

Integrações Front ↔ Backend

FRONTEND ↔ BACKEND

Comunicação: REST API + WebSocket

REST API:

- Base URL: configurável via NEXT_PUBLIC_API_URL
- Autenticação: JWT Bearer Token
- Headers obrigatórios:
 - Authorization: Bearer <access_token>
 - X-Tenant-Id: <tenant_id>
- Interceptors Axios:
 - Request: Adiciona token e tenant automaticamente
 - Response: Trata erros 401 (token expirado)

Endpoints principais:

- GET /branches # Listar filiais
- GET /smart-caldas # Listar SmartCaldas
- GET /smart-caldas/:id # Detalhes SmartCalda
- POST /smart-caldas # Criar SmartCalda
- PUT /smart-caldas/:id # Atualizar SmartCalda
- DELETE /smart-caldas/:id # Excluir SmartCalda
- GET /smart-calda-maintenance # Histórico de manutenções
- POST /smart-calda-maintenance # Criar manutenção
- GET /audit/logs # Logs de auditoria

WebSocket:

- Namespace: /container-connection-status
- Conexão:  client
- Eventos:
 - subscribe: Inscrever-se em status de container
 - unsubscribe: Cancelar inscrição
 - connection-status-update: Atualização de status
- Autenticação: Token JWT no handshake

PLC MANAGER ↔ BACKEND

Comunicação: REST API

Autenticação:

- Login via /auth/login (Keycloak)
- Recebe access_token e refresh_token
- Token incluído em todas as requisições
- Header X-Tenant-ID obrigatório

Endpoints utilizados:

- GET /branches # Listar filiais disponíveis
- GET /smart-caldas?branchId=xxx # Buscar SmartCaldas da filial
- GET /smart-caldas/:id # Detalhes do SmartCalda

Fluxo de sincronização:

1. Usuário faz login no PLC Manager
2. Seleciona filial (branch)
3. CLP-Local busca SmartCaldas do backend para a filial
4. Cria containers locais baseados nos SmartCaldas
5. Containers são somente-leitura (não podem ser criados/editados localmente)

PLC MANAGER ↔ CLPs (Hardware)

Comunicação: OPC UA (OPC Unified Architecture)

Protocolo: OPC UA

- Endpoint padrão: opc.tcp://<IP>:4840
- Descoberta automática de variáveis
- Leitura/escrita de variáveis
- Monitoramento contínuo

Variáveis OPC UA:

- Node ID: identificador único da variável
- Path: caminho hierárquico
- Data Type: tipo de dado (BOOL, INT, REAL, etc.)
- Value: valor atual

PLC MANAGER ↔ BANCO LOCAL

Comunicação: PostgreSQL (Nuvem)

Banco de dados local:

- Tabelas: containers, variables, variable_history, write_commands
- Armazena configuração local
- Histórico de valores de variáveis
- Status de conexão dos containers
- Comando para serem executados

BACKEND ↔ KEYCLOAK

Comunicação: REST API + JWT

Funcionalidades:

- Autenticação de usuários
- Validação de tokens JWT
- Gerenciamento de roles
- Multi-tenancy (realms)
- Token introspection