



Sistema de Gerenciamento de Academia  
*Documentação Técnica*

**Autores:**

João Gabriel Gonzalez  
Henrique Rodrigues Motta

**SENAI LIMEIRA “LUIZ VARGA”**  
2025

## SUMÁRIO

SUMÁRIO.....	2
1.VISÃO DO PRODUTO .....	3
1.1 Objetivos Principais .....	3
1.2 Diferenciais .....	3
2.LEVANTAMENTO DE REQUISITOS .....	4
2.1 Requisitos Funcionais .....	4
2.2 Requisitos Não Funcionais.....	5
2.3 Regras de Negócio.....	5
3.ARQUITETURA DO SISTEMA .....	6
3.1 Padrão MVC .....	6
3.2 Roteamento .....	6
4.MÓDULOS DO SISTEMA.....	7
4.1 Dashboard Analytics.....	7
4.2 Controle de Acesso RFID .....	7
4.3 SAGEF - Sistema de Gestão de Treinos.....	7
4.4 Agendamento de Aulas .....	7
4.5 Gestão de Usuários .....	8
5.TECNOLOGIAS UTILIZADAS .....	9
6.ESTRUTURA DO PROJETO .....	10
7.Banco de Dados.....	11
7.1 Estrutura Geral .....	11
7.2 Relacionamentos Principais .....	12
8.WIREFRAME .....	13
8.1 Página Inicial (Landing Page) .....	13
8.2 Academias Mais Próximas .....	13
9.METODOLOGIA Scrum .....	14
9.1 Papéis.....	14
9.2 Quadro Kanban .....	14
9.3 Sprints.....	14
9.4 Cerimônias.....	15
10.BACKLOG DO PRODUTO .....	16
11.CONCLUSÃO.....	17



## 1. VISÃO DO PRODUTO

TechFit é uma plataforma integrada para academias da franquia TechFit, desenvolvida com o objetivo de modernizar a gestão operacional, melhorar a experiência dos alunos e automatizar processos. O sistema reúne diferentes módulos em um único ambiente digital, oferecendo uma solução completa e escalável.

### 1.1 Objetivos Principais

- Reduzir processos manuais que geram erros, retrabalho e lentidão
- Oferecer autonomia ao aluno para agendar, cancelar ou acompanhar treinos sem depender de atendimento presencial
- Garantir segurança e confiabilidade com autenticação digital via RFID
- Apoiar decisões gerenciais com relatórios automáticos e dashboards intuitivos
- Padronizar a metodologia de trabalho interno usando frameworks ágeis (Scrum/Kanban)
- Ampliar a competitividade da academia oferecendo tecnologia que melhora a experiência do aluno

### 1.2 Diferenciais

- **Plataforma modular:** cada funcionalidade pode ser usada de forma independente ou integrada
- **Controle de Acesso IoT:** integração com Arduino e leitores RFID para entrada automatizada
- **Dashboard Analytics:** gráficos e métricas em tempo real com Chart.js
- **Interface intuitiva:** design responsivo com Bootstrap 5 para alunos e administradores
- **Escalabilidade:** arquitetura preparada para inclusão de novos módulos (SAGEF automatizado, gamificação, app mobile)

## 2. LEVANTAMENTO DE REQUISITOS

O levantamento de requisitos foi realizado através da análise das necessidades operacionais de uma academia, identificando as funcionalidades essenciais para o sistema TechFit.

