

# SAFEVAX V2

## Documento de Requisitos – Projeto SafeVax V2

### Introdução

O projeto SafeVax V2 tem como objetivo oferecer uma solução automatizada, segura e escalável para o monitoramento e a proteção no armazenamento de vacinas em ambientes hospitalares. Utilizando dispositivos IoT e uma arquitetura distribuída, o sistema garante que as condições ideais de conservação sejam mantidas, contribuindo para a segurança dos pacientes e a eficiência dos processos de gerenciamento.

---

### Escopo

O sistema SafeVax V2 abrange as seguintes funcionalidades:

- **Monitoramento em Tempo Real:**
    - Coleta de dados de temperatura e estado dos freezers por meio de sensores (ex.: DHT11 e sensores de abertura/fechamento de portas).
  - **Registro e Controle de Acessos:**
    - Registro dos acessos através de tecnologia RFID.
  - **Integração e Comunicação:**
    - Envio dos dados coletados, criptografados e compactados (usando Huffman), para um broker MQTT.
    - Estruturação de tópicos MQTT que possibilitam a identificação individual de cada freezer e seus sensores.
  - **Processamento Centralizado:**
    - Um servidor Flask capta as informações do broker MQTT, descompacta e descriptografa os dados e os registra em um banco de dados PostgreSQL.
    - O servidor Flask expõe os dados via API HTTP para consumo por outros módulos.
  - **Visualização e Notificações:**
    - Um dashboard interativo, desenvolvido com Streamlit, se conecta à API do servidor Flask para gerar gráficos, tabelas e relatórios históricos.
    - Um aplicativo mobile, desenvolvido em Flutter, consome dados (criptografados e compactados) do servidor MQTT ou API, realiza a devida descriptografia e descompactação e envia alertas e notificações aos gestores.
- 

### Objetivos

- **Garantir a eficácia das vacinas:**  
Monitorar e manter as condições ideais de armazenamento por meio de medições precisas e automáticas.
  - **Melhorar a segurança:**  
Registrar os acessos e identificar os responsáveis pela abertura dos refrigeradores utilizando RFID.
  - **Automatizar a gestão:**  
Reduzir a dependência de processos manuais e minimizar erros humanos, utilizando comunicação via MQTT e processamento centralizado.
  - **Facilitar a análise dos dados:**  
Disponibilizar informações de forma clara e interativa por meio de dashboards (Streamlit) e relatórios históricos.
  - **Assegurar a integridade e a segurança dos dados:**  
Aplicar algoritmos de compactação (Huffman) e criptografia robusta em todas as etapas da comunicação entre os dispositivos IoT, o servidor e as interfaces de visualização e alerta.
- 

### Requisitos Funcionais

#### Monitoramento e Controle

- **RF1:** O sistema deverá monitorar a temperatura interna dos refrigeradores em tempo real utilizando o sensor DHT11.
- **RF2:** O sistema deverá registrar o estado (aberto/fechado) da porta dos containeres através de um sensor de luz infravermelho PIR. O sensor ficará fixado na parte superior do container, voltado para a parte móvel da porta. Ao detectar a alteração na reflexão da luz infravermelha – que indica a abertura ou fechamento da porta – o sensor informará o sistema imediatamente. Além disso, se a porta for aberta e não for identificada a presença de um cartão RFID dentro de 10 segundos, o sistema deverá emitir um alerta.
- **RF3:** O sistema deverá identificar e registrar os usuários que acessam os freezers utilizando tecnologia RFID.

## Integração e Comunicação

- **RF5:** O sistema deverá enviar os dados coletados, devidamente criptografados e compactados (usando o algoritmo de Huffman), para um broker MQTT.
- **RF6:** O sistema deverá utilizar uma estrutura de tópicos MQTT para identificar individualmente cada freezer e seus respectivos sensores.
- **RF7:** Um servidor Flask deverá captar os dados enviados via MQTT, descompactá-los e descriptografá-los, registrando as informações em um banco de dados SQLite.
- **RF8:** O servidor Flask deverá disponibilizar os dados processados via API HTTP para consumo por outras interfaces do sistema.

## Alertas e Notificações

- **RF9:** O sistema deverá enviar notificações automáticas via WhatsApp aos gestores em casos de irregularidades.
- **RF10:** Um aplicativo mobile, desenvolvido em Flutter, deverá ser implementado para receber dados (por MQTT ou API), descriptografar e descompactar as informações e emitir alertas e notificações imediatas aos usuários quando parâmetros críticos forem ultrapassados.

## Visualização de Dados

- **RF11:** Um dashboard interativo, utilizando a ferramenta Streamlit, deverá ser disponibilizado para a visualização dos dados coletados de forma clara e legível.
- **RF12:** O dashboard deverá apresentar gráficos, tabelas e relatórios históricos para auxiliar na análise e tomada de decisões.

---

## Requisitos Não Funcionais

### Performance

- **RNF1:** O sistema deverá realizar a coleta e transmissão dos dados com baixa latência, garantindo atualizações em tempo real.
- **RNF2:** A compactação dos dados deverá reduzir significativamente o volume de informações transmitidas, sem comprometer a integridade dos dados.

### Segurança

- **RNF3:** A comunicação entre os dispositivos IoT, o broker MQTT, o servidor Flask e os aplicativos deverá ser protegida por criptografia robusta.
- **RNF4:** O acesso ao dashboard e aos dados deverá ser controlado por autenticação e autorização.

### Usabilidade

- **RNF5:** O aplicativo mobile em Flutter deverá apresentar uma interface intuitiva e responsiva.
- **RNF6:** O dashboard desenvolvido em Streamlit deverá oferecer uma experiência de usuário clara, com design responsivo.

### Escalabilidade e Manutenção

- **RNF7:** A arquitetura do sistema deverá ser modular e escalável.
- **RNF8:** O código deverá ser documentado e seguir boas práticas de desenvolvimento.

---

## Requisitos de Integração e Melhoria

- **RM1:** Implementar algoritmos de compactação de dados (mínimo dois), utilizando o algoritmo de Huffman.
  - **RM2:** Implementar um algoritmo de criptografia robusto.
  - **RM3:** Desenvolver um aplicativo mobile em Flutter que se conecte ao servidor MQTT ou à API do Flask.
  - **RM4:** Desenvolver um dashboard interativo utilizando Streamlit.
  - **RM5:** Implementar um mecanismo de cadastro e gerenciamento de múltiplos freezers, utilizando tópicos MQTT estruturados.
- 

## Considerações Finais

O projeto SafeVax V2 visa transformar o monitoramento e a segurança no armazenamento de vacinas em ambientes hospitalares através de uma abordagem integrada que alia tecnologia IoT, segurança de dados e interfaces amigáveis para o usuário. Este documento de requisitos servirá de base para o desenvolvimento e a validação do sistema, garantindo que todas as funcionalidades essenciais e melhorias sejam implementadas de acordo com as necessidades dos stakeholders e as melhores práticas de engenharia.