

Documento de Documentação do Projeto Move-On.

SPRINT 1

1. Ideia Inicial do Projeto

O projeto, intitulado **Move-on**, tem como principal objetivo oferecer uma plataforma web interativa e acessível para a gestão completa de academias. O sistema permite o cadastro de alunos e instrutores, a gestão de planos e treinos, o controle de pagamentos e a organização de agendas de aulas. Além disso, disponibiliza relatórios financeiros e de frequência, tudo em uma interface intuitiva voltada para eficiência, transparência e acompanhamento da evolução dos alunos.

O público-alvo principal são academias, personal trainers e alunos que desejam praticidade no gerenciamento de suas rotinas de treino, controle de pagamentos e acesso a informações claras sobre desempenho e frequência. Espera-se que o projeto gere impacto positivo ao facilitar a administração do negócio e ao promover uma experiência mais organizada e motivadora para os alunos.

Como o projeto foi desenvolvido por uma **equipe multidisciplinar**, ele também representa uma oportunidade de aprendizado prático em desenvolvimento web full-stack com foco em arquitetura MVC, boas práticas de segurança e design responsivo, além de explorar soluções digitais aplicadas ao setor fitness.

2. Levantamento de Requisitos

Por ser um projeto autoral, os requisitos foram definidos com base em análise de mercado e objetivos pessoais de aprendizado e usabilidade. Estes foram os papéis exercidos no desenvolvimento:

- **Desenvolvedores Full-Stack:** Responsáveis por toda a lógica do sistema (front-end, back-end e banco de dados).
- **Designers UX/UI:** Criação do layout, identidade visual e responsividade da plataforma.
- **Equipe de Documentação e Planejamento:** Elaboração da documentação, roteiros e organização do escopo.

3. Esboço da Solução

4. Planejamento do Desenvolvimento com SCRUM

Funcionalidades essenciais definidas:

Como o projeto foi desenvolvido por uma equipe, o processo de SCRUM foi utilizado como forma de organização do trabalho, com sprints semanais e reuniões de alinhamento entre os integrantes.

- Estrutura inicial do projeto (pasta MVC, configuração)
- Criação do banco de dados
- Criação da página de login/cadastro e níveis de acesso (administrador, aluno, instrutor)
- Lógica de autenticação e autorização
- Cadastro e gestão de alunos e instrutores (CRUD completo)
- Gestão de planos e treinos personalizados
- Controle de pagamentos e situação financeira
- Agenda de aulas e frequência
- Relatórios financeiros e de desempenho dos alunos
- Dashboard com indicadores principais
- Estilização geral e responsividade
- Validação e testes com usuários reais
- Publicação do projeto

Sprints:

- **Sprint 1:** Estrutura MVC, banco de dados, autenticação e níveis de acesso
- **Sprint 2:** Cadastro de alunos/instrutores, gestão de planos e treinos, controle de pagamentos
- **Sprint 3:** Agenda de aulas, relatórios, dashboard, estilização e testes finais

5. Documentação da Tecnologia

Tecnologias Utilizadas:

- **HTML5, CSS3 e JavaScript:** Front-end
- **PHP:** Lógica do sistema
- **MySQL + PDO:** Banco de dados e segurança de acesso
- **Arquitetura MVC:** Separação de responsabilidades no código
- **XAMPP:** Ambiente local de desenvolvimento
- **GitHub:** Versionamento e controle de código
- **Visual Studio Code:** IDE de desenvolvimento

Padrões de Código:

- **PHP:**
 - o Comentários explicativos
 - o Nomes de função descritivos
 - o Evitar lógica pesada nas Views
- **HTML/CSS:**
 - o Classes descritivas
 - o Estilo externo via arquivo .css
 - o Uso de variáveis CSS e @media para responsividade

O Repositório de Github pode ser achado [asssqui](#)

6. Testes e Validação

Para garantir a qualidade da interface e funcionalidade do sistema, foram realizados testes com usuários reais (alunos, instrutores e gestores de academia) seguindo este roteiro:

- Apresentação da proposta do projeto
- Solicitação de uso do sistema sem explicações para avaliar a usabilidade
- Anotação de pontos de confusão ou dificuldades na navegação (ex.: cadastro de alunos, registro de treinos, controle de pagamentos)
- Coleta de sugestões de melhoria diretamente com os usuários
- Ajustes implementados com base no feedback recebido

Além disso, foram realizados testes de segurança, incluindo:

- Validação de formulários (cadastro de alunos, planos, pagamentos)

- Proteção contra injeção de SQL com uso de *prepared statements* via PDO
- Verificação de inputs e sessões para evitar acessos não autorizados
- Encriptação das senhas dos usuários (administradores, instrutores e alunos)