

Arquitetura - Frontend

Tecnologias

- Next.js 15.5.4: Framework React
- React 19.1.0: Biblioteca UI
- TypeScript 5: Tipagem estática
- TailwindCSS 4: Estilização
- Zustand 5.0.7: Estado global
- React Query 5.85.3: Cache e requisições
- Axios 1.11.0: Cliente HTTP
- [Socket.IO](#) Client 4.7.5: WebSocket
- next-intl 4.3.4: Internacionalização
- Zod 4.1.0: Validação de schemas
- React Hook Form 7.62.0: Formulários

Frontend Principal

- App Router (Next.js 15)
- Internacionalização (i18n) - PT, EN, ES
- Autenticação integrada com Keycloak
- Sistema de permissões granular (scopes)
- Multi-tenancy
- Gerenciamento de estado com Zustand
- Requisições HTTP com React Query + Axios
- WebSocket para status de conexão e atualização das informações do Smart-calda

ESTRUTURA PRINCIPAL (raiz do projeto — `components/`, `services/`, `hooks/` e `stores/` ficam dentro de `app/`; `contexts/` fica na raiz, fora de `app/`):

`app/`

```
├── [locale]/           # Rotas por idioma
|   ├── login/          # Tela de login
|   ├── selecionar-unidade/ # Seleção de unidades
|   ├── gerenciamento-unidades/ # CRUD de unidades
|   ├── gerenciamento-usuarios/ # CRUD de usuários
|   ├── gerenciamento-smart-caldas/ # Gestão de SmartCaldas
|   └── manutencao-smart-caldas/ # Manutenção de SmartCaldas
```

- | └─ auditoria/ # Logs de auditoria
- | └─ ... # + cadastro-auxiliares, controle-estoque, dashboard-gerencial, dashboard-operacional, gerenciamento-fazendas, gerenciamento-fornecedores, gerenciamento-insumos-agricolas, gerenciamento-supervisorios, movimentacoes, notas-entradas, ordens-servicos, relatorios, romaneios, sobre, unauthorized
- | └─ components/ # Componentes React
- | └─ ui/ # Componentes base (shadcn/ui)
- | └─ guards/ # Proteção de rotas/ações
- | └─ SmartCaldas/ # Componentes de SmartCaldas
- | └─ LayoutClient.tsx # Layout principal
- | └─ ... # + Audit, AuxiliaryRecords, DashboardOperacional, Defensives, Farms, NotasEntradas, Reports, ServiceOrders, SmartCaldasMaintenance, Sobre, Stock, StockMovements, SupervisorioDesigner, SupervisorioViews, Suppliers, Unidades, Users&Roles, truck-routes, pdf
- | └─ services/ # Serviços de API
- | └─ api.service.ts # Cliente Axios configurado
- | └─ auth.service.ts # Autenticação Keycloak
- | └─ token.service.ts # Validação de tokens JWT
- | └─ clp-websocket.service.ts # WebSocket para status CLP
- | └─ smartCaldas.service.ts # API de SmartCaldas
- | └─ ... # + audit, auxiliary-records, classes, configurations, cost-center, dashboard-gerencial, defensive/defensive-labels, drivers, farm, geographic, invoice, location, maintenance, operations, reports, responsible-parties, roles, stock*, supervisorio*, supplier, truck-routes, trucks, unidade, user, work-order*
- | └─ hooks/ # Custom hooks
- | └─ useAuth.ts # Hook de autenticação
- | └─ useApi.ts # Hook para requisições
- | └─ useCLPConnectionStatus.ts # Status de conexão CLP
- | └─ ... # + dezenas de hooks por domínio (useFarms, useStock, useSuppliers, useTrucks, useDrivers, useReports, useInvoices, useWorkOrders etc.)
- | └─ stores/ # Estado global
- | └─ auth.store.ts # Store Zustand para auth
- | contexts/ # Contextos React (raiz do projeto)
- | └─ PermissionsContext.tsx # Context de permissões

FUNCIONALIDADES PRINCIPAIS:

- Autenticação com Keycloak (JWT tokens)
- Gerenciamento de unidades (branches)
- Gerenciamento de SmartCaldas
- Manutenção de SmartCaldas
- Monitoramento de status de conexão CLP (WebSocket)
- Sistema de auditoria
- Cadastro de fornecedores, defensivos, caminhões, motoristas
- Gestão de operações agrícolas
- Exportação de PDFs
- Sistema de permissões granular (modulo:acao)