Fluxo de Dados - arquitetura

FLUXO DE AUTENTICAÇÃO

1. Usuário acessa frontend → /login
2. Frontend chama Keycloak → /auth/realms/{realm}/protocol/openid-connect/token
3. Keycloak valida credenciais e retorna JWT tokens
4. Frontend armazena tokens no Zustand (localStorage)
5. Frontend extrai tenant_id do token JWT
6. Frontend envia token em todas as requisições (header Authorization)
7. Backend valida token com Keycloak
8. Backend verifica permissões/roles do usuário

FLUXO DE SINCRONIZAÇÃO PLC MANAGER

1. Usuário abre PLC Manager
2. PLC Manager verifica sessão salva (session.json)
3. Se não autenticado, mostra tela de login
4. Usuário faz login → backend valida com Keycloak
5. PLC Manager recebe tokens e salva em session.json
6. PLC Manager busca branches disponíveis
7. Usuário seleciona filial (branch)
8. PLC Manager busca SmartCaldas da filial: GET /smart-caldas?branchId=xxx
9. Para cada SmartCalda, cria container local:
    - Extrai IP/porta da URL (opc.tcp://IP:PORT)
    - Cria CLPContainer com configuração OPC UA
    - Mapeia SmartCalda ID → Container ID
10. Containers aparecem na interface
11. Usuário pode conectar/desconectar containers
12. Status de conexão é atualizado no banco local
13. Status pode ser consultado via WebSocket pelo frontend

FLUXO DE MONITORAMENTO CLP

1. Usuário conecta container no PLC Manager

2. PLC Manager estabelece conexão OPC UA com CLP
3. PLC Manager descobre variáveis automaticamente (opcional)
4. Usuário inicia monitoramento contínuo
5. PLC Manager lê variáveis periodicamente (intervalo configurável)
6. Valores são atualizados na interface
7. Valores são salvos no banco local (histórico)
8. Status de conexão é atualizado no banco local
9. Backend consulta status no banco local (polling)
10. Backend envia atualização via WebSocket para frontend
11. Frontend atualiza UI com status em tempo real

FLUXO DE MANUTENÇÃO

1. Usuário registra manutenção no frontend
2. Frontend envia POST /smart-calda-maintenance
3. Backend salva manutenção no banco
4. PLC Manager pode consultar histórico via API
5. Histórico aparece no frontend
