

Growth Equestre — UI Web em Node.js (Hackathon)

Arquitetura, Plano de Execução e Backlog (MVP para substituir Streamlit).

Data: 10/02/2026

Resumo

Proposta prática para substituir a UI em Streamlit por uma UI Web em Node.js (Express + EJS), mantendo integração com o backend existente via proxy HTTP (/api). O MVP entrega: Dashboard, Kanban (CRM), listas de Leads/Parceiros e configuração básica.

Escopo do MVP

- UI Web (Express + EJS) com design consistente e responsivo.
- Proxy /api/* -> BACKEND_URL (evita CORS).
- Dashboard com KPIs (/crm/summary).
- Kanban com drag-and-drop (/crm/kanban; /crm/kanban/move).
- Leads e Parceiros (listas com busca).
- Fallback com mocks quando rotas ainda não existirem.

Arquitetura (alto nível)

Navegador -> UI Web (porta 3100) -> chamadas fetch para /api/* -> proxy para BACKEND_URL (porta 3000) -> resposta.

Contrato mínimo de API (recomendado)

- GET /health -> { status: 'ok' }
- GET /crm/summary -> { leads_total, partners_total, events_total }
- GET /crm/kanban -> { columns: [{ id, title, cards:[...] }] }
- POST /crm/kanban/move -> body: { card_id, from, to }
- GET /crm/leads -> [{ id, name, stage, cnae }]
- GET /crm/partners -> [{ id, name, segment, cnae }]

Plano de execução (sprints)

- Sprint 0: setup UI base + proxy; layout; .env endpoints.
- Sprint 1: MVP demo (dashboard + kanban + listas + mocks).
- Sprint 2: polimento (CRUD mínimo, auth opcional, logs e testes).

Backlog

Backlog detalhado: docs/github_issues.md (no pacote da UI Web).