

Universidade Tiradentes
CIÊNCIA DA COMPUTAÇÃO

AGNALDO FILIPE
ANDRÉ FELIPE
GABRIEL ALMEIDA
MILTON FELIPE
THIAGO VINICIUS

PROJETO DE PROGRAMAÇÃO - GRUPO 02
CADASTRO E GERENCIAMENTO DE CURSOS

**AGNALDO FILIPE
ANDRÉ FELIPE
GABRIEL ALMEIDA
MILTON FELIPE
THIAGO VINICIUS**

**PROJETO DE PROGRAMAÇÃO - GRUPO 02
CADASTRO E GERENCIAMENTO DE CURSOS**

Atividade sobre um projeto apresentado como requisito parcial da avaliação da disciplina Projeto de Programação, ministrada pela Prof. Layse Santos Souza, no 2º semestre de 2025.

Sumário

1	INTRODUÇÃO	4
2	JUSTIFICATIVA	5
3	OBJETIVOS	6
3.1	OBJETIVO GERAL	6
3.2	OBJETIVOS ESPECÍFICOS	6
4	METODOLOGIA	7
5	RESULTADOS E DISCUSSÕES	8
6	CONSIDERAÇÕES FINAIS	9
7	REFERÊNCIAS	10
8	ANEXOS	11

1 INTRODUÇÃO

O avanço das tecnologias da informação tem impulsionado a modernização de sistemas acadêmicos e administrativos, exigindo soluções cada vez mais eficientes, seguras e automatizadas. Nesse contexto, o gerenciamento de cursos e turmas representa uma das atividades mais recorrentes e críticas em instituições de ensino, pois envolve o controle de informações sobre alunos, instrutores, categorias, matrículas, avaliações e disponibilidade de vagas. A crescente digitalização desses processos busca não apenas reduzir o tempo de execução de tarefas administrativas, mas também aumentar a transparência, a confiabilidade dos dados e a eficiência das operações educacionais.

Com o intuito de aplicar os conhecimentos adquiridos na disciplina de Projeto de Programação, este trabalho apresenta o desenvolvimento de um sistema de **Cadastro e Gerenciamento de Cursos**, elaborado em linguagem Java e estruturado a partir de boas práticas de programação orientada a objetos (POO). O sistema busca oferecer uma plataforma prática e acessível para o gerenciamento completo do ciclo de vida de um curso, desde o seu cadastro até a emissão de relatórios relacionados a instrutores, categorias e alunos matriculados.

O projeto foi desenvolvido com o uso de ferramentas amplamente aplicadas no mercado de tecnologia, como **Spring Framework**, para o gerenciamento de dependências e injeção de serviços, e **MySQL**, como banco de dados relacional responsável pela persistência das informações. A integração entre essas tecnologias permitiu ao grupo compreender de forma prática conceitos fundamentais da engenharia de software, como persistência de dados, encapsulamento, modularização e reutilização de código.

Além da aplicação técnica, o projeto proporcionou um ambiente de aprendizado colaborativo, permitindo aos integrantes do grupo exercitar habilidades de planejamento, modelagem, codificação e integração de diferentes tecnologias. O uso de metodologias ágeis, especialmente o **Scrum**, garantiu uma divisão eficiente das tarefas e acompanhamento contínuo do progresso semanal, simulando um ambiente profissional de desenvolvimento.

Dessa forma, o desenvolvimento deste sistema não apenas cumpre uma função acadêmica, mas também aproxima os alunos da realidade do mercado, promovendo a consolidação de competências essenciais da engenharia de software moderna, como o trabalho em equipe, a comunicação técnica e a capacidade de adaptação a desafios complexos.

2 JUSTIFICATIVA

A necessidade de sistemas informatizados para o gerenciamento de cursos é cada vez mais evidente nas instituições de ensino e plataformas educacionais. Processos manuais de controle e cadastro de cursos frequentemente resultam em inconsistências, retrabalhos e perda de informações, o que compromete a qualidade dos serviços prestados e a experiência dos usuários. Além disso, com o crescimento das ofertas de ensino a distância e híbrido, tornou-se indispensável contar com ferramentas automatizadas que garantam agilidade e precisão na administração de dados acadêmicos.

