

# Documento de Requisitos do Sistema — Plataforma IoT Multi-Dispositivo

## 1. Introdução

### 1.1 Objetivo

O objetivo deste sistema é prover uma **plataforma web unificada** para gerenciamento de **dispositivos IoT** instalados em diferentes lugares de clientes. A plataforma deve permitir:

- Administração de **usuários e clientes**.
  - Cadastro de **lugares** (onde os dispositivos estão instalados).
  - Cadastro e monitoramento de **dispositivos de diferentes tipos** (umidificadores, sensores de umidade do solo, temperatura, iluminação, etc).
  - Coleta e visualização de **leituras históricas** enviadas pelos dispositivos.
  - Configuração remota de parâmetros de cada dispositivo.
- 

### 1.2 Escopo

O sistema será composto por duas áreas principais:

#### Administrador

- Gestão de usuários e permissões.
- Gestão de lugares e dispositivos vinculados.
- Dashboard global com estatísticas por cliente e por tipo de dispositivo.
- Cadastro de **novos tipos de dispositivos IoT** suportados.

#### Cliente

- Acesso exclusivo ao seu ambiente.
  - Gestão dos lugares vinculados à sua conta.
  - Cadastro e configuração de dispositivos IoT em cada lugar.
  - Visualização das leituras em tempo real e histórico.
  - Configuração de parâmetros específicos de cada tipo de dispositivo.
-

## 2. Requisitos Funcionais

### 2.1 Autenticação

- RF01: Login seguro com autenticação via JWT.
- RF02: Controle de permissões (Administrador e Cliente).

### 2.2 Usuários

- RF03: Administrador pode cadastrar, editar, ativar/desativar usuários.
- RF04: Cliente pode gerenciar apenas sua própria conta.

### 2.3 Lugares

- RF05: Usuário pode cadastrar, editar, desativar lugares (soft delete).
- RF06: Lugares contêm dados de endereço e status ativo/inativo.

### 2.4 Dispositivos

- RF07: Usuário pode cadastrar dispositivos em lugares vinculados.
- RF08: Cada dispositivo deve ter:
  - Nome
  - Tipo (ex: umidificador\_ar, sensor\_solo, temperatura, luz)
  - Configuração (armazenada em JSON)
  - Status (online/offline)
- RF09: Usuário pode ativar/desativar dispositivos (soft delete).
- RF10: Administrador pode cadastrar **novos tipos de dispositivos suportados**.

### 2.5 Leituras

- RF11: Cada dispositivo envia leituras periódicas via API/MQTT.
- RF12: Leituras ficam armazenadas com:
  - ID do dispositivo
  - Timestamp
  - Dados (JSON flexível)
- RF13: Usuário pode filtrar leituras por período de tempo.

- RF14: Usuário pode visualizar gráficos e relatórios exportáveis (PDF/Excel).

## 2.6 Dashboard

- RF15: Administrador vê dashboard global:
  - Usuários ativos
  - Lugares ativos
  - Dispositivos online/offline por tipo
  - Estatísticas gerais
- RF16: Cliente vê dashboard individual:
  - Lugares e dispositivos vinculados
  - Status em tempo real
  - Últimas leituras por dispositivo

---

## 3. Requisitos Não Funcionais

- RNF01: Arquitetura **frontend + backend separados**.
  - RNF02: Banco de dados PostgreSQL (Neon).
  - RNF03: Autenticação e autorização via JWT.
  - RNF04: APIs REST desenvolvidas em **FastAPI**.
  - RNF05: Frontend em **React + Vite**.
  - RNF06: Armazenamento de leituras e configurações em formato JSON (flexibilidade).
  - RNF07: Suporte a múltiplos clientes (multi-tenant).
  - RNF08: Containerização com Docker.
  - RNF09: Comunicação com dispositivos via HTTP/MQTT.
-

## 4. Modelo Conceitual

### 4.1 Entidades Principais

- **Usuário**
    - id, nome, email, senha\_hash, role (admin/cliente), ativo
  - **Lugar**
    - id, nome, endereço (cep, rua, número, bairro, cidade, estado, complemento), ativo, usuario\_id
  - **Dispositivo**
    - id, nome, tipo, lugar\_id, config (JSON), status (online/offline), ativo
  - **Leitura**
    - id, dispositivo\_id, timestamp, dados (JSON)
  - **TipoDispositivo**
    - id, nome, descricao, parametros\_config (JSON Schema)
- 

## 5. Casos de Uso (exemplos resumidos)

- **UC01 – Autenticação:** Usuário realiza login com email/senha → sistema gera JWT.
  - **UC02 – Gerenciar Lugares:** Usuário cria ou desativa lugar → vinculado à sua conta.
  - **UC03 – Cadastrar Dispositivo:** Usuário escolhe um lugar, informa nome, tipo e parâmetros.
  - **UC04 – Receber Leituras:** Dispositivo envia dados → sistema armazena em leituras.
  - **UC05 – Visualizar Dashboard:** Cliente acessa painel → sistema exibe dispositivos e leituras.
  - **UC06 – Administrar Tipos de Dispositivo:** Admin cadastra novo tipo com parâmetros de configuração.
-

## **6. Futuras Extensões**

- Integração com alertas por email/WhatsApp em caso de falhas.
- Integração com IA para análise preditiva das leituras.