

# Documento de Requisitos do Produto (PRD)

## Sistema de Gestão e Acompanhamento - CT JF (versão atualizada)

Data: 2025-10-20

Responsável: Daniel Nunes Duarte

Matrícula: 20250013550

---

### 1.1. Visão Geral do Produto

#### Nome do produto/sistema

CT JF Manager — Sistema de Gestão e Acompanhamento para Centro de Treinamento Josuelmo Farias

#### Dados do Responsável

Nome: Daniel Nunes Duarte

Matrícula: 20250013550

#### Resumo Executivo

O CT JF Manager é uma plataforma digital desenvolvida para modernizar a gestão do Centro de Treinamento Josuelmo Farias, especializado em Jiu-Jitsu Brasileiro. A plataforma reúne recursos para instrutores, alunos e pais/responsáveis: planejamento de aulas, registro de presença, acompanhamento técnico com histórico de técnicas aprendidas, videoteca e central de campeonatos.

O objetivo é substituir processos manuais (cadernos, planilhas e comunicação fragmentada por redes sociais) por um sistema que dê transparência ao progresso dos alunos e facilite o trabalho do mestre. Para o MVP priorizamos: auto-cadastro dos alunos, registro de aulas e presenças via checklist, atribuição automática de técnicas aprendidas e XP, e uma página pública para pais com botão para solicitar aula experimental via WhatsApp.

#### Objetivos do Produto

- Automatizar o acompanhamento técnico e reduzir trabalho administrativo do mestre.
  - Aumentar engajamento através de progressão (XP) e ranking mensal.
  - Centralizar informações de campeonatos, técnicas e comunicados.
  - Facilitar a captação de alunos com uma página institucional clara.
  - Prover transparência para pais/responsáveis sobre evolução dos alunos.
-

## 1.2. Problema e Oportunidade

### Contexto

Academias de Jiu-Jitsu frequentemente usam métodos analógicos ou comunicação dispersa (WhatsApp/Instagram). Isso gera perda de informação e dificuldade avaliações objetivas para progressão de faixa.

### Problema

1. Falta de rastreabilidade das técnicas aprendidas por aluno.
2. Falta de feedback visual para o aluno sobre seu progresso.
3. Informações sobre campeonatos e eventos espalhadas por múltiplos canais.
4. Pais sem acesso a informações objetivas sobre método, segurança e rotina.
5. Alto esforço manual do mestre em tarefas administrativas.
6. Dificuldade de apresentar a academia de forma profissional a interessados.

### Solução Proposta

Plataforma centralizada com: cadastro (auto) de alunos, gestão de aulas, registro de presenças por checklist, marcação automática de técnicas aprendidas, videoteca com vídeos do professor, dashboards para aluno e mestre, ranking mensal e portal público com botão para aula experimental via WhatsApp.

### Público-Alvo / Personas

- Professor Rafael — Mestre: quer dados objetivos e menos burocracia.
  - Lucas — Atleta: quer acompanhar evolução e oportunidades de competição.
  - Marina — Mãe: busca segurança, metodologia e facilidade para matricular o filho.
  - Amanda — Iniciante adulta: quer aprender fundamentos e ver seu progresso.
-

### 1.3. Escopo do Produto

#### O que está incluído nesta versão (MVP)

1. Autenticação e perfis (auto-cadastro de alunos; cadastro manual pelo mestre/admin como exceção).
2. Carteirinha digital (nome, graduação; QR code OPCIONAL).
3. Gestão de aulas: cadastro pelo mestre, seleção de técnicas e registro de presença via checklist (checkbox).
4. Progressão técnica automática: técnicas marcadas como aprendidas para alunos presentes; XP calculado automaticamente.
5. Videoteca de técnicas com vídeos do YouTube (embed).
6. Dashboards para aluno e mestre (frequência, XP, técnicas aprendidas).
7. Ranking mensal (frequência + XP).
8. Painel de campeonatos com manifestação de interesse.
9. Página pública para pais: Sobre, Equipe, FAQ, Horários e botão/form de aula experimental (redireciona para WhatsApp).
10. Logs de auditoria para ações sensíveis (ex.: mudança de faixa).

