média)' showing four categories with smiley face icons: 'Assiduidade' (smiling face with heart eyes), 'Participação' (smiling face with a small heart), 'Responsabilidade' (smiling face with a wide open mouth), and 'Sociabilidade' (frowny face with a sad expression). The right screenshot shows the 'Criar Simulado' (Create Simulation) screen. At the top, it says 'Novo Simulado' and 'Configure e crie simulados para suas turmas'. Below this is a section titled 'Informações Básicas' with fields for 'Título do Simulado' (Title) and 'Turma' (Class). Further down are fields for 'Disciplina' (Subject) and 'Período de Disponibilidade' (Availability Period) with start and end date fields. At the bottom, there's a section for 'Questões' (Questions) with a button to 'SELECIONAR QUESTÕES' (Select Questions) and a green button to '+ CRIAR NOVA QUESTÃO' (Create New Question). A note at the bottom says 'Selecione uma disciplina primeiro para ver as' (Select a subject first to see).

Figuras 3 e 4: Exemplo do uso do aplicativo para o carômetro e simulados. Fonte: Os autores (2025).

A adoção do desenvolvimento híbrido mostrou-se eficaz para garantir a compatibilidade com dispositivos Android e iOS a partir de um único código-base. Isso valida as observações de Silva e Sotto (2017) e Chaudhary (2018) sobre a redução de custos e a curva de aprendizado otimizada proporcionada pelo Ionic.

3.3 AUTOMAÇÃO DE AVALIAÇÕES E CUSTOMIZAÇÃO

Diferentemente de soluções genéricas, o EduNext implementou um módulo de Simulados específico para a realidade da escola. Esta funcionalidade permite ao professor criar bancos de questões e gerar provas automatizadas, onde o sistema realiza a correção instantânea de questões objetivas e auxilia na atribuição de notas discursivas.

A abordagem de desenvolver um sistema próprio, capaz de integrar o Carômetro visual com a gestão complexa de Simulados, alinha-se ao que Manoppo e Istiono (2023) defendem em seu estudo de caso: a reconstrução e customização de sistemas permitem superar limitações de plataformas rígidas, oferecendo a flexibilidade necessária para atender aos processos pedagógicos únicos da instituição.

4 CONSIDERAÇÕES FINAIS

O presente estudo explorou e aplicou conceitos fundamentais de desenvolvimento de software, arquitetura orientada a objetos e abordagens de programação ágil no contexto da gestão educacional. Especificamente, destacaram-se a relevância da customização de sistemas frente a soluções genéricas, a adoção de uma arquitetura modular baseada em API REST e a eficácia do desenvolvimento híbrido móvel com Ionic. A fundamentação teórica robusta acerca do uso de Spring Boot com Java no backend e do Ionic/React no frontend validou as escolhas tecnológicas como pilares para a construção de uma solução escalável, segura e portátil.

Os resultados e a discussão confirmaram a viabilidade técnica da solução EduNext (nome fictício), materializada em uma API robusta documentada via Swagger e uma interface móvel funcional. O módulo Carômetro demonstrou ser uma resposta direta e eficiente ao gargalo de identificação nos conselhos de classe, enquanto o módulo de Simulados endereçou a automação das avaliações. Deste modo, o trabalho atingiu seu objetivo geral de apresentar o desenvolvimento do sistema, cumprindo os objetivos específicos ao levantar os requisitos, modelar a arquitetura e aplicar os conceitos de Programação Orientada a Objetos com sucesso.

Para futuras pesquisas, sinaliza-se a possibilidade de explorar a integração de ferramentas de Inteligência Artificial (IA) no módulo de Simulados para análise preditiva de desempenho ou a inclusão de um módulo de feedback em tempo real para os alunos. Outra lacuna a ser explorada é a avaliação da usabilidade e da satisfação do usuário (corpo docente e coordenação) a longo prazo, após a implantação completa, visando aprimorar a experiência de interação com o sistema e mensurar o impacto pedagógico e administrativo da solução customizada.

REFERÊNCIAS

CALÇA, João Victor Justo; CARLSON FILHO, Carlos Magnus. Análise Comparativa entre os Frameworks Django e Spring Boot. Informática & Negócios, [S.I.], 2022.

CHAUDHARY, Priyanka. Ionic Framework. International Research Journal of Engineering and Technology (IRJET), [S.I.], v. 5, n. 5, p. 3181-3185, maio 2018.

JANI, Yash. Spring Boot for Microservices: Patterns, Challenges, and Best Practices. European Journal of Advances in Engineering and Technology, [S.I.], v. 7, n. 7, p. 73-78, 2020.

MANOPPO, Arnoldus Yitzhak Petra; ISTIONO, Wirawan. Cloud-Based ERP System Backend Design, study case: PT Cranium Royal Aditama. Ultimatics: Jurnal Teknik Informatika, Tangerang, v. 15, n. 2, p. 129-136, dez. 2023.

SILVA, Ewerthon Patrício Araujo da; SOTTO, Eder Carlos Salazar. A utilização do Ionic Framework no desenvolvimento de aplicações híbridas em arquitetura orientada a serviço. Interface Tecnológica, Taquaritinga, v. 15, n. 1, p. 97-108, 2018.