

# PLANO DE PROJETO – Gerenciador de Academias

## 1. Identificação do Projeto

- Nome: Gerenciador de Academias
- Data de Início: 01/maio/2025
- Data Estimada de Conclusão: 30/junho/2025
- Gerente de Projeto / Scrum Master: (a ser definido)
- Cliente / Product Owner: (representante da academia ou professor orientador)
- Equipe de Desenvolvimento: 3 a 5 membros

## 2. Objetivo do Projeto

Desenvolver um sistema web para academias com dois módulos principais:

- Módulo 1: Gerenciamento de alunos (cadastro, planos, matrículas e pagamentos)
- Módulo 2: Gerenciamento de atividades (treinos, exercícios, acompanhamento)

## 3. Escopo

Incluído no escopo:

- Cadastro e consulta de alunos, planos e pagamentos
- Registro de presença
- Cadastro e associação de treinos e exercícios
- Emissão de relatórios
- Interface web (responsiva)
- Autenticação com JWT
- API RESTful com documentação via Swagger

Fora do escopo:

- Aplicativo mobile
- Integração com dispositivos de treino
- Módulo de nutrição ou avaliação física
- Integrações com sistemas bancários

#### 4. Cronograma Geral (Macro)

Semana	Entregas Previstas
1-2	Planejamento, setup de ambiente e requisitos
3-4	Cadastro de alunos, planos e pagamentos
5-6	Módulo de treinos e associação com alunos
7-8	Relatórios, segurança e testes finais

#### 5. Metodologia de Desenvolvimento

- Metodologia: Ágil - Scrum
- Sprints: 4 Sprints de 2 semanas
- Cerimônias Scrum:
  - Sprint Planning (início da sprint)
  - Daily Scrum (diária de 15 minutos)
  - Sprint Review (apresentação do incremento)
  - Sprint Retrospective (análise de melhorias)

#### 6. Papéis e Responsabilidades

Papel	Responsável	Atividades principais
Product Owner	Representante do cliente/professor	Priorização do backlog, validação das entregas
Scrum Master	Facilitador	Remoção de impedimentos, condução das cerimônias
Dev Team	Equipe de devs (3-5 alunos)	Codificação, testes, documentação técnica

#### 7. Recursos Necessários

- Ferramentas de Desenvolvimento: Java, Spring Boot, MySQL, React ou Angular
- Gerenciamento: Trello, Jira ou GitHub Projects
- Versionamento: Git + GitHub
- Testes: Postman, Swagger, JUnit
- Infraestrutura: Hospedagem local ou em nuvem (Heroku, Vercel, Render, etc.)

## 8. Riscos e Estratégias de Mitigação

Risco	Probabilidade	Impacto	Mitigação
Falta de tempo dos membros	Alta	Alto	Reuniões curtas diárias para controle
Dificuldades técnicas com Spring Boot	Média	Médio	Mentoria ou uso de tutoriais / pares
Atrasos na entrega de requisitos	Média	Alto	Reuniões frequentes com o Product Owner
Perda de dados	Baixa	Alto	Backups automáticos do banco de dados

## 9. Critérios de Sucesso

- Funcionalidades entregues conforme escopo
- Relatórios gerados corretamente
- Sistema com segurança básica (login, JWT)
- Interface intuitiva e responsiva
- Documentação completa (Swagger, README, banco)

## 10. Comunicação

- Reuniões semanais: Segunda e sexta-feira
- Daily Scrum: 15 minutos por dia, via Discord ou presencial
- Ferramentas de comunicação: WhatsApp, Discord, Slack
- Repositório: GitHub (privado ou público com controle de branches)