#### O que está fora do escopo (por enquanto)

- Scanner/Leitura de QR para check-in (não implementado nesta versão).
  - Upload de fotos de perfil — usar iniciais/avatars gerados.
  - Número de matrícula.
  - Sistema de pagamentos, chat interno, agendamento particular, app nativo, integração com wearables, análise de vídeo por IA.
  - Armazenamento automático de leads do formulário de aula experimental (nesta versão redireciona ao WhatsApp; salvar leads é v2).
-

#### 1.4. Requisitos Funcionais (mín.10; numerados e priorizados)

[Alta] RF01: O sistema deve permitir que novos usuários (alunos) façam auto-cadastro informando nome completo, e-mail, senha, data de nascimento e telefone.

[Alta] RF02: O sistema deve autenticar usuários por e-mail e senha e direcionar para dashboards conforme o tipo de usuário (aluno, mestre, admin).

[Alta] RF03: O sistema deve permitir que alunos informem, no cadastro, graduação (opcional) e categoria (infantil, juvenil, adulto). O mestre/administrador pode cadastrar alunos manualmente quando necessário (fluxo secundário).

[Alta] RF04: O sistema deve permitir que o mestre cadastre aulas realizadas informando data, horário, turma, descrição e técnicas ensinadas (seleção da biblioteca).

[Alta] RF05: O sistema deve permitir que o mestre registre presenças por checklist (checkbox) para cada aula. (Leitura de QR code fica fora do escopo nesta versão.)

[Alta] RF06: Ao confirmar presenças, o sistema deve marcar automaticamente as técnicas da aula como aprendidas no histórico dos alunos presentes.

[Alta] RF07: O sistema deve calcular e atribuir XP automaticamente: presença = 10 XP; técnica nova aprendida = 5 XP (por técnica).

[Alta] RF08: O sistema deve atualizar o XP\_total do aluno automaticamente; não permitir edição manual do XP via interface.

[Alta] RF09: O sistema deve permitir que o mestre altere manualmente a graduação do aluno; ao confirmar, o sistema deve resetar o XP\_total do aluno para 0 e gravar um registro de auditoria (quem, data/hora, graduação anterior e nova).

[Média] RF10: O sistema deve permitir que usuários editem campos do perfil (telefone, endereço e outros não sensíveis); o e-mail não deve ser editável via interface.

[Média] RF11: O sistema deve gerar uma carteirinha digital para cada aluno contendo nome, graduação e identificação (iniciais/ID). A inclusão de QR code fica opcional.

[Média] RF12: O sistema deve exibir uma videoteca categorizada com vídeos embed do YouTube e permitir busca e filtros.

[Média] RF13: O sistema deve indicar visualmente quais técnicas cada aluno já aprendeu ao navegar na videoteca.

[Média] RF14: O sistema deve exibir dashboard do aluno com total de presenças, XP\_total, técnicas aprendidas, gráfico de frequência e percentual até a próxima graduação.

[Média] RF15: O sistema deve exibir dashboard do mestre com frequência média, lista de alunos próximos à graduação ( $>80\%$  do XP necessário) e alunos com baixa frequência ( $<40\%$  no mês).

[Alta] RF16: O sistema deve permitir cadastro e gerenciamento de campeonatos (nome, data, local, link de inscrição, categorias e prazo).

[Média] RF17: O sistema deve permitir que alunos manifestem interesse em campeonatos; o mestre deve ver a lista de interessados.

[Baixa] RF18: O sistema deve calcular mensalmente um ranking por categoria (frequência  $60\% + \text{XP } 40\%$ ) e manter histórico de rankings.

[Baixa] RF19: O sistema deve permitir que o mestre edite/exclua uma aula em até 48 horas após sua criação.

[Baixa] RF20: O sistema deve fornecer na página pública um formulário simples para aula experimental que, ao submeter, redirecione o visitante para o WhatsApp do CT com mensagem pré-preenchida (não cria automaticamente usuário nem salva lead nesta versão).