### 2.1 Requisitos Funcionais

ID	Requisito	Descrição	Prioridade
RF01	Cadastro de Usuários	O sistema deve permitir o cadastro de alunos com dados pessoais, medidas corporais, objetivos e disponibilidade de horários	Alta
RF02	Gestão de Planos	O sistema deve permitir criar, editar e visualizar planos com diferentes preços e benefícios	Alta
RF03	Controle de Pagamentos	O sistema deve registrar pagamentos, controlar vencimentos e status financeiro dos usuários	Alta
RF04	Montagem de Treinos	O sistema deve permitir criar treinos personalizados associando exercícios com séries, repetições e ordem de execução	Alta
RF05	Catálogo de Exercícios	O sistema deve manter uma base de exercícios categorizados por grupo muscular, dificuldade e equipamento	Média
RF06	Controle de Acesso RFID	O sistema deve registrar entradas na academia através de tags RFID, validando status do usuário	Alta
RF07	Registro de Entradas	O sistema deve armazenar histórico de acessos com data, hora, tipo de entrada e status	Média
RF08	Sistema de Pontuação	O sistema deve calcular e armazenar pontos por grupo muscular trabalhado, gamificando o treino	Baixa
RF09	Gestão de Funcionários	O sistema deve cadastrar funcionários com níveis de acesso diferenciados	Média
RF10	Agendamento de Aulas	O sistema deve permitir criar aulas com instrutor, data, horário, sala e controle de vagas	Média
RF11	Inscrição em Aulas	O sistema deve permitir que usuários se inscrevam em aulas e registrem presença e avaliação	Média
RF12	Dashboard Administrativo	O sistema deve exibir indicadores e gráficos de desempenho da academia	Média
RF13	Autenticação	O sistema deve permitir login de usuários e funcionários com diferentes níveis de permissão	Alta
RF14	First Training	O sistema deve oferecer um treino inicial guiado para novos usuários	Baixa

## 2.2 Requisitos Não Funcionais

ID	Requisito	Descrição	Categoria
RNF01	Responsividade	A interface deve se adaptar a diferentes tamanhos de tela (desktop, tablet, mobile)	Usabilidade
RNF02	Tempo de Resposta	O sistema deve responder às requisições em no máximo 3 segundos	Desempenho
RNF03	Disponibilidade	O sistema deve estar disponível 99% do tempo em horário comercial	Confiabilidade
RNF04	Compatibilidade	O sistema web deve funcionar nos navegadores Chrome, Firefox e Edge	Portabilidade
RNF05	Integração Arduino	O módulo RFID deve se comunicar com Arduino para leitura das tags	Interoperabilidade
RNF06	Banco de Dados	O sistema deve utilizar MySQL como SGBD	Tecnologia
RNF07	Interface Intuitiva	O sistema deve ser utilizável sem necessidade de treinamento extensivo	Usabilidade

## 2.3 Regras de Negócio

ID	Regra
RN01	Usuários com pagamento atrasado devem ter acesso bloqueado na catraca
RN02	Tags RFID expiradas ou inativas não permitem entrada
RN03	Aulas só podem ser ministradas por funcionários cadastrados
RN04	Inscrição em aulas é limitada ao número de vagas disponíveis
RN05	Cada usuário deve estar vinculado a um plano ativo
RN06	Funcionários possuem níveis de acesso: 1 (básico), 2 (intermediário), 3 (administrador)
RN07	Pontuação é calculada automaticamente com base nos exercícios realizados

### 3. ARQUITETURA DO SISTEMA

#### 3.1 Padrão MVC

O projeto segue o padrão Model-View-Controller (MVC) implementado em PHP puro, sem frameworks externos. Esta escolha proporciona controle total sobre a arquitetura, facilita o aprendizado dos fundamentos do padrão e reduz overhead desnecessário.

**Models:** Responsáveis pela lógica de negócio e comunicação com o banco de dados via PDO. Utilizam namespaces organizados (models\, models\acesso\, models\sagef\, etc.).

**Views:** Templates PHP que renderizam a interface do usuário, separados por área (funcionario/, usuario/, agendamento/).