Nesse cenário, o desenvolvimento de um sistema de **Cadastro e Gerenciamento de Cursos** surge como uma proposta essencial para a modernização de processos internos, reduzindo o esforço humano e minimizando erros. O sistema busca atender tanto às necessidades administrativas quanto às pedagógicas, permitindo o controle integrado de cursos, categorias e instrutores de forma dinâmica, escalável e de fácil manutenção.

Do ponto de vista acadêmico, o projeto representa uma aplicação prática dos conceitos estudados na disciplina de Projeto de Programação e em outras áreas correlatas, como Banco de Dados, Engenharia de Software e Programação Orientada a Objetos. Ao criar uma solução completa e funcional, o grupo teve a oportunidade de desenvolver competências técnicas e interpessoais, essenciais para o exercício da profissão de cientista da computação.

Além de seu valor funcional, o projeto reforça a importância da aplicação de metodologias ágeis, como o **Scrum**, e de boas práticas de engenharia, como o uso da arquitetura MVC (Model-View-Controller) e a modelagem UML. Tais práticas promovem organização, produtividade e clareza na divisão das responsabilidades, fatores cruciais para o sucesso de projetos colaborativos.

Portanto, o desenvolvimento deste sistema justifica-se não apenas pela sua utilidade prática em ambientes de ensino, mas também pelo seu papel formativo no processo de aprendizagem dos alunos envolvidos. A experiência proporcionou aos integrantes do grupo uma visão mais realista do ciclo de vida de um software, abrangendo desde a análise de requisitos até a entrega final, fortalecendo sua capacidade de planejar, implementar e documentar soluções tecnológicas completas.

3 OBJETIVOS

3.1 OBJETIVO GERAL

Desenvolver um sistema completo de **Cadastro e Gerenciamento de Cursos**, capaz de realizar o controle de cursos, instrutores, categorias e matrículas, integrando as principais operações de um ambiente educacional digital.

3.2 OBJETIVOS ESPECÍFICOS

1. Implementar as operações de cadastro, edição e exclusão de cursos;
2. Estabelecer pré-requisitos e limites de vagas por curso;
3. Validar automaticamente o instrutor responsável, garantindo que esteja ativo e cadastrado;
4. Permitir associação de múltiplos instrutores e categorias a um mesmo curso;
5. Criar repositórios e tabelas de persistência (*curso* e *categoria_curso*);
6. Implementar busca avançada por categoria, carga horária e valor;
7. Gerar relatórios de cursos ativos e inativos, com contagem de alunos matriculados;
8. Garantir integridade referencial, impedindo exclusões de cursos com alunos inscritos.

4 METODOLOGIA

O desenvolvimento do sistema foi conduzido segundo a metodologia ágil **Scrum**, adaptada ao ambiente acadêmico. As atividades foram organizadas em ciclos semanais de trabalho, denominados *sprints*, nos quais eram definidas metas, implementadas funcionalidades e revisadas as entregas. Cada encontro possibilitou o acompanhamento da evolução do projeto, a correção de falhas e a integração de novas funcionalidades.

O grupo foi dividido em funções específicas: alguns integrantes ficaram responsáveis pela modelagem UML e estrutura de classes; outros atuaram na implementação da lógica de negócio e persistência de dados, enquanto outros se dedicaram à criação da interface e aos testes de funcionalidade. Essa divisão promoveu a especialização das tarefas e favoreceu o aprendizado colaborativo.

As principais ferramentas utilizadas foram:

- **Linguagem de Programação:** Java, pela sua robustez e amplo uso em sistemas corporativos;
- **Frameworks:** Spring, para controle de dependências e injeção de serviços, e Next.js, para interface moderna e responsiva;
- **Banco de Dados:** MySQL, responsável pela persistência e integridade das informações;
- **Ambiente de Desenvolvimento:** IntelliJ IDEA, pela integração nativa com Maven e suporte a frameworks Java;
- **Controle de Versão:** GitHub, utilizado para versionamento e colaboração do código-fonte.

Durante o desenvolvimento, foram aplicados conceitos de **Programação Orientada a Objetos**, como herança, polimorfismo e encapsulamento, além do uso de relacionamentos entre entidades e repositórios para acesso aos dados. O projeto seguiu a arquitetura em camadas (**MVC**), garantindo a separação das responsabilidades e facilitando futuras manutenções.