---

## 1.5. Requisitos Não-Funcionais (mín.5 categorizados)

### Usabilidade

RNF01: Interface intuitiva, orientada a usuários com baixa familiaridade tecnológica; rotas principais acessíveis em até 3 cliques.

RNF02: Feedback visual imediato (toasts, banners, validações em tempo real) em português.

### Performance

RNF03: Páginas principais (dashboard, videoteca, ranking) devem carregar em menos de 3 segundos em conexão banda larga (10 Mbps).

RNF04: Suportar até 200 usuários simultâneos sem degradação perceptível.

### Segurança

RNF05: Senhas armazenadas com hash bcrypt (salt suficientemente forte, p.ex. 10 rounds ou superior).

RNF06: Proteções contra CSRF e validação de entrada no servidor (Zod/TypeBox ou equivalente).

RNF07: Controle de acesso por roles (RBAC) — impedir que alunos acessem funcionalidades de mestre/admin.

RNF08: Logs de auditoria para ações críticas (mudança de graduação, exclusão de aulas, alterações de XP).

### **Compatibilidade**

RNF09: Compatível com Chrome, Firefox, Safari e Edge (últimas 2 versões).

RNF10: Suporte a tecnologias assistivas (WCAG 2.1 AA objetivo).

### **Responsividade**

RNF11: Totalmente funcional em dispositivos mobile (smartphones e tablets); elementos interativos com alvo de toque mínimo 44x44px.

### **Manutenibilidade**

RNF12: Código organizado em módulos, com documentação mínima da API e testes unitários para regras de negócio críticas (XP, reset de XP, registro de presença).

### **Disponibilidade**

RNF13: Disponibilidade mínima 99% no horário comercial (6h–23h); backups diários automáticos do banco de dados.

---

## **1.6. Regras de Negócio (mín.5)**

RN01: Um usuário só pode ter uma conta por e-mail. Cadastros duplicados são rejeitados.

RN02: Senhas mínimas de 8 caracteres, com pelo menos uma letra e um número.

RN03: Apenas “Mestre” ou “Administrador” podem cadastrar aulas, técnicas e campeonatos. O cadastro de alunos é preferencialmente via auto-cadastro; mestre/admin pode cadastrar como exceção.

RN04: Uma técnica só pode ser marcada como aprendida para um aluno se o aluno esteve presente em, pelo menos, uma aula onde ela foi ensinada.

RN05: XP é calculado automaticamente (presença + técnicas novas) e não pode ser editado manualmente pela interface.

RN06: O registro de presença só pode ser feito no dia da aula ou até 24 horas após sua data.

RN07: A mudança de graduação é manual: quando o mestre altera a graduação de um aluno, o sistema reseta o XP\_total do aluno para 0 e grava um registro de auditoria com os detalhes da operação.

RN08: A carteirinha digital é gerada no cadastro do aluno; o QR code é opcional e não é utilizado para check-in nesta versão.

RN09: O formulário de aula experimental na página pública redireciona o visitante ao WhatsApp do CT com mensagem pré-preenchida; o site não cria automaticamente um usuário nem salva necessariamente o lead nesta versão.

RN10: Aulas não devem ser cadastradas para datas futuras segundo a política adotada; a configuração pode ser alterada por necessidade.

---

## 1.7. Casos de Uso (4–6 principais)

### UC01 - Auto-Cadastro do Aluno

- Ator principal: Visitante / Aluno
- Pré-condições: nenhuma
- Fluxo principal:
  1. Visitante acessa /cadastro
  2. Preenche nome, e-mail, senha, data de nascimento, telefone, (opcional) graduação e categoria
  3. Submete; validação; conta criada
  4. Sistema gera carteirinha digital (QR opcional) e redireciona ao dashboard do aluno
- Pós-condições: Conta ativa; carteirinha disponível

### UC02 - Cadastrar Aula e Registrar Presenças

- Ator principal: Mestre
- Pré-condições: Mestre autenticado; alunos cadastrados; técnicas na biblioteca
- Fluxo principal:
  1. Mestre cadastra aula (data, horário, turma, descrição, técnicas)
  2. Sistema salva aula e abre registro de presenças
  3. Mestre marca checkboxes dos alunos presentes