**Controllers:** Intermediários que processam requisições, interagem com models e direcionam para views apropriadas.

#### 3.2 Roteamento

O sistema utiliza um roteador simples baseado em switch-case no arquivo index.php, mapeando URLs amigáveis para arquivos PHP específicos. As rotas são organizadas por área de acesso:

Rota	Descrição
<b>Área Pública</b>	
/	Home page com apresentação, planos e academias
/academias	Lista de unidades TechFit
/login	Autenticação de usuários e funcionários
<b>Área do Funcionário</b>	
/funcionario	Dashboard principal com analytics e gráficos
/funcionario/register/exercicios	Cadastro de exercícios
/funcionario/register/estudantes	Cadastro de alunos
/funcionario/register/classes	Cadastro de aulas em grupo
/funcionario/register/treino	Montagem manual de treinos personalizados
/funcionario/RFID	Monitor de leituras RFID
<b>Área do Usuário</b>	
/usuario	Painel principal do aluno
/usuario/profile	Perfil e dados pessoais
/usuario/user/schedule	Agenda de aulas
/usuario/user/training	Treinos atribuídos

## 4. MÓDULOS DO SISTEMA

### 4.1 Dashboard Analytics

O painel administrativo oferece visão em tempo real das operações da academia através de gráficos interativos (Chart.js) e tabelas dinâmicas. Funcionalidades incluem:

- Estatísticas gerais: usuários ativos, treinos ativos, aulas futuras, pagamentos atrasados
- Gráfico de distribuição de planos (pizza)
- Evolução de treinos por mês (linha)
- Pontuação por grupos musculares (barras)
- Tabelas: últimos acessos, exercícios populares, usuários mais ativos, próximas aulas
- Atualização em tempo real via API REST (api/dashApi.php)

### 4.2 Controle de Acesso RFID

Sistema de controle de entrada automatizado utilizando tecnologia IoT:

- **Hardware:** Arduino com módulo MFRC522 (leitor RFID)
- **Comunicação:** Serial via Python (rfid\_reader.py)
- **Validação:** Verificação de tag cadastrada e status de pagamento do aluno
- **Registro:** Histórico completo de entradas/saídas com data, hora e status
- **Models:** RFIDTags.php, RegistroEntrada.php

### 4.3 SAGEF - Sistema de Gestão de Treinos

O SAGEF (Sistema Automatizado de Geração de Exercícios Físicos) é responsável pela gestão de treinos personalizados. *Na versão atual, os treinos são criados manualmente pelos funcionários.*

**Funcionalidades implementadas:**

- Cadastro de exercícios com nome, tipo e nível de dificuldade
- Montagem de treinos associando exercícios a um aluno específico
- Definição de séries, repetições, carga e tempo de descanso
- Controle de status do treino (ativo/inativo)
- Sistema de pontuação por grupos musculares

**Evolução planejada:**

- Geração automática de treinos baseada em dados biométricos do aluno (peso, altura, objetivo)
- Ajuste dinâmico de cargas conforme feedback de desempenho
- Recomendações inteligentes de exercícios

### 4.4 Agendamento de Aulas

- Cadastro de aulas em grupo com nome, tipo, data, horário e instrutor
- Controle de vagas disponíveis e totais
- Inscrição de alunos em aulas
- Visualização de próximas aulas no dashboard
- **Models:** Aula.php, Participacao.php, ParticipacoesAula.php

## 4.5 Gestão de Usuários

Cadastro completo de alunos com dados biométricos e pessoais:

- Dados pessoais: nome, gênero, data de nascimento, endereço
- Dados biométricos: altura, peso, porcentagem de massa magra
- Informações de treino: objetivo, experiência anterior, tempo de treino, disponibilidade
- Plano e status de pagamento



## 5. TECNOLOGIAS UTILIZADAS

Categoria	Tecnologia	Aplicação
Backend	PHP 8+	Lógica de negócio, processamento de dados, APIs
	MySQL	Banco de dados relacional
	PDO	Abstração de banco com prepared statements
Frontend	HTML5/CSS3	Estrutura e estilos das páginas
	Bootstrap 5.3	Framework CSS responsivo, componentes
	SCSS/Sass	Pré-processador CSS com variáveis
	JavaScript ES6+	Interatividade, validações, chamadas AJAX
	Chart.js 4.5	Gráficos interativos do dashboard
IoT/Hardware	Arduino	Microcontrolador para leitura RFID
	MFRC522	Módulo leitor de cartões RFID
	Python	Script de comunicação serial

