**Roadmap com checkboxes**

Objetivo: estabilizar o MVP, habilitar notificações (e‑mail/WhatsApp) em produção e deixar o deploy redondo.

**M0 — Estabilização Dev (agora)**

* .env local vs. produção padronizados (MYSQL\_\*, WA\_\*, SMTP\_\*, JWT\_SECRET)
* Backend: CORS restrito em prod (liberar localhost:3001 só no dev)
* Logs: morgan dev e prefixos de erro (já adotado)
* Script scripts/test-db.mjs funcional
* Ajustar Api do front: **dev** → http://localhost:3002 | **prod** → VITE\_API\_URL=/api

**M1 — Fluxos essenciais do MVP**

* **Auth**: /auth/register, /auth/login, /auth/me (JWT 7d, JWT\_SECRET obrigatório)
* **Catálogo**: /establishments e /servicos?establishmentId=…
* **Slots**: GET /slots e POST /slots/toggle (apenas estabelecimento)
* **Agendamentos**: POST /agendamentos (transação + validação), GET /agendamentos//agendamentos/estabelecimento, PUT /agendamentos/:id/cancel
* Front: fluxo Cliente (login → listar serviços → escolher slot → confirmar → ver “Meus agendamentos” → cancelar)
* Front: fluxo Estabelecimento (login → gestão de slots/serviços)

**M2 — Notificações (email + WhatsApp)**

* Mailer: nodemailer com SMTP (senha de app) e HTML simples
* WhatsApp Cloud API: módulo lib/whatsapp.js (env: WA\_API\_VERSION, WA\_PHONE\_NUMBER\_ID, WA\_TOKEN)
* Rota de teste: /notify/whatsapp/text e /notify/whatsapp/template
* **Envio ao criar agendamento**: e‑mail + WhatsApp (template se fora da janela; texto se 24h aberta)
* Template **custom** “appointment\_confirmation” com variáveis (nome, data/hora)

**M3 — Webhook & rastreio**

* Webhook GET/POST /webhooks/whatsapp (verify + events)
* Persistir statuses (notifications): provider\_message\_id, status, error
* Atualizar usuarios.last\_inbound\_at quando receber mensagem do cliente (abre janela de 24h)
* Telinha de auditoria básica (tabela de notificações)

**M4 — Observabilidade & Segurança**

* Índices no DB (FKs e consultas mais usadas)
* Rate limit básico em auth e /notify/\*
* Sanitização/validação (e.g., Joi/zod) nos payloads críticos
* Rotacionar segredos: WA\_TOKEN, SMTP\_PASS

**M5 — Deploy/Infra**

* **Backend** via systemd (agendamentos-backend.service) com EnvironmentFile=.env
* **Frontend** build → /var/www/agendamentos (Nginx)
* Nginx: SPA + proxy /api/\* → 127.0.0.1:3002; webhook em HTTPS
* HTTPS com Let’s Encrypt (certbot)
* Firewall: abrir 80/443, não expor 3002
* Script de deploy: git pull + npm ci + restart + build front + rsync dist/ + reload Nginx

**Diagramas**

**1) Mind map (visão do sistema)**

mindmap

root((Agendamentos Online — MVP))

Frontend (SPA)

Auth/Login

Catálogo (Estabelecimentos/Serviços)

Slots (seleção)

Agendar

Meus agendamentos / Cancelar

Backend (API Express)

Auth (JWT)

Estabelecimentos/Serviços

Slots

Agendamentos

Notificações

Webhooks (WhatsApp)

Banco (MariaDB)

usuarios

estabelecimentos

servicos

slots

agendamentos

notifications

webhook\_events

Integrações

SMTP (e‑mail)

WhatsApp Cloud API

Template (fora 24h)

Texto (janela 24h)

Infra

Nginx (SPA + proxy)

systemd (backend)

VPS/HTTPS

**2) Sequência: criar agendamento + notificações**

sequenceDiagram

actor Cliente

participant FE as Frontend

participant API as Backend (Express)

participant DB as MariaDB

participant SMTP as Email (SMTP)

participant WA as WhatsApp API

Cliente->>FE: Seleciona serviço/slot e confirma

FE->>API: POST /agendamentos (JWT)

API->>DB: BEGIN; INSERT agendamento; COMMIT

API-->>FE: { ok, id, quando }

API->>SMTP: sendMail(confirmação)

API->>WA: messages(template ou text)

WA-->>API: { message\_id }

WA-->>API: Webhook (status: sent/delivered/read/failed)

API->>DB: INSERT notifications(status, provider\_id)

**Extras (futuro)**

* Lembretes automáticos (T‑24h / T‑1h) com fila e retries
* Painel Estabelecimento com filtros e KPIs (ocupação, no‑shows)
* Templates WhatsApp adicionais: lembrete, cancelamento, reagendamento