4. Mestre confirma; sistema grava presenças, marca técnicas aprendidas e atribui XP
- Pós-condições: Presenças e técnicas aprendidas registradas; XP\_total atualizado

#### **UC03 - Alterar Graduação (Mudança de Faixa)**

- Ator principal: Mestre
- Pré-condições: Mestre autenticado; aluno com XP acumulado
- Fluxo principal:
  1. Mestre acessa perfil do aluno e seleciona “Alterar Graduação”
  2. Seleciona nova graduação e confirma
  3. Sistema grava alteração, reseta XP\_total para 0 e gera log de auditoria (quem, quando, graduação anterior/nova)
- Pós-condições: Graduação atualizada; XP resetado; auditoria registrada

#### **UC04 - Solicitar Aula Experimental (Público → WhatsApp)**

- Ator principal: Visitante / Pai
- Pré-condições: Página pública disponível
- Fluxo principal:
  1. Visitante clica em “Aula Experimental”
  2. Preenche formulário rápido (nome, telefone, horário preferido, mensagem)
  3. Ao submeter, site abre WhatsApp (wa.me/) com mensagem pré-preenchida contendo os dados
  4. Sistema exibe confirmação local: “Você será redirecionado ao WhatsApp”
- Pós-condições: Contato via WhatsApp; (opcional) lead salvo em v2

#### **UC05 - Consultar Videoteca e Estudar Técnica**

- Ator principal: Aluno



- Pré-condições: Autenticado; técnicas cadastradas
  - Fluxo principal:
    1. Acessa /videoteca, filtra por categoria/graduação
    2. Vê lista com marcação das técnicas já aprendidas
    3. Clica em técnica → abre vídeo embed + detalhes
  - Pós-condições: Aluno revisa conteúdo; views podem ser registradas para analytics (v2)
- 

## 1.8. Modelagem de Dados (Conceitual)

### Entidades Principais (resumo)

- **Usuario:** id (PK), nome, email (unique), senha\_hash, telefone, data\_nascimento, tipo\_usuario (aluno/mestre/admin), ativo, data\_cadastro, data\_ultima\_alteracao.
- **Aluno** (estende Usuario): usuario\_id (FK), id\_unico (UUID — para QR code; OPCIONAL/nullable), graduacao\_atual (enum), categoria (enum), xp\_total, data\_inicio, responsavel\_nome?, responsavel\_telefone?, created\_at.
- **Tecnica:** id, nome, descricao, categoria, graduacao\_minima, video\_youtube\_url, video\_youtube\_id, thumbnail\_url, criada\_por\_usuario\_id, data\_cadastro.
- **Aula:** id, data\_aula, horario\_inicio, horario\_fim?, turma, descricao, cadastrada\_por\_usuario\_id, data\_cadastro, data\_alteracao.
- **Aula\_Tecnica:** aula\_id, tecnica\_id (PK composto).
- **Presenca:** id, aula\_id, aluno\_id, data\_registro, xp\_atribuido — unique(aula\_id, aluno\_id).
- **Aluno\_Tecnica\_Aprendida:** id, aluno\_id, tecnica\_id, aula\_id?, data\_aprendizado — unique(aluno\_id, tecnica\_id).
- **Campeonato:** id, nome, data\_campeonato, local, descricao, categorias\_disponiveis (JSON), link\_inscricao, prazo\_inscricao, cadastrado\_por\_usuario\_id, data\_cadastro.
- **Campeonato\_Interesse:** id, campeonato\_id, aluno\_id, categoria\_pretendida, peso, idade\_categoria, data\_manifestacao —

unique(campeonato\_id, aluno\_id).

- **Ranking\_Mensal:** id, mes\_referencia (date), aluno\_id, presencas\_mes, xp\_mes, pontuacao\_total, posicao — unique(mes\_referencia, aluno\_id).
- **Solicitacao\_Aula\_Experimental** (opcional v2): id, nome, email, telefone, idade\_interessado, horario\_interesse, mensagem, status, data\_solicitacao, data\_ultimo\_contato.
- **Log\_Auditoria:** id, usuario\_id, acao, entidade\_tipo, entidade\_id, dados\_anteriores (JSON), dados\_novos (JSON), ip\_origem, data\_hora.