## 6. ESTRUTURA DO PROJETO

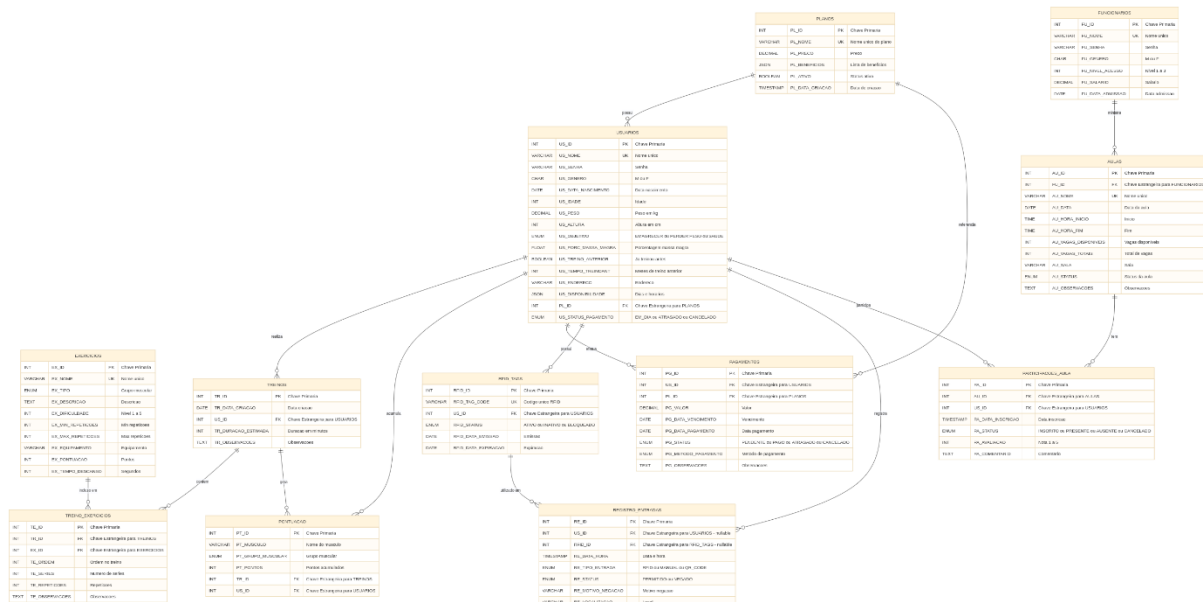
O projeto está organizado seguindo o padrão MVC com separação clara de responsabilidades:

### techfit-system/

- **api/** - Endpoints REST (dashApi.php)
- **Assets/** - Recursos estáticos
  - ino/ - Código Arduino (codigo\_arduino.ino)
  - js/ - Scripts JavaScript (dashboard.js)
  - scss/ - Estilos fonte (style.scss)
  - style/ - CSS compilado
- **config/** - Configurações (Database.php, Config.php)
- **controllers/** - Controladores organizados por módulo
  - acesso/ - DashboardController
  - agendamento/ - AulaController, CalendarioController
  - sagef/ - exercicioController, treinoController
- **core/** - Classes base (Session.php)
- **models/** - Models de dados
  - acesso/ - RFIDTags, RegistroEntrada
  - agendamento/ - Aula, Participacao
  - pagamento/ - Pagamentos, Planos
  - sagef/ - Exercicio, Treino, Pontuacao
- **public/** - Arquivos públicos (home, login, images)
- **views/** - Templates de interface
  - funcionario/ - Telas administrativas
  - usuario/ - Área do aluno
- **Documentação/** - Modelos de BD e prints
- **index.php** - Roteador principal

## 7. Banco de Dados

O banco de dados **TechFitDatabase** foi projetado para gerenciar uma academia de forma completa, abrangendo desde o controle de acesso até o acompanhamento de treinos e desempenho dos usuários.



### 7.1 Estrutura Geral

O modelo é composto por **12 tabelas** organizadas em módulos funcionais:

**Gestão de Pessoas:** As tabelas **USUARIOS** e **FUNCIONARIOS** armazenam informações cadastrais, sendo que usuários possuem dados específicos de perfil físico (peso, altura, percentual de massa magra) e objetivos de treino.

