

## Proposta final de projeto para a disciplina Programação Orientada a Objetos

### Sistema de Gerenciamento de Academia

O projeto tem como objetivo simular o funcionamento de uma academia, permitindo o cadastro de alunos, personal trainers, planos e a gestão de agendamentos.

#### Requisitos mínimos:

#### Classes:

O sistema deverá ter as seguintes classes obrigatoriamente. (Podendo ter classes adicionais perante a necessidade descoberta por cada grupo):

- **Pessoa**: Classe abstrata com atributos como *nome*, *CPF*, *data de nascimento*, *contato* e *senha*.
  - **Aluno**: Subclasse de Pessoa com atributos como *plano contratado*, *data de matrícula* e *avaliações físicas*.
  - **PersonalTrainer**: Subclasse de Pessoa com atributos como *especialidade*, *CREF* e *horário de atendimento*.
  - **Funcionario**: Subclasse de Pessoa com atributo *cargo*, para identificar se é um atendente, gerente, etc.
- **Plano**: Classe com atributos como *nome*, *duração*, *valor*, e *descrição* (ex: musculação, natação, etc.).
- **Agendamento**: Classe com atributos como *data*, *horário*, *aluno* e *personal trainer*. Aqui são atributos mínimos, podem ser incluídos novos, como um status, para confirmar se a avaliação foi feita, se foi cancelada, etc.
- **Avaliacao**: Classe com atributos *aluno*, *data*, *personal trainer* e *descricao*.
- **SistemaAcademia** (Classe main): Gerencia a interação com o usuário através de um menu.

#### Características de Funcionamento:

Este sistema será executado como um menu interativo no console.

Na versão final (a ser entregue) é esperado que o sistema seja populado com os valores referentes aos alunos, planos e funcionários (equipe e personal trainer) através da leitura de um arquivo de texto. Para testes iniciais, entretanto, valores podem ser atribuídos diretamente.

Login: O usuário informa seu CPF e senha. O sistema identifica se é aluno, funcionario ou personal trainer e valida se a senha digitada está correta. Para cada tipo de usuário, haverá um menu diferente:

#### Caso 01:

Caso o usuário seja um **Aluno**, o sistema fornecerá um menu com as opções:

1. Visualizar dados pessoais e plano contratado.
2. Solicitar agendamento de horário com personal trainer.
3. Visualizar histórico de agendamentos.
4. Cancelar agendamento.
5. Visualizar avaliações físicas.

#### **Caso 02:**

Caso o usuário seja um **Personal Trainer**, o sistema fornecerá um menu com as opções:

1. Visualizar agenda de atendimentos.
2. Registrar avaliações físicas dos alunos.
3. Visualizar lista de avaliações realizadas.

#### **Caso 03:**

Caso o usuário seja um **Funcionário**, o sistema fornecerá um menu com as opções:

1. Cadastrar novo plano
2. Cadastrar novo aluno
3. Cadastrar novo Personal Trainer
4. Emitir relatório de planos
5. Emitir relatório de alunos
6. Emitir relatório de equipe (funcionários e personal trainers)
7. Emitir relação de avaliações físicas por período

#### **Considerações finais:**

Toda operação (cadastro, agendamento e avaliação) deverá ser registrada em um arquivo de texto de saída que armazena as operações realizadas durante aquela execução do sistema.

Todo relatório gerado deve ser registrado em um arquivo texto de saída individual. O sistema deve realizar ao menos um tratamento de erros personalizado

**OBS:** Acima de tudo, desenvolvimento é um trabalho de criatividade. Pensem bem, conversem entre si e idealizem como deve ser a modelagem e o fluxo da aplicação antes de efetivamente partir pro código.

**Boa sorte!**

Jacqueline Oliveira