Deploy no Coolify (Frontend + Backend)

Este repositório possui:

- **Frontend**: Next.js (porta 3000)
- **Backend**: Medusa (porta 9000)
- **Postgres** + **Redis**

A forma mais simples no Coolify é usar **Docker Compose** com 1 stack.

1) Criar a aplicação no Coolify

1. No Coolify, crie uma nova aplicação a partir do repositório.
2. Selecione **Docker Compose**.
3. Aponte para o arquivo: `docker-compose.coolify.yml`.

2) Configurar domínios/rotas

Você normalmente vai querer **2 domínios**:

- **Frontend**: `https://www.seudominio.com` → serviço `frontend` (porta 3000)
- **Backend**: `https://api.seudominio.com` → serviço `backend` (porta 9000)

> Importante: o frontend roda no browser e precisa acessar o backend por uma URL pública.

3) Variáveis de ambiente (Coolify)

Use o arquivo `.env.coolify.example` como referência e configure no Coolify:

- `NEXT_PUBLIC_MEDUSA_BACKEND_URL=https://api.seudominio.com`
- `STORE_CORS=https://www.seudominio.com`
- `ADMIN_CORS=https://api.seudominio.com`
- `JWT_SECRET` e `COOKIE_SECRET` com valores fortes

Sanity (se estiver usando):

- `NEXT_PUBLIC_SANITY_PROJECT_ID`
- `NEXT_PUBLIC_SANITY_DATASET`

4) Primeira inicialização (migrations/seed)

Dependendo do estado do banco, pode ser necessário rodar migrations/seed do Medusa.
No Coolify, abra o terminal do container `backend` e rode, conforme necessário:

- `npm run migrate`
- `npm run seed`

5) Checks rápidos

- Backend: `https://api.seudominio.com/store`
- Admin (Medusa): `https://api.seudominio.com/app`
- Frontend: `https://www.seudominio.com`

Se der erro de CORS, revise `STORE_CORS`/`ADMIN_CORS`.