

# Documento de Requisitos INFOVAX

## 1. Visão Geral do Projeto

### 1.1. Objetivo do Documento

Este documento descreve os requisitos necessários para o desenvolvimento do sistema InfoVax, que tem como objetivo controlar o uso de refrigeradores em ambientes que armazenam vacinas ou materiais médicos que não podem ficar em temperatura ambiente.

### 1.2. Escopo do Projeto

O projeto InfoVax envolverá a implementação de um sistema inteligente que monitora, quantifica e alerta o usuário sobre temperatura e registro de utilização (quem utilizou, quando utilizou). O sistema atuará como uma plataforma de consumo de dados do sistema SafeVax, permitindo uma visualização centralizada e amigável das informações enviadas pelos refrigeradores. O sistema deverá ser implementado até dezembro de 2024 e está previsto para ser utilizado em hospitais, postos de saúde e empresas privadas da área da saúde.

## 2. Requisitos Funcionais

**RF001:** Consumir dados de sensores do sistema SafeVax.

- **Importância:** Essencial.
- **Descrição:** O sistema deve receber, processar e exibir dados enviados pelo sistema SafeVax, incluindo temperaturas, registros de acessos e alertas de uso inadequado.

**RF002:** Exibir temperatura em tempo real.

- **Importância:** Essencial.
- **Descrição:** A plataforma deverá exibir a temperatura de cada refrigerador conectado em tempo real, com histórico gráfico acessível.

**RF003:** Notificar o usuário sobre alertas críticos.

- **Importância:** Essencial.
- **Descrição:** Emitir notificações (push) no aplicativo em caso de temperaturas fora do limite ou utilizações irregulares.

**RF004:** Gerenciar usuários e permissões.

- **Importância:** Importante.

- **Descrição:** Permitir o cadastro de usuários e a definição de permissões, controlando quem pode acessar os dados de cada refrigerador.

**RF005:** Gerar relatórios sobre uso e condições do refrigerador.

- **Importância:** Importante.
- **Descrição:** Disponibilizar relatórios periódicos com informações consolidadas sobre temperatura, alertas e registros de uso.

### 3. Requisitos Não Funcionais

**RNF001:** Garantir compatibilidade com o sistema SafeVax.

- **Importância:** Essencial.
- **Descrição:** O InfoVax deve ser compatível com as APIs e protocolos de comunicação definidos pelo SafeVax.

**RNF002:** Interface responsiva e intuitiva.

- **Importância:** Importante.
- **Descrição:** O sistema deve ser desenvolvido com design responsivo, adaptável a dispositivos móveis e desktops, com usabilidade priorizada.

**RNF003:** Tempo de resposta rápido.

- **Importância:** Importante.
- **Descrição:** O sistema deve apresentar baixo tempo de latência no processamento e exibição de dados.

### 4. Requisitos Técnicos

#### 4.1. Plataforma

O sistema será desenvolvido utilizando o framework Flutter, visando multiplataforma (Android, iOS e Web). A comunicação será baseada em APIs REST ou MQTT fornecidas pelo SafeVax.

#### 4.2. Linguagens e Tecnologias

- **IDE:** Visual Studio Code ou Android Studio.
- **Frontend:** Flutter/Dart.
- **Backend:** Integração via APIs externas do sistema SafeVax.
- **Banco de Dados:** SQLite ou Firebase (dependendo do escopo final do armazenamento local/remoto).

### 5. Cronograma de Desenvolvimento

Fase	Data de Início	Data de Término
Levantamento de Requisitos	13/12/2024	14/12/2024
Desenvolvimento do Frontend	14/12/2024	20/01/2024
Integração com APIs SafeVax	20/12/2024	21/12/2024
Testes e Validação	20/12/2024	21/12/2024

## 6. Riscos e Mitigações

### 6.1. Risco 1: Atrasos no desenvolvimento.

- **Descrição:** Caso o cronograma não seja cumprido, o projeto pode ser atrasado.
- **Mitigação:** Adotar metodologias ágeis (Scrum), com sprints semanais e acompanhamento constante do progresso.

### 6.2. Risco 2: Falhas de integração com APIs externas.

- **Descrição:** Caso ocorra uma falha na integração de sistemas externos, o sistema pode ser prejudicado.
- **Mitigação:** Realizar testes de integração antecipada e monitoramento contínuo após a implantação.

### 6.3. Risco 3: Problemas de usabilidade da interface.

- **Descrição:** A interface pode não atender às expectativas dos usuários.
- **Mitigação:** Realizar testes de usabilidade durante o desenvolvimento, ajustando a interface com base no feedback coletado.