Centralizador

Objetivo principal

Sistema de backoffice que cadastra e administra os múltiplos ambientes do sistema Novo Avanço hospedados na AWS (um ambiente por conta/cliente), controlando empresas, usuários, grupos e permissões, e funcionando como ponto único para sistemas externos descobrirem a URL do ambiente correto de um cliente a partir de e-mail ou hash.

Principais tecnologias

NestJS / TypeScript / Prisma + PostgreSQL / Redis / Angular 17 + Angular Material / JWT

Principais funcionalidades

- Autenticação: Login/logout com JWT e sessão do usuário logado
- Ambientes AWS: CRUD dos ambientes (nome, URL base, access key, homepage)
- Grupos de Ambientes: Agrupamento de ambientes AWS, associados a labels e perfis
- Usuários por Ambiente: Vínculo de usuários a ambientes específicos, com upsert em lote
- Empresas: CRUD de empresas clientes vinculadas a um ambiente AWS (CNPJ, endereço, produtos, bloqueio)
- Perfis de acesso: Perfis de permissão associados a grupos de ambientes
- Labels: Etiquetas visuais (cor/ícone) para organizar grupos de ambientes
- Usuários do sistema: CRUD de usuários administradores/operadores do Centralizador
- Resolução de ambiente: Endpoint que descobre a URL base do ambiente de um usuário a partir de e-mail ou hash
- Configurações: Tela de configurações do sistema no frontend

Principais integrações

- PostgreSQL: Banco de dados principal da aplicação, via Prisma ORM
- Redis: Presente na infraestrutura Docker do projeto (uso específico no código não confirmado)
- Autenticação via hash (header authorizationhash / HASHGUARD): Canal usado por sistemas externos para consultar/cadastrar hashes e resolver o ambiente do cliente

- Endpoint "listForCotacao": Lista de ambientes consumida por um sistema externo de Cotação (integração não totalmente confirmada apenas pelo código)

Revision #1
Created 2026-08-18 18:29:43 UTC by Pedro Souza
Updated 2026-08-18 18:29:43 UTC by Pedro Souza