Planejamento Completo do VIPme2

# Visão Geral do Produto

- Modelo: app premium de compra única (sem anúncios)

- Offline-first: tudo funciona sem internet (Room + DataStore)

- Catálogo público: site estático (GitHub Pages/Cloudflare Pages/Netlify)

- Licença: JWT assinado (validação offline) + revalidação eventual online

- Trial: 7 dias; depois bloqueio com CTA 'Comprar agora' ou 'Restaurar compra'

- Sincronização: eventual, idempotente, segura (WorkManager + endpoints simples)

- Internacionalização: 20+ idiomas (Noto Sans; RTL)

- Mensageiros: intents diretas + fallback web

- CI/CD: GitHub Actions (app e site), canário + rollback

# 1) Design System

Cores: Azul petróleo escuro (#0D3B4F), Azul claro (#4A90E2), Dourado (#F5A623), Fundo claro (#F8F9FA), Branco (#FFFFFF)

Texto: #1F1F1F / #707070, Erro: #E53935, Sucesso: #43A047

Tipografia: Noto Sans (global), títulos 22–24sp Bold, subtítulos 18sp SemiBold, corpo 14sp Regular

Componentes: Botões arredondados, cards com sombra leve, campos 48dp

Acessibilidade: contraste, TalkBack, RTL, toque 48dp

# 2) Fluxos do Usuário

Dashboard (atalhos), Produtos (CRUD+share), Clientes (CRUD+fidelidade), Vendas (carrinho, recibo), Relatórios (filtros/export), Estoque&Caixa, Loja Online, Configurações

# 3) Wireframes & Diagrama

Wireframes simples (ASCII) e diagrama de navegação gerado.

# 4) Catálogo Online

JSON exportado pelo app, estrutura por locale (pt-BR, en, etc)

HTML responsivo com grid de cards

Cache com ETag + fallback

Expiração automática em 15 dias sem sync

# 5) Licenciamento Premium

Compra única via Play Store (productId = vipme2.premium)

JWT assinado ES256 validado offline

Trial de 7 dias → tela boas-vindas → bloqueio → Premium confirmado

Revalidação eventual 7–14 dias

Telas wireframes: boas-vindas, bloqueio, confirmado

# 6) Sincronização

Room = fonte de verdade

Fila pending\_ops + WorkManager com retries

POST /sync/upload (batch, idempotência), GET /sync/download (delta)

Resolução: Last-Write-Wins + soma de deltas

ETag para catálogo

Segurança: TLS, token, rate-limit

# 7) CI/CD e Deploy

Repositório GitHub com /app (Android) e /catalog (web)

Workflows: android-ci.yml (build/test/AAB), web-deploy.yml (validação + Pages)

Deploy canário em staging; produção em main

Rollback por commit revert

# 8) Dados Locais

Room Entities: Product, Customer, Sale, StockMovement, CashEntry

Extras: pending\_ops, sync\_state

DataStore: configs (idioma, moeda, loja, cliente default), licença, lastSyncWeb

# 9) Integração com Mensageiros

WhatsApp, Telegram, Signal, LINE, KakaoTalk, WeChat, Zalo, Viber

Deep links com fallback web

# 10) Escalabilidade & Performance

Paging 3 + Room, Coil (cache LRU + disco), WebP nas imagens, WorkManager para jobs pesados

# 11) Testes & Monitoramento

Unitários, instrumentados, E2E offline, performance em listas grandes, acessibilidade

Métricas: license\_activated, trial\_started, trial\_expired, sale\_completed, etc

# 12) Segurança

JWT ES256 com chave pública no app

TLS, tokens com escopo

Backup/export criptografado

Play Integrity opcional

Página web sem PII, expira em 15 dias

# 13) Requisitos Legais

Compra única clara, política de reembolso Play Store, moeda local, política de privacidade, acessibilidade básica

# 14) Checklist QA/PR

App offline, trial e bloqueio, compra/restauração, listas grandes OK, export CSV/PDF OK, intents mensageiros OK, catálogo expira, acessibilidade verificada

# 15) Roadmap

Semanas 1-2: infra app, CRUD básico, trial

Semanas 3-4: vendas, recibos, relatórios

Semanas 5-6: licença Play Billing, sync

Semanas 7-8: polimento, i18n, métricas, CI/CD, publicação

# 16) Ações imediatas

Fechar schema local (Room + DataStore)

Implementar trial

Montar catálogo web em staging

Subir CI Android + Web (Actions)