# **Documento de Requisitos – Sistema de Gestão de Alunos (Autoescola)**

## **1. Objetivo**

Criar uma **aplicação web em JavaScript** para gestão de alunos de uma autoescola.

A aplicação deve permitir:

**Cadastrar Usuário**

**Cadastrar cliente**

**Atualizar cliente**

**Deletar cliente**

**Anexar documentos do cliente**

## **2. Contexto**

A aplicação será utilizada **a somente para exercícios de testes de software**.

Não terá uso em produção.

Não será utilizado banco de dados; as informações deverão ser **armazenadas em memória (variáveis ou constantes)**.

## **3. Funcionalidades**

### **3.1 Sistema de Autenticação**

* **Tela de login** com validação de usuário e senha
* **Usuário padrão**: Admin / Admin
* **Gestão de usuários** com diferentes tipos (admin/usuário)
* **Controle de acesso** baseado em permissões
* **Criar novos usuários** com diferentes tipos
* **Ativar/Desativar usuários** do sistema
* **Alterar senhas** de forma segura
* **Excluir usuários** (exceto administradores)

### **3.2 Cadastro de Alunos**

* Nome completo
* CPF (com validação)
* Telefone (com máscara)
* Data de nascimento
* Categoria da CNH (A, B, AB, C, D, E)
* Endereço
* Observações
* Status (ativo/inativo)
* Data de cadastro

### **3.3 Atualização de Cliente**

Permitir alterar dados já cadastrados.

O cliente deverá ser identificado pelo CPF.

### **3.4 Exclusão de Cliente**

Remover cadastro de cliente da lista.

Confirmar exclusão antes da remoção.

### **3.5 Anexar Documentos**

* **Anexar documentos** para cada aluno
* **Visualizar lista** de documentos anexados
* **Remover documentos** quando necessário
* Suporte a múltiplos formatos (PDF, imagens, documentos)

### **3.5 Gestão de Alunos**

* **Cadastrar novo aluno** com dados completos
* **Editar informações** de alunos existentes
* **Excluir alunos** com confirmação
* **Visualizar detalhes** completos de cada aluno
* **Busca e filtros** por nome, CPF ou telefone

## **4. Regras**

Toda a **estrutura de arquivos** da aplicação deverá ser construída (separação em pastas de componentes, serviços e modelos).

Todas as **bibliotecas necessárias** deverão ser adicionadas.

## **6. Requisitos Funcionais (RF)**

**RF001** – Permitir cadastrar cliente.

**RF002** – Permitir atualizar cliente pelo CPF.

**RF003** – Permitir excluir cliente.

**RF004** – Permitir anexar documentos ao cliente.

**RF005** – Listar clientes cadastrados.

## **7. Requisitos Não Funcionais (RNF)**

**RNF001** – A aplicação deve ser construída em **JavaScript puro**.

**RNF002** – Os dados devem ser armazenados apenas em memória (arrays/objetos).

**RNF003** – A interface deve ser responsiva e simples.

**RNF004** – O código deve ser organizado em módulos separados (model, service, components).