Projeto de Site Integrado — Academia [Nome da academia]

# Este projeto tem como objetivo o desenvolvimento de um site para uma academia, com funcionalidades que incluem a gestão de alunos, professores, planos de serviços, agendamentos, produtos e outras áreas relacionadas. O projeto é concebido principalmente para fins acadêmicos, como parte do portfólio do desenvolvedor, e poderá ser utilizado como base para um Trabalho de Conclusão de Curso (TCC) ou monografia.

O site será desenvolvido com a intenção de demonstrar habilidades técnicas adquiridas durante a formação em Engenharia de Software, e não tem a finalidade de venda. Contudo, caso a proprietária do estabelecimento manifeste interesse na aquisição do projeto ou decida patrociná-lo no futuro, tal negociação será considerada como uma oportunidade adicional, gerando lucro para o desenvolvedor.

Este trabalho busca ser uma contribuição para o aprendizado prático e, caso haja interesse em comercializar ou patrocinar o projeto, será uma possibilidade a ser considerada posteriormente.

# 1. Descrição do Projeto

Objetivo: Desenvolver um site moderno, interativo e funcional que centralize todos os serviços oferecidos pela academia, atendendo de forma integrada alunos, professores e gestores.

Proposta de Valor: Facilitar o acesso a informações, otimizar a comunicação entre alunos e equipe, melhorar o controle de serviços e expandir as vendas de produtos da academia.

# 2. Funcionalidades por Área

## Área Pública (Visitantes)

- Página inicial com destaque dos serviços  
- Páginas de serviços (Academia, Pilates, Estética, Fisioterapia)  
- Página "Sobre nós"  
- Página de contatos (telefones, endereço, mapa, WhatsApp)  
- Visualização de planos e combos com fotos e preços

## Área do Aluno (com login)

- Consulta de avaliação física e histórico de medidas  
- Agendamento de avaliações e sessões (horários disponíveis)  
- Acesso a planos e contratação online

## Área do Professor (com login)

- Lançamento de medidas e avaliações físicas  
- Gestão de agenda de atendimentos

## Área de E-commerce

- Loja de suplementos e roupas de academia  
- Carrinho de compras e checkout com pagamento online

# 3. Tecnologias a serem usadas

- Frontend: HTML5, CSS3 (com Tailwind CSS), JavaScript (React.js)  
- Backend: Node.js (Express)  
- Banco de dados: PostgreSQL  
- Autenticação: JWT (JSON Web Tokens)  
- Hospedagem: Vercel ou Render (Frontend) / Railway ou Render (Backend)  
- Integrações: Pagamento (Stripe ou Mercado Pago), Agendamento (FullCalendar)  
- Outras ferramentas: Git e GitHub (versionamento), Figma (protótipos)

# 4. Fluxo de Trabalho

1. Planejamento — Definição de escopo e protótipos (Figma)  
2. Design — Criação dos layouts e identidade visual  
3. Desenvolvimento  
 - Site institucional  
 - Área do aluno/professor  
 - E-commerce  
4. Integrações — Banco de dados, login, agendamento e pagamento  
5. Testes — Testes de funcionalidades, ajustes e correções  
6. Lançamento — Publicação e entrega final

# 5. Cronograma (início: 01/06/2025)

|  |  |  |
| --- | --- | --- |
| Fase | Início | Término |
| Planejamento | 01/06/2025 | 11/06/2025 |
| Design | 12/06/2025 | 28/06/2025 |
| Site institucional | 29/06/2025 | 27/07/2025 |
| Área do aluno/professor | 28/07/2025 | 08/09/2025 |
| E-commerce | 09/09/2025 | 13/10/2025 |
| Integrações | 14/10/2025 | 07/11/2025 |
| Testes | 08/11/2025 | 29/11/2025 |
| Ajustes finais | 30/11/2025 | 10/12/2025 |

# 6. Segurança do Sistema

## A. Autenticação e Controle de Acesso

- Utilização de \*\*JWT (JSON Web Tokens)\*\* ou \*\*OAuth2\*\* para autenticação de usuários e controle de acesso.  
- Implementação de \*\*multifactor authentication (MFA)\*\*.  
- Controle de Acesso baseado em Função (RBAC), garantindo que usuários tenham permissões adequadas para suas funções.

## B. Criptografia

- \*\*Hashing\*\* de senhas com algoritmos seguros como \*\*bcrypt\*\* ou \*\*Argon2\*\*.  
- Criptografia de dados sensíveis (como informações de pagamento) usando \*\*TLS/SSL\*\* tanto para tráfego quanto armazenamento.

## C. Proteção contra Injeção de SQL

- Uso de \*\*prepared statements\*\* ou \*\*ORMs\*\* para evitar injeções de SQL.  
- Validação e filtragem de entradas para prevenir comandos maliciosos no banco de dados.

## D. Proteção contra Ataques de Cross-Site Scripting (XSS)

- Validação e \*\*sanitização de entradas de usuários\*\* (formularios, comentários, etc.) para evitar XSS.  
- Implementação de \*\*Content Security Policy (CSP)\*\* para mitigar ataques XSS.

## E. Proteção contra CSRF (Cross-Site Request Forgery)

- Uso de \*\*tokens CSRF\*\* para operações críticas (como alterações no perfil ou pagamentos) para garantir solicitações legítimas.

# 7. Proteção do Banco de Dados

## A. Backup Regular

- Realização de \*\*backups automáticos diários\*\* do banco de dados.  
- Armazenamento seguro dos backups em locais criptografados e externos, como \*\*cloud storage\*\*.

## B. Acesso Restrito ao Banco de Dados

- Implementação do \*\*princípio do menor privilégio\*\* para o acesso ao banco de dados.  
- Uso de autenticação forte e credenciais armazenadas de forma segura.

## C. Monitoramento de Acessos

- Ativação de \*\*logs de auditoria\*\* no banco de dados para registrar acessos e alterações.  
- Utilização de \*\*ferramentas de monitoramento de intrusão\*\* para detectar tentativas de acesso não autorizado.

## D. Isolamento de Dados

- Manutenção de dados sensíveis (como pagamentos) em tabelas isoladas e com criptografia de alto nível (AES-256).

# 8. Manutenção Periódica

## A. Atualizações Regulares

- Garantir que todas as dependências do sistema (frameworks, bibliotecas, etc.) estejam sempre atualizadas.  
- Monitoramento constante de atualizações de segurança para pacotes e frameworks utilizados.

## B. Monitoramento Contínuo

- Implementação de \*\*monitoramento de performance\*\* e \*\*detecção de falhas\*\* em tempo real.  
- Verificação regular dos \*\*logs de erro\*\* para identificar problemas no sistema.

## C. Testes de Segurança Regulares

- Realização de \*\*testes de penetração\*\* regulares para identificar vulnerabilidades de segurança.  
- Utilização de ferramentas de \*\*verificação de vulnerabilidades\*\* em bibliotecas e pacotes do sistema.

## D. Backup e Recuperação

- \*\*Testes periódicos\*\* de recuperação de backups para garantir que a restauração dos dados seja rápida e eficiente.

## E. Auditorias e Relatórios de Segurança

- \*\*Auditorias regulares\*\* dos logs de acesso para detectar comportamentos suspeitos e garantir que o sistema esteja em conformidade com as normas de privacidade.  
- \*\*Relatórios de conformidade\*\* com a LGPD, GDPR, ou outras legislações de proteção de dados.