**Planos e Pagamentos:** A tabela **PLANOS** define os pacotes oferecidos com seus benefícios em formato JSON, enquanto **PAGAMENTOS** controla o histórico financeiro dos usuários, incluindo status e métodos de pagamento.

**Treinos e Exercícios:** O sistema permite a montagem de treinos personalizados através da relação entre **TREINOS**, **EXERCICIOS** e a tabela associativa **TREINO\_EXERCICIOS**, que define ordem, séries e repetições. A tabela **PONTUACAO** implementa um sistema de gamificação por grupo muscular.

**Controle de Acesso:** As tabelas **RFID\_TAGS** e **REGISTRO\_ENTRADAS** gerenciam a entrada na academia via cartão RFID, código QR ou registro manual, permitindo rastrear acessos permitidos e negados.

**Aulas e Agendamentos:** **AULAS** registra as atividades oferecidas pelos funcionários, enquanto **PARTICIPACOES\_AULA** controla inscrições e presença dos usuários, incluindo avaliações.

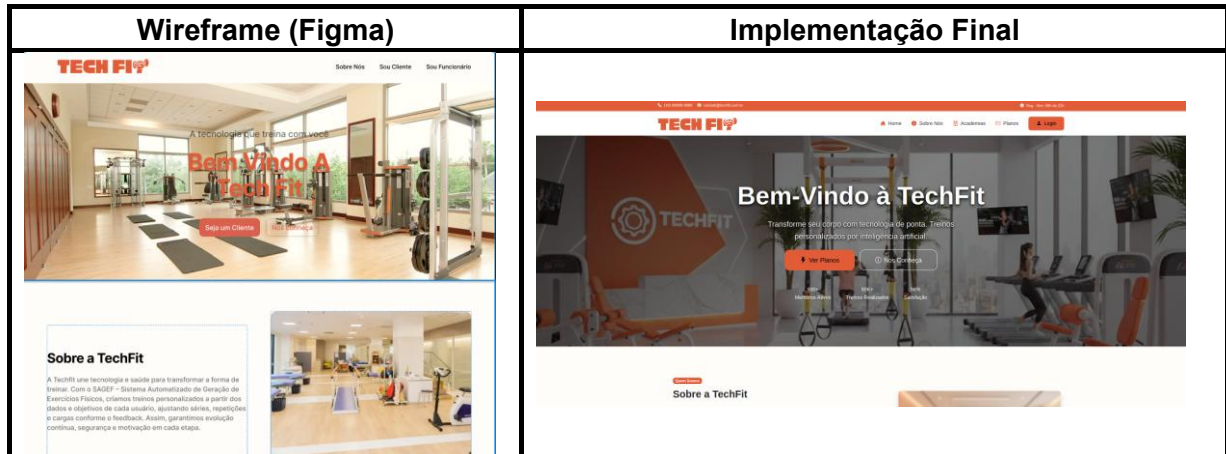
## 7.2 Relacionamentos Principais

O modelo segue uma arquitetura relacional onde USUARIOS é a entidade central, conectando-se a planos, treinos, pagamentos, tags RFID e participações em aulas. Todos os relacionamentos utilizam chaves estrangeiras para garantir integridade referencial.

## 8. WIREFRAME

Esta seção apresenta a evolução das interfaces do sistema TechFit, comparando os wireframes iniciais desenvolvidos no Figma com as telas implementadas.