5 RESULTADOS E DISCUSSÕES

O sistema de Cadastro e Gerenciamento de Cursos foi desenvolvido com sucesso, atendendo a todos os requisitos funcionais e não funcionais estabelecidos. As funcionalidades principais, como o cadastro, edição e exclusão de cursos, foram implementadas de forma eficiente, garantindo a integridade dos dados e a interação fluida entre as camadas do sistema.

A integração entre as entidades *Curso*, *Instrutor*, *Categoria* e *Usuário* permitiu a criação de um ambiente robusto e modular. Além disso, foram implementadas validações automáticas para evitar inconsistências, como a tentativa de exclusão de cursos com alunos já matriculados.

Os testes realizados demonstraram estabilidade no funcionamento das operações e consistência no tratamento das regras de negócio. A interface do sistema se mostrou intuitiva e responsiva, permitindo que os usuários realizassem as tarefas com facilidade.

Entre os resultados complementares, destacam-se os recursos adicionais:

- Filtros de busca por categoria, carga horária e valor do curso;
- Verificação automática de vagas disponíveis antes da matrícula;
- Relatórios completos sobre cursos ativos e inativos, com contagem de alunos.

A adoção da arquitetura em camadas e do padrão MVC resultou em um código bem estruturado e de fácil manutenção. O uso de GitHub como ferramenta de versionamento também proporcionou rastreabilidade e controle das alterações realizadas por cada membro do grupo. De forma geral, o sistema alcançou um nível satisfatório de completude e desempenho, refletindo o comprometimento e a colaboração de todos os integrantes.

6 CONSIDERAÇÕES FINAIS

O desenvolvimento do sistema de **Cadastro e Gerenciamento de Cursos** representou uma importante oportunidade de integração entre teoria e prática, permitindo a aplicação de diversos conceitos aprendidos ao longo da disciplina de Projeto de Programação. O grupo conseguiu, de forma colaborativa, projetar e implementar um sistema funcional, modular e bem estruturado, seguindo os princípios da Programação Orientada a Objetos e boas práticas de engenharia de software.

A adoção da metodologia ágil **Scrum** contribuiu significativamente para a organização das tarefas, possibilitando o acompanhamento contínuo das entregas semanais e a adaptação do cronograma conforme as necessidades do projeto. Além disso, o uso de ferramentas como GitHub e MySQL consolidou o aprendizado sobre controle de versão e persistência de dados.

O sistema final apresentou desempenho satisfatório, com todas as funcionalidades implementadas e testadas, incluindo recursos adicionais como filtros de busca, relatórios e validações automáticas. O trabalho em equipe foi essencial para o alcance dos objetivos e para o amadurecimento técnico e profissional dos integrantes.

1. O trabalho em equipe foi essencial para a integração das diferentes etapas do projeto;
2. O uso de metodologias ágeis facilitou o planejamento e o cumprimento das metas semanais;
3. A implementação prática consolidou os conhecimentos sobre POO, banco de dados e arquitetura MVC;
4. A experiência reforçou a importância da documentação e da colaboração no desenvolvimento de sistemas reais.

7 REFERÊNCIAS

- PRESSMAN, R. *Engenharia de Software*. 8. ed. São Paulo: McGraw-Hill, 2016.
- SOMMERVILLE, I. *Engenharia de Software*. 10. ed. São Paulo: Pearson, 2018.
- ORACLE. *Documentação Oficial do Java*. Disponível em: <https://docs.oracle.com/en/java/>.
- MYSQL. *Documentação Oficial do MySQL*. Disponível em: <https://dev.mysql.com/doc/>.
- OMG. *UML Specification*. Disponível em: <https://www.omg.org/spec/UML/>.
- SPRING. *Spring Boot Documentation*. Disponível em: <https://spring.io/projects/spring-boot>.
- SPRING. *Spring Web – REST Services Guide*. Disponível em: <https://spring.io/guides/gs/rest-service/>.
- SPRING. *Spring Data JPA Documentation*. Disponível em: <https://spring.io/projects/spring-data-jpa>.
- SPRING. *Spring Security Documentation*. Disponível em: <https://spring.io/projects/spring-security>.
- ECLIPSE FOUNDATION. *Jakarta Bean Validation Specification*. Disponível em: <https://jakarta.ee/specifications/bean-validation/>.
- OKTA. *JWT – Java JWT Library*. Disponível em: <https://github.com/jwt/jjwt>.
- ORACLE. *MySQL Connector/J Documentation*. Disponível em: <https://dev.mysql.com/doc/connector-j/en/>.
- SPRINGDOC. *Springdoc OpenAPI*. Disponível em: <https://springdoc.org/>.
- APACHE. *Apache Maven Project*. Disponível em: <https://maven.apache.org/>.
- VERCEL. *Next.js Documentation*. Disponível em: <https://nextjs.org/docs>.
- AXIOS. *Axios HTTP Client*. Disponível em: <https://axios-http.com/>.
- MDN WEB DOCS. *Fetch API Documentation*. Disponível em: https://developer.mozilla.org/en-US/docs/Web/API/Fetch_API.
- OPENJS FOUNDATION. *Node.js Documentation*. Disponível em: <https://nodejs.org/en/docs>.
- NPM INC. *npm Documentation*. Disponível em: <https://docs.npmjs.com/>.