### Relacionamentos Principais

1. Usuario 1:1 Aluno (quando tipo\_usuario = aluno).
2. Usuario 1:N Aula (mestre cadastra aulas).
3. Usuario 1:N Tecnica (mestre cadastra técnicas).
4. Aula N:N Tecnica (via Aula\_Tecnica).
5. Aula 1:N Presenca; Aluno 1:N Presenca.
6. Aluno N:N Tecnica (via Aluno\_Tecnica\_Aprendida).
7. Campeonato 1:N Campeonato\_Interesse; Aluno 1:N Campeonato\_Interesse.
8. Aluno 1:N Ranking\_Mensal.

### Diagrama ER (simplificado, textual)

[Usuario] 1--1 [Aluno]  
[Usuario] 1--\* [Aula]  
[Usuario] 1--\* [Tecnica]  
[Usuario] 1--\* [Campeonato]  
[Aula] \*--\* [Tecnica] (via Aula\_Tecnica)  
[Aula] 1--\* [Presenca]  
[Aluno] 1--\* [Presenca]  
[Aluno] \*--\* [Tecnica] (via Aluno\_Tecnica\_Aprendida)  
[Aula] 1--\* [Aluno\_Tecnica\_Aprendida]  
[Campeonato] 1--\* [Campeonato\_Interesse]  
[Aluno] 1--\* [Campeonato\_Interesse]  
[Aluno] 1--\* [Ranking\_Mensal]

---

## 1.9. User Stories

- Como visitante, quero me cadastrar como aluno, para que eu possa acessar minha área e acompanhar meu progresso.
  - Como aluno, quero acessar a videoteca de técnicas, para revisar aulas fora do tatame.
  - Como aluno, quero ver minha posição no ranking mensal, para me motivar a treinar mais.
  - Como mestre, quero cadastrar aulas e registrar presenças rapidamente, para reduzir o trabalho administrativo.
  - Como mestre, quero alterar a graduação de um aluno e que o sistema resete o XP automaticamente, para manter coerência da progressão.
  - Como pai, quero solicitar uma aula experimental via site (redirecionamento para WhatsApp), para falar com a academia rapidamente.
  - Como administrador, quero logs de auditoria das operações críticas, para rastreabilidade.
- 

## 1.10. Wireframes / Fluxo de Navegação (descrição)

### Estrutura de Navegação Principal

- Público: Início → Sobre → Equipe → FAQ → Horários → Aula Experimental (redireciona para WhatsApp) → Login/Cadastro.
- Aluno (após login): Dashboard → Meu Progresso → Videoteca → Ranking → Campeonatos → Carteirinha → Perfil.
- Mestre: Dashboard Mestre → Gestão de Aulas → Registrar Presenças → Biblioteca de Técnicas → Gestão de Campeonatos → Gestão de Alunos.
- Admin: Todas as funcionalidades do Mestre + Gestão de Usuários, Logs e Configurações.

### Fluxos Principais (resumo)

- Auto-cadastro → criação de conta e carteirinha (QR opcional) → dashboard.
- Mestre cadastra aula → registra presenças via checklist → sistema marca

técnicas aprendidas e atribui XP.

- Alteração de graduação pelo mestre → reset de XP e audit log.
  - Formulário público de aula experimental → redirecionamento ao WhatsApp do CT com mensagem pré-preenchida.
- 

### **Critérios de Aceitação (MVP)**

- Alunos conseguem auto-cadastrar e autenticar; carteirinha gerada (QR opcional).
  - Mestre cadastra aulas e registra presenças por checklist; técnicas são marcadas como aprendidas.
  - XP\_total é calculado e atualizado automaticamente; não editável via UI.
  - Mestre altera graduação e XP é resetado com log de auditoria.
  - Página pública permite solicitar aula experimental e redireciona ao WhatsApp (sem criar usuário automaticamente).
  - Videoteca reproduz vídeos via embed; técnicas aprendidas marcadas visualmente.
  - Dashboards básicos (aluno e mestre) apresentam métricas essenciais.
-