
Lista de Exercícios 05

Sistema de Academia 🏋️

Você já possui a classe **Aluno** e **PlanoTreino**.

Agora, incremente o projeto considerando:

1. Nova classe:

- Crie as classes **Instrutor** e **Exercicios**, com atributos como nome, especialidade etc.

Classe Instrutor

- **Atributos e Métodos**
 - nome (String) – Nome completo do instrutor.
 - cref (String) – Registro profissional do Conselho de Educação Física.
 - especialidade (String) – Área de atuação (ex.: “Musculação”, “Crossfit”, “Pilates”).
 - alunos (List<Aluno>) – Lista de alunos associados a ele.
 - mentor (Instrutor) – Associação reflexiva (um instrutor pode ter outro como mentor).
 - Construtor(es) para inicializar nome, CREF e especialidade.
 - Getters e setters para atributos.
 - adicionarAluno(Aluno aluno) – Associa um aluno ao instrutor.
 - removerAluno(Aluno aluno) – Remove a associação com um aluno.
 - setMentor(Instrutor mentor) – Define o mentor do instrutor.
 - getResumo() – Retorna uma String para exibir os dados do instrutor.

○

Classe Exercicio

- Representa cada atividade dentro de um PlanoTreino.
- **Atributos e Métodos:**
 - nome (String) → Ex.: “Supino Reto”
 - series (int) → Ex.: 3
 - repeticoes (int) → Ex.: 12
 - carga (double) → Ex.: 30.5 (kg)
 - Construtor(es).
 - getResumo() Retorna uma String para exibir os dados do exercício.

○

2. Associação

- Um **Aluno** deve estar vinculado a um **Instrutor**.
- Um **Instrutor** pode orientar **vários alunos**.

3. Associação com multiplicidade

- Um **Aluno** pode ter **0 ou 1** plano de treino ativo.
- Um **PlanoTreino** pertence a **1 aluno apenas**.

4. Composição

- Um **PlanoTreino** é composto por **Exercícios**.
- Se o plano for removido, todos os exercícios desaparecem também.

5. Associação reflexiva

- Um **Instrutor** pode ter outro instrutor como **mentor** (associação reflexiva da classe Instrutor).

6. Diagrama UML

- Atualize o diagrama de classes incluindo Instrutor, Exercício e todos os novos relacionamentos com multiplicidade (1..*, 0..1 etc.).

7. Implementação em Java

- Crie as classes e métodos necessários.
- Atualize a classe Main para **testar os cenários**:
 - Criar alunos e instrutores.
 - Associar um aluno a um instrutor.
 - Definir um plano de treino com exercícios.
 - Testar a relação de mentor entre instrutores.