



# PROJETO DE SISTEMAS DA INFORMAÇÃO

---

**Atividade Prática – ATP:**  
**Aplicar conhecimento em *design* orientado a objetos**

**Atividade Individual - Grupo 222:**  
**Integrante:** Vanessa Milani Ratusznei  
**Curso:** Análise e Desenvolvimento de Sistemas

---



## Etapa 1: Diagrama de Casos de Uso

Descrição do Diagrama de Casos de Uso do Sistema de Academia:

### Atores:

- Aluno
- Professor
- Administrador
- Médico
- Sistema Financeiro

### Atores e Casos de Uso:

#### Aluno:

- Consultar Treinos Diários
  - 1. Associado diretamente ao Aluno
  - 2. Include: Registrar Treinamento Programado
- Agendar Aulas
  - 1. Associado diretamente ao Aluno
  - 2. Include: Consultar Agenda via Web
- Agendar Avaliações Físicas
  - 1. Associado diretamente ao Aluno
  - 2. Include: Consultar Agenda via Web
- Registrar Feedback sobre Treino
  - 1. Associado diretamente ao Aluno
- Consultar Evolução Física
  - 1. Associado diretamente ao Aluno
  - 2. Include: Manter Evolução Física do Aluno
- Comunicar-se com Professor
  - 1. Associado diretamente ao Aluno
  - 2. Include: Acessar chat professor/aluno

#### Professor:

- Registrar Treinamento Programado
  - 1. Associado diretamente ao Professor
- Manter Evolução Física do Aluno
  - 1. Associado diretamente ao Professor
  - 2. Extend: Registrar Feedback sobre Treino
- Manter Avaliações Periódicas do Aluno
  - 1. Associado diretamente ao Professor



- 
- 2. Extend: Registrar Feedback sobre Treino
  - Comunicar-se com Aluno
    - 1. Associado diretamente ao Professor
    - 2. Include: acessar chat professor/aluno

#### Administrador:

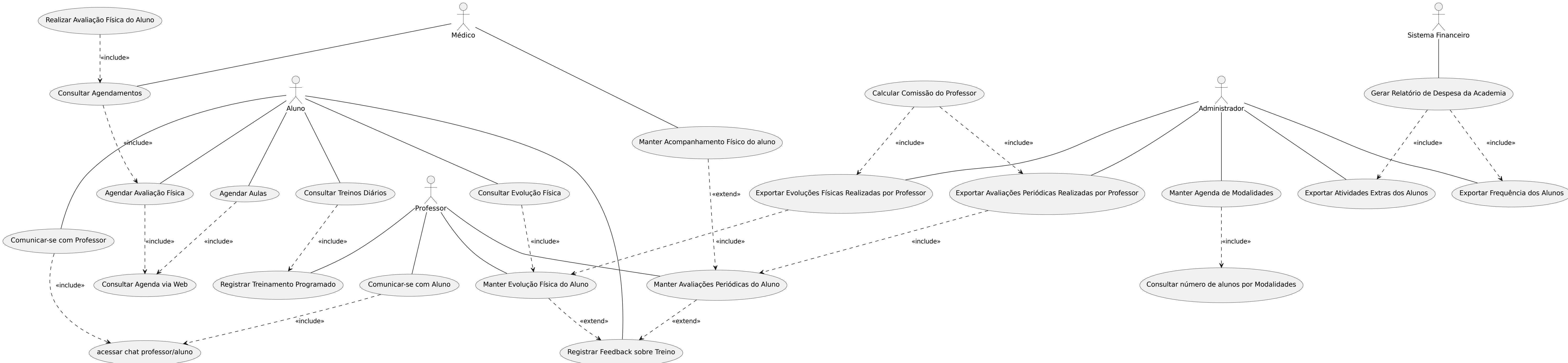
- Exportar Atividades Extras dos Alunos
  - 1. Associado diretamente ao Administrador
- Exportar Frequência dos Alunos
  - 1. Associado diretamente ao Administrador
- Exportar Avaliações Periódicas Realizadas por Professor
  - 1. Associado diretamente ao Administrador
  - 2. Include: Manter Avaliações Periódicas do Aluno
- Exportar Evoluções Físicas Realizadas por Professor
  - 1. Associado diretamente ao Administrador
  - 2. Include: Manter Evolução Física do Aluno
- Calcular Comissão do Professor
  - 1. Include: Exportar Avaliações Periódicas Realizadas por Professor e Exportar Evoluções Físicas Realizadas por Professor
- Manter Agenda de Modalidades
  - 1. Associado diretamente ao Administrador
  - 2. Include: Consultar Número de Alunos por Modalidade

