


-  Sistema de Gestão — Lar São Francisco na Providência de Deus
 -  Visão Geral
 -  Funcionalidades
 -  Arquitetura e Stack
 - Destaques técnicos
 -  Preview
 -  Como executar localmente
 - Pré-requisitos
 - Passos
 -  Deploy (Firebase Hosting)
 -  Estrutura do Projeto
 -  Perfis de Acesso (RBAC)
 -  Licença


Sistema de Gestão — Lar São Francisco na Providência de Deus

Plataforma digital completa para gestão de instituições de saúde e assistência social



Visão Geral

O **Sistema de Gestão ALSF** nasceu para digitalizar e modernizar integralmente as operações do [Lar São Francisco na Providência de Deus](#) — uma instituição de saúde e assistência social. O sistema substitui planilhas e processos manuais por uma plataforma web robusta, em tempo real e com controle de acesso por perfil.

 Desenvolvido de *zero a produção* em ambiente real, com dados reais e usuários ativos.



Funcionalidades

Módulo	Descrição
 Dashboard	Visão executiva com KPIs, alertas e gráficos em tempo real
 Recepção	Registro de atendimentos, fila de espera e triagem digital
 Consultório	Interface médica com prontuário, prescrição e histórico do paciente
 Farmácia	Dispensação de medicamentos, rastreamento por missão e controle de estoque
 Estoque	Gestão completa de insumos com movimentações e alertas de reposição
 Financeiro	Controle de receitas, despesas e relatórios por período
 Missões / Eventos	Planejamento, alocação de recursos e acompanhamento de missões sociais
 Beneficiários	Cadastro e histórico completo dos assistidos
 Voluntários	Gestão da equipe voluntária e vinculação a missões
 Arrecadação	Campanhas de captação de recursos e doações
 Calendário	Visão integrada de eventos e missões
 Notificações	Sistema de alertas internos em tempo real
 Usuários & Papéis	RBAC com 9 perfis distintos (admin, médico, farmácia, recepção...)
 Modo Apresentação	Dashboard executivo para projeção em reuniões e apresentações



Arquitetura e Stack

Frontend (SPA)

React 19 · TypeScript · Vite · Lucide Icons

Backend-as-a-Service
Firebase Auth · Cloud Firestore · Firebase Hosting

AI Integration
Google Generative AI (Gemini) – análises e insights

Destaques técnicos

- ⚡ **Tempo real** — todos os dados sincronizados via Firestore `onSnapshot`
- 🔒 **RBAC completo** — 9 perfis de acesso com rotas e abas protegidas por role
- 📱 **Mobile-first** — interface totalmente responsiva com painel de filtros colapsável
- 🤖 **AI-powered** — análise automatizada de dados com Google Gemini
- 🔒 **Segurança** — Firestore Security Rules restritas por autenticação e role
- 💾 **Persistência inteligente** — aba ativa salva no `localStorage` e restaurada após reload
- 🏠 **Modo Missão** — painel dedicado para operações de campo com dados offline-first



Preview

🌐 Acesse ao vivo: gestaoalsf.web.app



Como executar localmente

Pré-requisitos

- [Node.js 18+](#)
- Conta no [Firebase](#) com projeto configurado

Passos

```
# 1. Clone o repositório
git clone https://github.com/eupedrodiogo/gestao_alsf.git
```

```
cd gestao_alsf

# 2. Instale as dependências
npm install

# 3. Configure as variáveis de ambiente
cp .env.example .env.local
# Preencha com suas credenciais do Firebase

# 4. Inicie o servidor de desenvolvimento
npm run dev
```

A aplicação estará disponível em <http://localhost:5173>

Deploy (Firebase Hosting)

```
# Build de produção
npm run build

# Deploy para Firebase Hosting
npm run deploy --only hosting
```

Estrutura do Projeto

```
gestao_alsf/
├── index.tsx           # Aplicação principal (componentes e lógica de UI)
├── types.ts            # Interfaces e tipos TypeScript
├── firebase.ts         # Inicialização e configuração do Firebase
├── AuthContext.tsx    # Contexto de autenticação (React Context + Firebase
Auth)
├── useFirestore.ts     # Hook genérico para CRUD em tempo real com Firestore
├── Login.tsx           # Tela de autenticação
├── SetupAdmin.tsx      # Configuração inicial do administrador
├── MissionControl.tsx  # Painel de controle de missões de campo
├── MissionModePanel.tsx # Interface dedicada para operações em missão
├── FinancialModal.tsx   # Modal de lançamentos financeiros
├── StockMovementModal.tsx # Modal de movimentações de estoque
├── VolunteerModal.tsx  # Modal de gestão de voluntários
├── PresentationSlides.tsx # Slides para modo de apresentação executiva
├── PresentationDashboard.tsx # Dashboard para projeção/apresentação
├── offlineMode.ts      # Suporte a modo offline
├── scripts/            # Utilitários de sincronização de dados
└── firestore.rules     # Regras de segurança do Firestore
```

```
|— firebase.json
|— vite.config.ts
```

```
# Configuração de deploy (Hosting + Firestore)
# Configuração do bundler
```

Perfis de Acesso (RBAC)

Perfil	Acesso
admin	Acesso total a todos os módulos
operador	Todos os módulos exceto gestão de usuários
medico	Consultório, triagem e histórico de pacientes
farmacia	Farmácia, estoque de medicamentos e missões
recepcao	Registro e fila de atendimento
triagem	Triagem e sinais vitais
voluntario	Painel de voluntários e missões
estoque	Gestão de inventário
financeiro	Módulo financeiro

Licença

Este projeto está sob licença proprietária. Todos os direitos reservados ao autor.
O código é disponibilizado publicamente para fins de portfólio profissional.

Desenvolvido com  por **Pedro Diogo**

[GitHub](#) · [LinkedIn](#)