Autenticação e Segurança

KEYCLOAK

Servidor: Keycloak (Docker)
Porta: 8080 (padrão)

Realms:

- Cada tenant tem um realm no Keycloak
- Multi-tenancy via realms

Clientes:

- iagro-public-client (frontend)
  - Access Type: public
  - Direct Access Grants: ON
- iagro-backend (backend)
  - Access Type: confidential
  - Service Accounts: ON

Roles:

- iagro:master: Acesso total
- iagro:admin: Administrador
- iagro:operator: Operador
- Roles customizadas por módulo

Permissões:

- Sistema de scopes verificado no código via `@Scopes('modulo:acao')` (ex.: gestao:read, usuarios:create)
- Armazenamento real no Keycloak: atributo `perm:<modulo>` em `rawAttributes` das roles, contendo um JSON serializado das ações permitidas (ex.: `{"read":true,"create":true}`) — não uma string simples `modulo:acao`

JWT TOKENS

Estrutura:

- Header: algoritmo, tipo (JWT)
- Payload: user info, roles, tenant, exp
- Signature: assinatura do Keycloak

Campos importantes:

- tenant_id: ID do tenant
- unit_id: ID da unidade
- realm_access.roles: Roles do usuário
- exp: Data de expiração

Validação:

- Backend valida token com Keycloak
- Verifica assinatura e expiração
- Extrai tenant_id para isolamento

SEGURANÇA

- HTTPS em produção (recomendado)
- Tokens JWT com expiração
- Refresh tokens para renovação
- Validação de permissões em cada requisição
- Auditoria de todas as operações
- Soft delete para recuperação de dados
- Isolamento de dados por tenant
