

# **Guia Técnico do Projeto OpSafe**

## **Diretrizes Técnicas, Arquiteturais, Padrões de Qualidade e Segurança**

Caio César Ponte

15 de novembro de 2025

## Introdução

Este documento técnico define os padrões, tecnologias, métodos de engenharia, regras de qualidade e diretrizes arquiteturais que serão seguidos no projeto **OpSafe**. Ele serve como guia para desenvolvimento, validação, versionamento, governança, testes, segurança, LGPD e boas práticas para manter a plataforma escalável, segura e consistente.

# 1 Padrões de Interface

## 1.1 Família de Ícones

A plataforma utilizará a biblioteca **Lucide Icons** como padrão visual principal. Características:

- Traços lineares, leves e minimalistas.
- Conjunto amplo e padronizado.
- Adaptado a dark/light mode.

## 1.2 Família de Cores

A identidade visual seguirá um conjunto modular:

- **Primária:** #1A73E8 (azul operacional)
- **Secundária:** #0A2F5A (azul profundo – seriedade e segurança)
- **Atenção:** #F9A825
- **Sucesso:** #43A047
- **Erro:** #E53935
- **Cinzas:** escala de #111111 a #F5F5F5

## 1.3 Família de Fontes

- **Inter** (plataforma Web)
- **Roboto** (mobile e compatibilidade Android)
- **SF Pro Text** (compatibilidade iOS)

## 1.4 Responsividade e Grid

A aplicação segue:

- **CSS Grid + Flexbox**
- Breakpoints: 480px, 768px, 1024px, 1440px
- Suporte completo ao Mobile-first

## 1.5 Layout

- **Layout Master-Detail** para equipamentos.
- **Cards modulares** para dashboards.
- **Side Navigation** em modo desktop.
- **Bottom Navigation** em mobile.

## 2 Arquitetura de Software

### 2.1 Clean Architecture – Camadas

- **Domain:** Regras de negócio puras, entidades.
- **Use Cases:** Operações atômicas da aplicação.
- **Infrastructure:** Database, API, drivers externos.
- **Presentation:** Camada de UI (React / Mobile).

### 2.2 SOLID

Aplicado em todos os módulos:

- SRP: Um módulo = uma responsabilidade.
- OCP: Código aberto para extensão, fechado para modificação.
- LSP: Módulos substituíveis sem quebrar fluxos.
- ISP: Interfaces pequenas, específicas.
- DIP: Camadas superiores dependem de abstrações, não implementações.

### 2.3 Padrões de Código

- TypeScript estrito.
- ESLint + Prettier.
- Commits semânticos (feat/fix/docs/refactor/test).
- Versionamento semantic versioning (semver).

## 3 Banco de Dados

### 3.1 Modelo NoSQL – Padrão

O OpSafe utilizará um banco **MongoDB** ou **Firestore** com os princípios:

- Coleções independentes.
- Documentos auto-contidos.
- Campos versionados para retrocompatibilidade.

### 3.2 Estrutura Base

- equipments
- operators
- assignments
- maintenance
- contracts
- alerts
- audit\_logs

### 3.3 Blockchain (Opcional e Modular)

Aplicável para:

- garantias de integridade de termos digitais;
- trilhas de auditoria imutáveis;

Usando:

- Hash SHA-256;
- Armazenamento fora da cadeia (off-chain) para dados pesados.

## **4 Segurança da Informação**

### **4.1 LGPD – Conformidade**

- Minimização de dados.
- Consentimento explícito para operadores.
- Relatórios de transparência.
- Controlador e operador bem definidos.

### **4.2 Criptografia**

- AES-256 para dados sensíveis.
- TLS 1.3 para tráfego.
- Hashing Argon2 para senhas.

### **4.3 Autenticação e Autorização**

- OAuth2 + JWT.
- Expiração de tokens + refresh tokens.
- RBAC (Role Based Access Control).

### **4.4 DevSecOps**

- SAST em cada PR.
- Dependabot para pacotes.
- Pipeline com bloqueio de CVEs acima de CVSS 7.

## **5 Métodos e Fluxos de Cadastro**

### **5.1 Cadastro de Equipamento**

- Identificador único.
- Categoria e contrato associado.
- Ciclo de vida inicial como “Disponível”.

### **5.2 Cadastro de Operador**

- Dados mínimos (LGPD).
- Função e posto.
- Termo de responsabilidade digital.

### **5.3 Cadastro de Manutenção**

- Preventiva ou corretiva.
- Data programada e realizada.
- Registro de custo.



## **6 Testes e Qualidade**

### **6.1 Testes Unitários**

- Cobertura mínima: 85%.
- Testes para cada use case isolado.

### **6.2 Testes de Integração**

- Interação entre camadas.
- Testes com banco real (ambiente isolado).

### **6.3 Testes End-to-End**

- Cypress / Playwright.
- Simular fluxo completo: entrega → termo digital → devolução.

### **6.4 Ambiente Controlado**

- DEV → STAGE → PROD
- STAGE com dados fictícios.
- Proibição de dados reais fora do PROD.

## **7 CI/CD – Entrega Contínua**

### **7.1 Pipeline**

- Build
- Testes unitários
- SAST
- Deploy automático para STAGE
- Deploy manual e aprovado para PROD

### **7.2 Logs e Observabilidade**

- Logs estruturados (JSON)
- Monitoramento via Prometheus ou Datadog
- Alertas de uptime

## **8 Regras de Negócio – Padronização**

### **8.1 Identificação Única**

Nenhum recurso pode existir sem ID único global.

### **8.2 Termos Digitais**

Toda entrega e devolução gera registro permanente auditável.

### **8.3 EPIs Vencidos**

Sistema bloqueia automaticamente distribuição.

### **8.4 Manutenção Obrigatória**

Itens com defeito entram em quarentena até liberação.

## 9 Definição de Pronto (DoD)

Um item de trabalho só é considerado concluído quando:

- Código revisado.
- Testes unitários e integração passando.
- Documentação atualizada.
- Sem falhas de lint.
- Sem vulnerabilidades críticas.

## 10 Conclusão

Este documento estabelece a base técnica, arquitetural e operacional que sustenta o projeto OpSafe. Ele deve ser utilizado como referência contínua pelo time de desenvolvimento, design, análise, segurança e gestão, garantindo consistência, escalabilidade e aderência à legislação aplicável, especialmente à Lei nº 14.967/2024 e à LGPD.