

Instituto Federal do Norte de Minas Gerais - Campus Montes
Claros

Ciência da Computação
Programação Orientada a Objetos

Minimundo- Sistema de Gestão Acadêmica

Dijalmir Barbosa de Oliveira Junior
João Pedro Santos Silqueira
Maria Fernanda Andrade Rodrigues
Roberto Ramos Ferreira

Professor: Luis Antonio Guisso Lopes

Montes Claros
Julho de 2025

Mini-Mundo: Sistema de Gestão Acadêmica

Visão Geral

O sistema foi projetado para gerenciar de forma robusta e integrada as principais atividades de uma instituição de ensino. Ele controla a estrutura acadêmica (disciplinas e turmas), gerencia os atores (professores e discentes) e registra o histórico completo do desempenho dos alunos, incluindo matrículas, avaliações e frequência. O modelo é centrado na entidade **Turma**, que representa a oferta de uma disciplina em um período específico, garantindo uma representação fiel e escalável do ambiente acadêmico.

Entidades Principais, Atributos e Comportamentos

- **Entidade**
 - **Representa:** Uma classe base abstrata que fornece uma identidade única (id) para todas as outras entidades principais do sistema. Garante a consistência da chave primária em todo o modelo.
 - **Atributos:**
 - id: long
- **Professor**
 - **Representa:** O profissional docente que ministra as turmas.
 - **Atributos:** nome, cpf, nascimento (LocalDate).
 - **Comportamentos (Métodos):**
 - listarTurmasAtuais() : List<Turma>: Retorna a lista de todas as turmas que o professor está lecionando no período corrente.
- **Discente**
 - **Representa:** O estudante matriculado na instituição.
 - **Atributos:** nome, cpf, nascimento (LocalDate).
 - **Comportamentos (Métodos):**
 - getHistoricoCompleto() : List<Matricula>: Fornece o histórico acadêmico completo do aluno, retornando todas as suas matrículas.
 - estaMatriculadoEm(turma : Turma) : boolean: Verifica se o aluno já possui uma matrícula ativa na turma especificada.
- **Disciplina**
 - **Representa:** A matéria de forma abstrata e atemporal (ex: "Programação Orientada a Objetos").
 - **Atributos:** nome, ementa.
 - **Comportamentos (Métodos):**
 - criarTurma(periodo: String, professor: Professor) : Turma: Atua como um método "fábrica" para criar uma nova oferta (Turma) desta disciplina.
- **Turma**
 - **Representa:** A oferta de uma Disciplina em um período específico, com um professor responsável e um conjunto de alunos matriculados. É a entidade central do modelo.
 - **Atributos:** periodo (ex: "2025/2").
 - **Comportamentos (Métodos):**

- matricularAluno(discente : Discente) : Matricula: Realiza a inscrição de um aluno nesta turma, aplicando regras de negócio (como verificação de vagas) e criando o registro de Matricula.
- listarAlunosMatriculados() : List<Discente>: Retorna a lista de todos os discentes inscritos.
- adicionarAula(conteudo : String, dataHora : LocalDateTime) : Aula: Cria um novo registro de encontro (aula) para esta turma.

- **Matricula**

- **Representa:** O vínculo formal e único de um Discente com uma Turma. É o registro que armazena todo o desempenho do aluno naquela turma específica.
- **Atributos:** dataMatricula (LocalDate), mediaFinal (Double), status (String).
- **Comportamentos (Métodos):**
 - calcularMediaFinal() : Double: Calcula a média ponderada do aluno com base em todas as Avaliacoes associadas.
 - atualizarStatus(novoStatus : String) : void: Modifica o estado do aluno na turma (ex: para "Aprovado" ou "Reprovado").
 - calcularPercentualFrequencia() : Double: Calcula a porcentagem de presença com base nos registros de Presenca.
 - lancarAvaliacao(descricao : String, nota : Double, peso : Double) : Avaliacao: Cria e associa um novo evento avaliativo a esta matrícula.

- **Aula**

- **Representa:** Um encontro específico de uma Turma, com data, hora e conteúdo ministrado.
- **Atributos:** conteudo, dataHora (LocalDateTime).
- **Comportamentos (Métodos):**
 - lancarFrequencia(frequencias : Map<Matricula, String>) : void: Registra a presença ou falta dos alunos para este encontro.
 - adicionarMaterialDeApoio(descricao : String, url : String) : void: Anexa um link ou referência a um material de apoio para esta aula.

- **Avaliacao**

- **Representa:** Um evento avaliativo (prova, trabalho) com nota, pertencente a uma Matricula.
- **Atributos:** descricao, nota (Double), peso (Double).

Diagrama UML

