

# **Documento Base do MVP – Sistema de Gestão de Autoescolas**

## **1. Visão geral**

Este documento define os requisitos iniciais do MVP de um sistema SaaS para gestão de auto escolas, com foco nos processos essenciais de alunos, aulas práticas e vendas. O objetivo é entregar rapidamente valor ao negócio, garantindo aderência às regras legais e base sólida para evolução futura.

---

## **2. Objetivos do MVP**

- Centralizar o cadastro e acompanhamento de alunos.
  - Registrar e controlar aulas práticas, garantindo carga horária mínima.
  - Gerenciar vendas de pacotes de aulas e situação financeira básica.
  - Suportar múltiplas autoescolas.
- 

## **3. Escopo do MVP**

### **3.1. Dentro do Escopo**

- Cadastro e gestão de auto escolas.
- Cadastro de usuários (Administrador e Instrutor).
- Cadastro e gestão de alunos.
- Criação de vendas (pacotes de aulas).
- Agendamento e confirmação de aulas práticas.
- Cálculo automático de carga horária realizada.
- Controle básico de status do aluno (Em formação / Pronto para exame).

### **3.2. Fora do Escopo (MVP Atual)**

- Integração com o sistemas governamentais (ex.: Detran/Renach).
  - Exames teóricos e práticos.
  - Aplicativo mobile.
  - Relatórios avançados.
  - Criação de documentos/relatórios.
  - Orquestração de containers com Kubernetes.
-

## **4. Atores do Sistema**

### **4.1. Administrador da Autoescola**

- Cadastra e gerencia alunos.
- Cria vendas e define pacotes.
- Agenda aulas práticas.
- Visualiza o Progresso do aluno.

### **4.2. Instrutor**

- Visualiza a agenda.
  - Confirma a realização da aula.
- 

## **5. Fluxo Principal do Sistema**

- 5.1. Administrador cadastra uma auto escola.
  - 5.2. Administrador cria um usuário instrutor.
  - 5.3. Administrador cadastrou um aluno.
  - 5.4. Administrador cria uma venda para o aluno.
  - 5.5. Administrador agenda aulas práticas vinculando o aluno e o instrutor.
  - 5.6. Instrutor realiza e confirma a aula.
  - 5.7. Sistema soma automaticamente a carga horária do aluno.
  - 5.8. Ao atingir a carga mínima o aluno passa para o status Pronto para exame.
- 

## **6. Histórias de Usuário**

### **6.1. Cadastro e Gestão da Autoescola**

- US-01: Como Administrador, quero cadastrar minha autoescola para utilizar o sistema de forma independente de outras autoescolas.

#### **Critério de Aceite:**

- A autoescola deve possuir CNPJ e nome .
- Os dados devem ficar isolados das demais auto escolas.

### **6.2. Gestão de Usuários**

- US-02: Como Administrador, quero cadastrar instrutores no sistema para vinculá-los às aulas práticas quando necessário.

#### **Critério de Aceite:**

- O instrutor possui acesso apenas às aulas em que está designado.
- O Instrutor não possui acesso a dados financeiros ou administrativos.

### **6.3. Gestão de Alunos**

- US-03: Como Administrador, quero cadastrar um aluno para acompanhar seu processo de formação.

#### **Critério de Aceite:**

- Aluno deve estar vinculado a uma autoescola.
- Deve ser possível visualizar status e carga horária.

### **6.4. Gestão de Vendas**

- US-04: Como Administrador, quero criar uma venda de pacote de aulas para registrar o acordo comercial com o aluno.

#### **Critério de Aceite:**

- A venda deve conter quantidade de aulas e valor total.
- A venda deve estar vinculada a um aluno.

### **6.5. Agendamentos de Aulas**

- US-05: Como Administrador, quero agendar aulas práticas para organizar a formação do aluno.

#### **Critério de Aceite:**

- A aula deve possuir data, hora, aluno e instrutor.
- A aula não pode ser agendada sem venda ativa.

### **6.6. Execução das Aulas**

- US-06: Como Instrutor, quero confirmar a realização da aula para validar a carga horária do aluno.

#### **Critério de Aceite:**

- A aula só pode ser confirmada uma única vez.
- Confirmação soma carga horária automaticamente.

### **6.7. Controle de Progresso**

- US-07: Como Administrador, quero visualizar o progresso do aluno para saber quando ele está pronto para exame.

#### **Critério de Aceite:**

- O sistema deve mostrar carga prevista, realizada e restante.
  - O status deve ser atualizado automaticamente.
-

## **7. Requisitos Funcionais**

### **7.1. Alunos**

- RF-01: Cadastrar aluno vinculado a uma autoescola.
- RF-02: Editar dados do aluno.
- RF-03: Visualizar carga horária prevista, realizada e restante.
- RF-04: Atualizar status do aluno automaticamente conforme carga horária.

### **7.2. Vendas**

- RF-05: Criar venda para um aluno.
- RF-06: Definir pacote com quantidade de aulas e valor total.
- RF-07: Associar venda às aulas do aluno.

### **7.3. Aulas Práticas**

- RF-08: Agendar aula prática.
  - RF-09: Vincular aula a aluno e instrutor.
  - RF-10: Confirmar a realização da aula.
- 

## **8. Requisitos Não Funcionais**

- 8.1. RNF-01: Sistema multi-tenant com isolamento por autoescola.
  - 8.2. RNF-02: Autenticação e autorização por perfil.
  - 8.3. RNF-03: API RESTful.
  - 8.4. RNF-04: Persistência em banco relacional.
  - 8.5. RNF-05: Código organizado em camadas.
- 

## **9. Premissas e Restrições**

- 9.1. O sistema será desenvolvido inicialmente como monólito.
  - 9.2. Backend em Java com Spring Boot.
  - 9.3. Frontend em Angular.
  - 9.4. Banco de dados relacional (PostgreSQL).
- 

## **10. Premissas e Restrições**

- 10.1. Integração com órgãos oficiais.
- 10.2. Relatórios gerenciais.
- 10.3. Gestão financeira avançada.
- 10.4. Escalabilidade com Kubernetes.