#### Médico:

- Manter Acompanhamento Físico do Aluno
  - 1. Associado diretamente ao Médico
  - 2. Extend: Manter Avaliação Periódica do Aluno
- Realizar Avaliação Física do Aluno
  - 1. Include: Consultar Agendamentos
- Consultar Agendamentos
  - 1. Associado diretamente ao Médico
  - 2. Include: Agendar Avaliação Física

#### Sistema Financeiro:

- Gerar Relatório de Despesa da Academia
  - 1. Associado diretamente ao Sistema Financeiro
  - 2. Include: Exportar Atividades Extras dos Alunos e Exportar Frequência dos Alunos





## **Etapa 2 - Especificação de Caso de Uso:**

- **Nome do Caso de Uso:**

Consultar Treinos Diários

- **Breve Descrição:**

Permite ao aluno consultar os treinos diários que foram registrados pelo professor.

- **Atores:**

Aluno

- **Precondições:**

1. O aluno deve estar autenticado no sistema.
2. O professor deve ter registrado o treino programado.

- **Fluxo Principal:**

1. O caso de uso inicia quando o aluno seleciona a opção "Consultar Treinos Diários" no sistema.
2. O sistema verifica a autenticidade do aluno.
3. O sistema recupera o treino programado registrado pelo professor para o dia atual.
4. O sistema exibe os detalhes do treino diário para o aluno.
5. O aluno visualiza o treino diário.

- **Fluxos Alternativos:**

Treino não registrado - Se não houver treino registrado pelo professor para o dia atual:

1. O sistema informa ao aluno que não há treino registrado para o dia atual.
2. O aluno retorna à tela principal.



- **Fluxos de Exceção:**

Falha na Autenticação - Se a autenticação do aluno falhar:

1. O sistema exibe uma mensagem de erro de autenticação.
2. O aluno retorna à tela de login.

- **Regras de Negócio:**

1. O treino diário só pode ser consultado por alunos autenticados.
2. O treino diário deve ser registrado previamente pelo professor.