8 ANEXOS

1 Diagrama de Classes

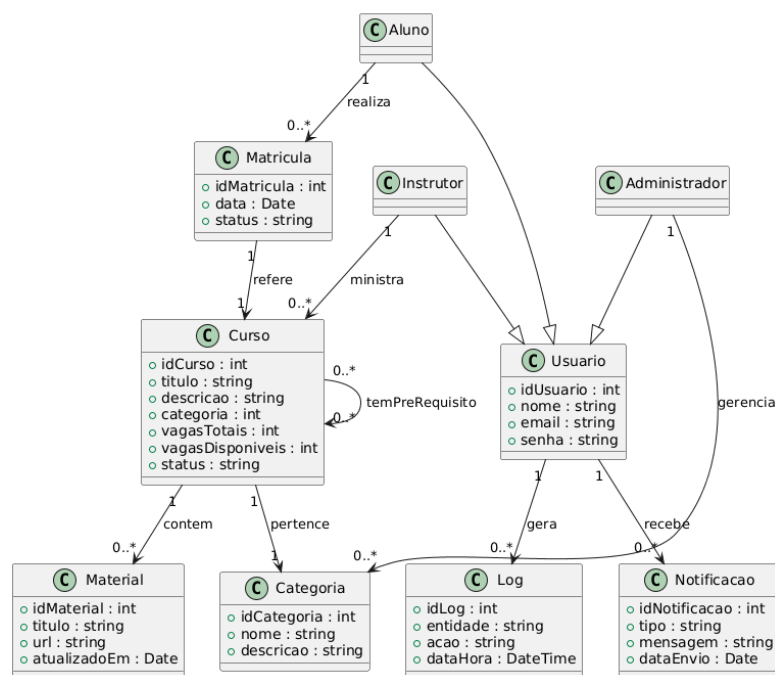


Figura 1 – Sistema de Cursos

2 Diagrama de Caso de Uso

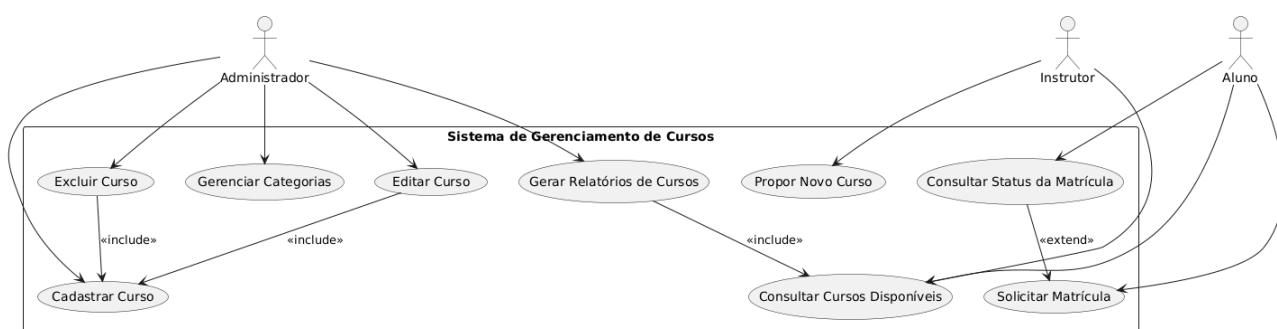


Figura 2 – Sistema de Gerenciamento de Cursos