

Requisitos Funcionais e Não Funcionais - SaaS Car Platform

Este documento define os **requisitos funcionais (RF)** e **não funcionais (RNF)** da plataforma. Ele serve como um **documento vivo** para guiar o desenvolvimento, priorização, testes e a evolução do sistema.

1. Requisitos Funcionais (RF)

Os requisitos funcionais descrevem **o que o sistema deve fazer** sob a perspectiva de negócio e do usuário.

RF-01 – Autenticação

Descrição: O sistema deve fornecer autenticação segura para todos os usuários.

- Registro de usuário (signup).
- Login de usuário.
- Logout de usuário.
- Autenticação baseada em **JWT**.
- Suporte para **refresh tokens**.
- Revogação de token no logout.

Atores: Cliente, Oficina.

RF-02 – Perfis de Usuário

Descrição: O sistema deve suportar diferentes papéis de usuário com permissões distintas.

- Papel de usuário **CLIENT**.
- Papel de usuário **WORKSHOP**.
- Controle de acesso baseado em funções (RBAC).
- Autorização aplicada no nível da API.

RF-03 – Gestão de Veículos

Descrição: Clientes devem ser capazes de gerenciar seus veículos.

- Criar veículo.
- Listar veículos.
- Atualizar veículo.
- Excluir veículo.
- Veículos pertencem a um único cliente.
- Oficinas não podem gerenciar veículos de terceiros diretamente.

Atores: Cliente.

RF-04 – Gestão da Oficina

Descrição: Oficinas devem gerenciar seu perfil público e catálogo de serviços.

- Criar perfil da oficina.
- Atualizar informações da oficina.
- Registrar serviços oferecidos.
- Definir horários de funcionamento / disponibilidade.
- Perfil da oficina vinculado 1:1 com um usuário.

Atores: Oficina.

RF-05 – Busca de Oficinas

Descrição: Clientes devem ser capazes de descobrir oficinas.

- Busca por localização geográfica.
- Ordenação por proximidade.
- Filtro por serviços oferecidos.
- Visualização de resumo nos resultados de busca.

Atores: Cliente.

RF-06 – Agendamento

Descrição: Gestão do ciclo de vida de agendamentos.

- Cliente cria solicitação de agendamento.
- Oficina aceita ou rejeita solicitações.
- Status do ciclo de vida: **PENDING**, **CONFIRMED**, **CANCELLED**, **DONE**.
- Apenas agendamentos **CONFIRMED** podem ser alterados para **DONE**.

Atores: Cliente, Oficina.

RF-07 – Avaliações e Ratings

Descrição: Clientes avaliam oficinas após a conclusão do serviço.

- Avaliação permitida apenas após status **DONE**.
- Uma avaliação por serviço concluído.
- Escala de 1 a 5 estrelas.
- Comentário opcional.
- Atualização automática da média da oficina.

Atores: Cliente.

RF-08 – Mensageria

Descrição: Comunicação direta entre as partes.

- Mensagens em tempo real via **WebSocket**.
- Permitido apenas se houver agendamento ou solicitação ativa.
- Restrito aos envolvidos no serviço.
- Histórico básico de mensagens.

Atores: Cliente, Oficina.

2. Requisitos Não Funcionais (RNF)

Os requisitos não funcionais definem os atributos de qualidade e restrições técnicas.

Segurança

- Senhas criptografadas com **bcrypt**.
- JWT com tempo de expiração curto.
- Armazenamento seguro de refresh tokens.
- **Rate limiting** em endpoints de autenticação.
- Proteção contra **IDOR** (Insecure Direct Object Reference).
- Validação de ownership em todos os recursos.

Performance

- Consultas geoespaciais otimizadas com **PostGIS**.
- Indexação de campos críticos no banco de dados.
- Paginação obrigatória em todos os endpoints de listagem.
- Camada de cache (**Redis**) para buscas e dados de leitura intensiva.

Escalabilidade

- Serviços de backend **stateless**.
- Suporte a escalonamento horizontal.
- Proxy reverso e balanceamento de carga (Nginx).
- Arquitetura baseada em containers independentes.

Qualidade e Testes

- Cobertura mínima de **80%** em serviços críticos.
- Testes de Unidade (Domínio e Services).
- Testes de Integração (API + DB).
- Execução automatizada via pipeline de CI.
- Linting e formatação de código padronizados.

Observabilidade

- Logs estruturados em formato **JSON**.
- Propagação de **Correlation ID**.
- Tratamento de erros centralizado.
- Endpoint de saúde: **GET /health**.

3. Diretrizes de Evolução

- Este documento deve ser atualizado em caso de novas features ou mudanças arquiteturais.
- Requisitos devem ser rastreáveis para endpoints, testes e user stories.

4. Status

- **Status do Documento:** Ativo
- **Escopo:** MVP e evolução inicial
- **Versão:** 1.0.0