### 8.1 Página Inicial (Landing Page)

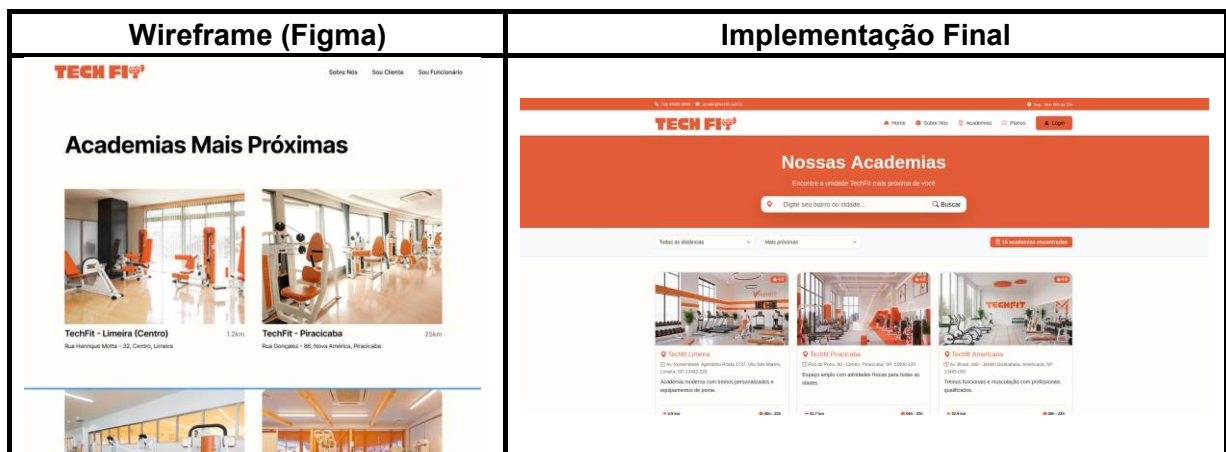


**Descrição:** Página de apresentação do sistema com hero banner, slogan "A tecnologia que treina com você", botões de chamada para ação ("Seja um Cliente" e "Nos conheça") e seção institucional "Sobre a TechFit" explicando o SAGEF.

#### Elementos implementados:

- Header com logo TechFit e navegação (Sobre Nós, Sou Cliente, Sou Funcionário)
- Banner principal com imagem da academia
- Seção sobre o sistema SAGEF
- Cards de unidades

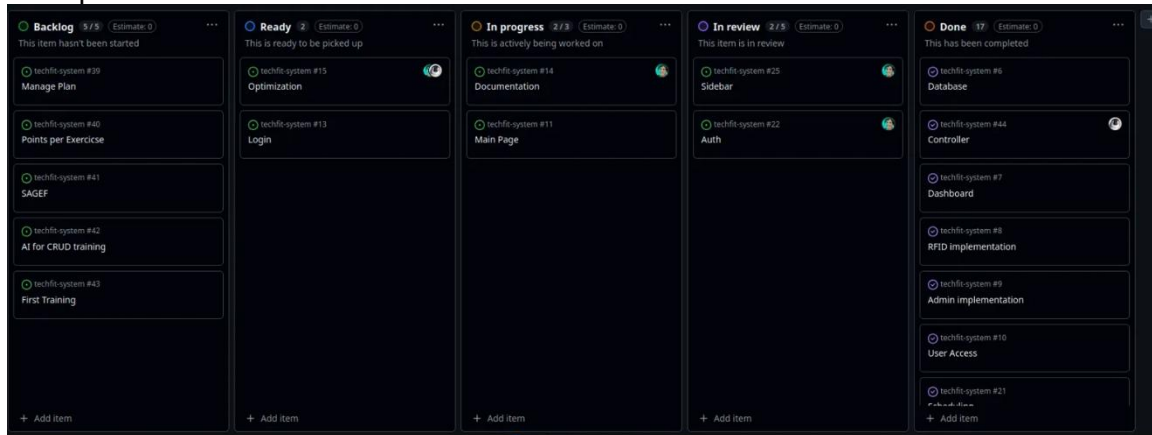
### 8.2 Academias Mais Próximas



**Descrição:** Listagem de unidades da academia em formato de cards com foto, nome da unidade, endereço e distância do usuário.

## 9. METODOLOGIA Scrum

O projeto TechFit adota a metodologia **Scrum** para gerenciamento do desenvolvimento, combinada com um quadro **Kanban** implementado no **GitHub Projects** para visualização e acompanhamento do fluxo de trabalho.