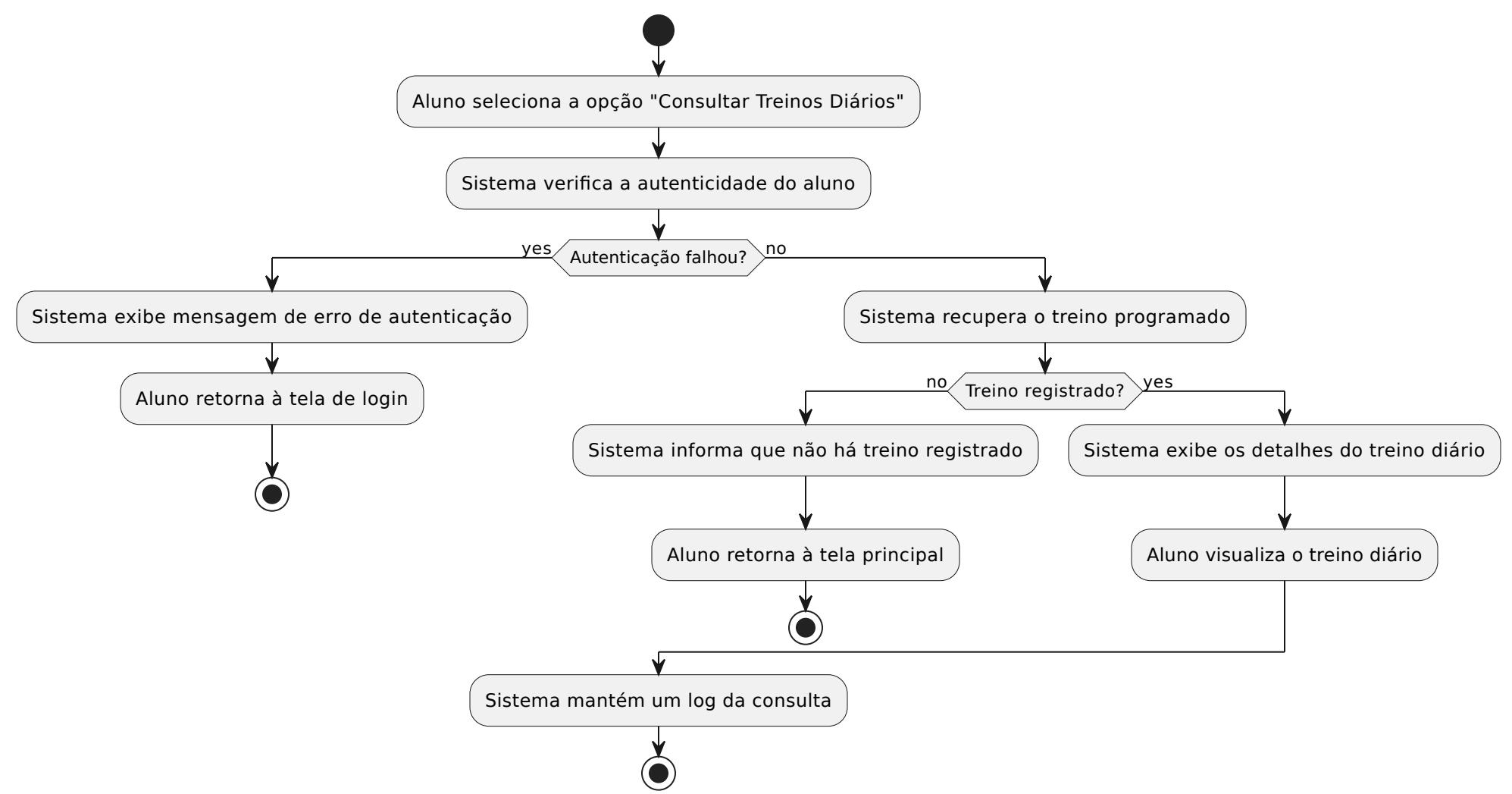
- **Pós-condições:**

1. O aluno visualiza o treino diário ou é informado de que não há treino registrado para o dia atual.
2. O sistema mantém um log da consulta realizada pelo aluno para fins de auditoria.

### **Etapa 3 - Diagrama de Atividades:**

Descrição do Diagrama de Atividades para o Caso de Uso "Consultar Treinos Diários":

- 1) O aluno seleciona a opção "Consultar Treinos Diários".
- 2) O sistema verifica a autenticidade do aluno.
  - a) Se a autenticação falhar:
    - i) O sistema exibe uma mensagem de erro de autenticação.
    - ii) O aluno retorna à tela de login.
  - b) Se a autenticação for bem-sucedida:
    - i) O sistema recupera o treino programado.
- 3) O sistema verifica se o treino está registrado.
  - a) Se não houver treino registrado:
    - i) O sistema informa que não há treino registrado.
    - ii) O aluno retorna à tela principal.
  - b) Se houver treino registrado:
    - i) O sistema exibe os detalhes do treino diário.
    - ii) O aluno visualiza o treino diário.
- 4) O sistema mantém um log da consulta.





## **Etapa 4 - Diagrama de Classes:**

Descrição do Diagrama de Classes do Sistema Academia:

### **Classes Identificadas:**

#### **1. Aluno:**

- Atributos:
  - id: int
  - nome: String
  - email: String
  - senha: String
- Métodos:
  - consultarTreinosDiarios()
  - agendarAulas()
  - agendarAvaliacaoFisica()
  - registrarFeedback()
  - consultarEvolucaoFisica()
  - comunicarComProfessor()

#### **2. Professor:**

- Atributos:
  - id: int
  - nome: String
  - email: String
  - senha: String
- Métodos:
  - registrarTreinamentoProgramado()



- manterEvolucaoFisicaAluno()
- manterAvaliacaoPeriodicaAluno()
- comunicarComAluno()

### 3. Administrador:

- Atributos:
  - id: int
  - nome: String
  - email: String
  - senha: String
- Métodos:
  - exportarAtividadesExtrasAlunos()
  - exportarFrequenciaAlunos()
  - exportarAvaliacaoPeriodicaProfessor()
  - exportarEvolucaoFisicaProfessor()
  - calcularComissaoProfessor()
  - manterAgendaModalidades()

### 4. Medico:

- Atributos:
  - id: int
  - nome: String
  - email: String
  - senha: String
- Métodos:
  - manterAcompanhamentoFisicoAluno()



- realizarAvaliacaoFisicaAluno()
- consultarAgendamentos()

## 5. SistemaFinanceiro:

- Método:
  - gerarRelatorioDespesaAcademia()

## 6. Treino:

- Atributos:
  - id: int
  - data: Date
  - descricao: String

## 7. AvaliacaoFisica:

- Atributos:
  - id: int
  - data: Date
  - resultado: String

## 8. Feedback:

- Atributos:
  - id: int
  - data: Date
  - comentario: String



## Relacionamentos Identificados:

- Aluno - Treino
  - Método: consultarTreinosDiarios()
- Aluno - AvaliacaoFisica
  - Método: agendarAvaliacaoFisica()
- Aluno - Professor
  - Método: registrarFeedback()
  - Método: comunicarComProfessor()
- Professor - Treino
  - Método: registrarTreinamentoProgramado()
- Professor - AvaliacaoFisica
  - Método: manterEvolucaoFisicaAluno()
  - Método: manterAvaliacaoPeriodicaAluno()
- Medico - AvaliacaoFisica
  - Método: realizarAvaliacaoFisicaAluno()

