**Agendamentos Online — MVP (Mapa mental)**

**1) Núcleo do produto**

* **Objetivo:** permitir que clientes façam agendamentos online com estabelecimentos.
* **Perfis:**
  + **Cliente** (marca/consulta/cancela; recebe confirmações e lembretes)
  + **Estabelecimento** (configura serviços/slots; vê/cancela agendamentos)
* **Domínios:**
  + **Estabelecimentos** → Serviços → **Slots** (disponibilidade) → **Agendamentos**
  + **Notificações:** E‑mail (SMTP) e WhatsApp (Cloud API)

**2) Fluxos principais**

* **Autenticação**
  + POST /auth/register → cria usuário (bcrypt)
  + POST /auth/login → valida credenciais → JWT (7d)
  + GET /auth/me (protegida)
* **Catálogo e disponibilidade**
  + GET /establishments (ou /estabelecimentos) | GET /servicos (por estabelecimento)
  + GET /slots?establishmentId=...&weekStart=YYYY-MM-DD
  + POST /slots/toggle (admin/estabelecimento)
* **Agendamento**
  + POST /agendamentos (protegida) → cria registro → **dispara notificações**
  + GET /agendamentos (meus) | GET /agendamentos/estabelecimento
  + PUT /agendamentos/:id/cancel
* **Notificações**
  + **E‑mail**: SMTP (Gmail app password)
  + **WhatsApp**:
    - Endpoint oficial: POST /{PHONE\_NUMBER\_ID}/messages
    - **Template** (ex.: hello\_world) entrega fora da janela de 24h
    - **Texto livre** só dentro da janela de 24h (cliente enviou msg nas últimas 24h)
  + Rotas de teste: /notify/whatsapp/text, /notify/whatsapp/template
  + **Webhook**: POST /webhooks/whatsapp (statuses: sent/delivered/read/failed; inbound messages)

**3) Arquitetura (alto nível)**

* **Frontend (SPA)**
  + Stack: Vite + JS/React (SPA)
  + Porta dev: **3001**
  + Config API: VITE\_API\_URL (produção: /api via Nginx)
  + Util Api (fetch): monta BASE por env/DEV e injeta Authorization (exceto login/register)
* **Backend (API Node/Express, ESM)**
  + Porta: **3002**
  + Middlewares: express.json(), CORS (3001 em dev), morgan (logs)
  + Routers: auth, servicos, agendamentos, slots, estabelecimentos, notifications, notify, webhooks/whatsapp
  + Auth: JWT (middleware auth), bcrypt, validação de envs
  + Config: suporta variáveis MYSQL\_\* (ou DB\_\*) via config.js
  + DB: mysql2/promise pool
* **Banco de Dados (MariaDB/MySQL)**
  + **Local dev:** túnel SSH 127.0.0.1:3307 → VPS:3306
  + **VPS prod:** 127.0.0.1:3306
* **Integrações**
  + **WhatsApp Cloud API**: WA\_PHONE\_NUMBER\_ID, WA\_TOKEN, WA\_API\_VERSION; allowed list em modo de teste; webhook
  + **SMTP**: nodemailer com senha de app do Gmail

**4) Modelo de dados (conceitual)**

* **usuarios**(id, nome, email, telefone, **senha\_hash**, tipo[cliente|estabelecimento], created\_at)
* **estabelecimentos**(id, nome, contato, …)
* **servicos**(id, estabelecimento\_id, nome, duracao\_min, preco, ativo)
* **slots**(id, estabelecimento\_id, datetime, disponivel)
* **agendamentos**(id, estabelecimento\_id, servico\_id, cliente\_nome, cliente\_email, cliente\_telefone, inicio, status[ativo|cancelado], created\_at)
* **notifications**(id, agendamento\_id, canal[email|whatsapp], to, payload\_json, provider\_message\_id, status, created\_at)
* **webhook\_events**(id, provider, payload\_json, created\_at) — opcional p/ auditoria

**5) Regras de negócio**

* **Reserva**: exige slot disponível; grava horário em timezone do estabelecimento (ex.: America/Sao\_Paulo)
* **Cancelamento**: atualiza status e (opcional) libera slot
* **Notificações**: enviadas pós‑commit; falha não impede criação do agendamento; registrar status
* **WhatsApp**:
  + Se **fora** da janela de 24h → usar **template** aprovado (com variáveis)
  + Se **dentro** da janela → pode enviar **texto** livre
  + Webhook atualiza notifications.status e usuarios.last\_inbound\_at

**6) Configuração / .env (produção)**

* **API**: PORT, JWT\_SECRET
* **DB**: MYSQL\_HOST, MYSQL\_PORT, MYSQL\_USER, MYSQL\_PASSWORD, MYSQL\_DATABASE
* **WhatsApp**: WA\_API\_VERSION, WA\_PHONE\_NUMBER\_ID, WA\_TOKEN, WHATSAPP\_ALLOWED\_LIST
* **Email**: SMTP\_HOST, SMTP\_PORT, SMTP\_USER, SMTP\_PASS, EMAIL\_FROM
* **Frontend build**: VITE\_API\_URL=/api

**7) Deploy & Infra**

* **VPS (Ubuntu/Debian)**
  + Backend: systemd service (/etc/systemd/system/agendamentos-backend.service)
  + Frontend: build SPA → /var/www/agendamentos (Nginx)
  + **Nginx**: serve SPA e proxy /api/\* → 127.0.0.1:3002/\*; webhook em HTTPS
  + **HTTPS**: certbot/Let’s Encrypt
  + **Firewall**: abrir 80/443; **não** expor 3002
* **Deploy (pull & restart)**
  + git pull --rebase origin main
  + Backend: npm ci → systemctl restart agendamentos-backend
  + Frontend: npm ci && npm run build → rsync dist/ → /var/www/agendamentos/ → systemctl reload nginx

**8) Observabilidade & diagnósticos**

* Logs: morgan dev, erros com prefixos ([auth/login], [notify/whatsapp])
* Testes rápidos:
  + GET /health / /api/health
  + DB: script scripts/test-db.mjs (PING, DB\_INFO)
  + WhatsApp: cURL direto no Graph → comparar com backend
* Métricas (futuro): tempos de resposta, taxa de erro por rota; fila de notificações

**9) Segurança**

* Nunca commitar .env; rotation de WA\_TOKEN e senhas de app
* Restringir MySQL a 127.0.0.1; acesso remoto só via túnel
* Bcrypt com custo adequado; JWT com segredo forte; expiração 7d
* CORS restrito em prod (mesmo domínio)
* Validação/sanitização de payloads; evitar senha em claro

**10) Roadmap curto**

* **Templates WhatsApp com variáveis** (confirmação e lembretes)
* **Webhook persistente** (statuses e inbound → abre janela 24h)
* **Scheduler** (lembrete T‑24h/T‑1h) com fila e retries
* **Painel do estabelecimento** (agenda, filtros, status)
* **Relatórios** (no‑shows, ocupação por serviço)
* **Rate limiting** / CAPTCHA em auth
* **Teste E2E** (login → agendar → notificar → cancelar)

**11) Mapa de rotas (resumo)**

* **Públicas**: GET /health, GET /, POST /auth/register, POST /auth/login
* **Protegidas (JWT)**: GET /auth/me, GET /servicos, GET /slots, POST /agendamentos, PUT /agendamentos/:id/cancel, listagens
* **Admin/Estab**: POST /slots/toggle, CRUD de serviços
* **Notificações**: POST /notify/whatsapp/text, POST /notify/whatsapp/template
* **Webhook**: GET /webhooks/whatsapp (verify), POST /webhooks/whatsapp (eventos)