### 9.1 Papéis

- **Product Owner (PO):** Representa os interesses da academia TechFit e dos usuários finais. Responsável por definir e priorizar o Product Backlog, validar os incrementos entregues e garantir o alinhamento estratégico do produto.
- **Scrum Master:** Atua como facilitador do processo Scrum, removendo impedimentos, conduzindo as cerimônias e promovendo a mentalidade ágil na equipe.
- **Equipe de Desenvolvimento:** Profissionais multidisciplinares responsáveis por transformar os itens do backlog em incrementos funcionais, trabalhando de forma colaborativa e auto-organizada.

### 9.2 Quadro Kanban

- O acompanhamento das tarefas é realizado através do GitHub Projects, organizado nas seguintes colunas:

Coluna	Descrição
<b>Backlog</b>	Itens identificados, mas ainda não iniciados
<b>Ready</b>	Tarefas priorizadas e prontas para desenvolvimento
<b>In Progress</b>	Itens em desenvolvimento ativo
<b>In Review</b>	Funcionalidades aguardando revisão de código ou validação
<b>Done</b>	Tarefas concluídas e integradas ao projeto

### 9.3 Sprints

- As Sprints possuem duração fixa de **2 semanas**, ao final das quais são entregues funcionalidades completas e testáveis. O ciclo inclui testes de usabilidade com usuários finais e revisões técnicas com code review em paralelo ao desenvolvimento.

## 9.4 Cerimônias

- **Sprint Planning:** Reunião de planejamento onde a equipe define quais itens do backlog serão desenvolvidos na Sprint.
- **Daily Scrum:** Reuniões diárias de 15 minutos para alinhamento da equipe sobre progresso e impedimentos.
- **Sprint Review:** Apresentação dos incrementos concluídos aos stakeholders para validação e feedback.
- **Sprint Retrospective:** Análise do processo para identificar pontos fortes e oportunidades de melhoria contínua.

## 10. BACKLOG DO PRODUTO

Módulo	Funcionalidades
<b>Dashboard Analytics</b>	Estatísticas em tempo real, gráficos de planos/treinos/grupos musculares, tabelas de acessos e exercícios populares
<b>Controle de Acesso</b>	Integração RFID com Arduino, registro de entradas/saídas, validação de status de pagamento
<b>SAGEF (Treinos)</b>	Cadastro de exercícios, montagem manual de treinos, associação com alunos, sistema de pontuação. (Automação planejada para versões futuras)
<b>Agendamento</b>	Cadastro de aulas em grupo, controle de vagas, inscrições de alunos, agenda integrada
<b>Gestão de Usuários</b>	Cadastro completo de alunos com dados biométricos, planos, status de pagamento
<b>Área do Aluno</b>	Perfil pessoal, visualização de treinos, agenda de aulas, histórico
<b>Pagamentos</b>	Gestão de planos (Starter, Basic, Advanced), controle de status (em dia/atrasado)



## 11. CONCLUSÃO

O TechFit System representa uma solução moderna e integrada para gestão de academias, combinando tecnologias web consolidadas com inovação em IoT para controle de acesso. A arquitetura MVC garante manutenibilidade e escalabilidade, enquanto a metodologia Scrum assegura entregas contínuas de valor.

A plataforma já oferece funcionalidades essenciais como dashboard analítico com gráficos em tempo real, controle de acesso via RFID, gestão completa de treinos e aulas, e áreas dedicadas para funcionários e alunos. O sistema de pontuação por grupos musculares adiciona um elemento de gamificação que incentiva o engajamento dos usuários.

Com a evolução planejada do SAGEF para geração automática de treinos baseada em dados biométricos, o TechFit se posiciona como uma solução completa que une gestão operacional eficiente com experiência diferenciada para o aluno, contribuindo para a competitividade da franquia no mercado fitness.

---

**TechFit System © 2025**

Desenvolvido por João Gabriel González e Henrique Rodrigues Motta  
Escola SENAI "Luiz Varga"