

Documentação – TrakNor MVP

1. Stack Tecnológica (MVP)

Camada	Escolha	Por quê
Frontend	React 18 + Vite + TypeScript	Build rápido e DX simples
UI	Mantine v7 + Tailwind utilities	Componentes prontos + design tokens
Estado	React Query	Cache/fetch sem Redux
Backend	Django 4.2 LTS + DRF	Produtivo + REST nativo
Auth	SessionAuth (Admin) + DRF Token	Basta p/ testes e Postman
Multitenancy	django-tenants (schema-per-tenant)	SaaS do dia 0
DB	PostgreSQL 16	Íntegro e compatível com schemas
DevOps	Docker Compose (api, web, db) + GitHub Actions	Um comando para rodar, testes CI

2. Mapa de Telas do MVP

Tela	Objetivo	Principais Componentes/Interações
1. Visão Geral (Dashboard)	Mostrar KPIs simples do PMOC e atividades pendentes	Cards: OS abertas, Equip. críticos; Gráfico “OS vs prazo” (ECharts); Tabela “Próximos checklists”
2. Ativos	CRUD de Equipamentos	Filtro por setor/cliente; Colunas: Tag, Modelo, Local, Próx. OS
3. Ordens de Serviço (OS)	Gerir execuções do PMOC/corretivas	Tabela com status badge; Drawer lateral para detalhes; Upload de evidências (img/pdf)

4. Solicitações	Registrar pedidos de manutenção (portal interno)	Formulário rápido (setor, descrição, prioridade); Kanban “Nova → Em triagem → Aprovada”
5. Planos	Cadastros de checklists recorrentes (mensal, trimestral, etc.)	Wizard em 3 passos: metadados, perguntas, periodicidade
6. Métricas	Relatórios rápidos (export CSV/PDF)	Filtro período/cliente; Gráficos de barras: OS/ativo
7. Estoque	Inventário de peças críticas	Quantidade, ponto de reposição, última saída
8. Procedimentos	Biblioteca de POPs/PMOC	Visualizador markdown/pdf; Tag “versão”
9. Relatórios	Geração manual de PDF PMOC	Selecionar período + local → botão “Gerar PDF”

3. Modelo de Dados (essencial)

Entidade	Campos-chave	Relações
Tenant	id, nome, cnpj	1-N Users
User	id, email, role	FK Tenant
Setor	id, nome	FK Tenant
Equipamento	id, tag, tipo, setor_id, próximo_serviço	FK Setor
WorkOrder	id, tipo, status, equipamento_id, data_agenda, responsavel_id	FK Equipamento
Solicitacao	id, setor_id, descrição, prioridade, status	
Plano	id, nome, frequência (mensal, ...)	N-N Perguntas
PerguntaChecklist	id, enunciado, tipo_resposta	
OS_ChecklistItem	workorder_id, pergunta_id,	

	resposta, anexo
EstoqueItem	id, descrição, quantidade, ponto_reposicao
Procedimento	id, título, arquivo_pdf, versao
RelatorioPMOC	id, período, tenant_id, arquivo_pdf

4. Arquitetura Lógica

React UI → DRF API → PostgreSQL (schema/tenant)

TokenAuth (api docs) / SessionAuth (admin)

5. Fluxo de CI/CD & Ambiente

1. Desenvolvimento local

docker compose up --build

2. Pipeline CI (GitHub Actions)

- Lint + PyTest + Vitest
- Build imagens backend:dev e frontend:dev

3. Deploy MVP (EC2 ou Lightsail)

git pull && docker compose up -d

4. Backups

- Cron pg_dump diário → S3

6. Segurança MVP

- HTTPS (Nginx + Let's Encrypt)
- Senhas PBKDF2, CSRF ativo
- Permissões por IsAuthenticated + DjangoModelPermissions
- Backup criptografado (AES-256)

7. Testes Automáticos

Camada	Ferramenta	Cobertura meta
--------	------------	----------------

Backend (unit)	PyTest + Factory Boy	≥ 70 %
Backend (API)	PyTest + DRF APITestCase	Rotas CRUD principais
Frontend	Vitest + React Testing Library	Hooks & pages críticos
E2E (opcional)	Cypress (headless)	Fluxo de criação OS

8. Roadmap de Sprints

Sprint	Foco	Saída “shippable”
S1	Setup repo, Docker, modelos Tenant/User	Login admin, health-check
S2	CRUD Setores + Equipamentos	List/Detail + filtros
S3	CRUD WorkOrder + Checklists	Geração OS & execução
S4	Telas Visão Geral + Solicitações	Dashboard simples & Kanban
S5	Planos + Procedimentos	Wizard plano + viewer PDF
S6	Relatórios PMOC (PDF) + Métricas	Botão “Gerar PDF”, export CSV
S7	Estoque + Testes E2E	Inventário mín. & pipeline Cypress
S8	Hardening + deploy piloto	Backup, SSL, doc final

Próximos Passos

1. Validar storyboard com stakeholders (arquivos PNG).
2. Criar issues no GitHub por sprint e módulo.
3. Configurar domínio de staging e certificado SSL.
4. Iniciar S1 no próximo ciclo de 2 semanas.