

# PSA Time – Documentação Técnica

**Versão:** 1.1 (Atualizada) **Plataforma:** Android (Nativo) **Linguagem:** Kotlin **Arquitetura:** MVVM (Model-View-ViewModel)

## 1. Visão Geral do Produto

O **PSA Time** é um aplicativo de saúde preventiva focado no monitoramento dos níveis de **PSA (Antígeno Prostático Específico)**. Diferente de simples anotadores de exames, o PSA Time implementa um algoritmo de inteligência clínica que analisa não apenas o valor absoluto, mas a **variação (velocidade de aumento)** do PSA ao longo do tempo para emitir alertas de risco (Verde, Amarelo, Vermelho).

### Objetivos Principais

1. Armazenar histórico de exames de forma persistente e segura.
2. Calcular deltas de variação com precisão decimal (2 casas) entre exames.
3. Alertar o usuário proativamente sobre a necessidade de novos exames (Notificações Inteligentes).
4. Suportar múltiplos idiomas (PT-BR e EN-US).
5. Garantir a segurança dos dados via Backup/Restore local (JSON).

## 2. Stack Tecnológico

O projeto foi construído utilizando as práticas modernas de desenvolvimento Android (Modern Android Development - MAD):

- **Linguagem:** Kotlin.
- **UI Toolkit:** XML Layouts com Material Design 3 (M3).
- **Arquitetura:** MVVM (Separação clara entre UI, Lógica e Dados).
- **Banco de Dados:** Room (SQLite abstraction layer).
- **Navegação:** Jetpack Navigation Component.
- **Assincronismo:** Kotlin Coroutines & Flow.
- **Internacionalização:** Android Resource System (`values-pt`, `values-en`) e `AppCompatDelegate`.
- **Agendamento:** AlarmManager & BroadcastReceiver.

### 3. Arquitetura da Solução

O app segue o padrão **MVVM**, garantindo que a lógica de negócios não esteja acoplada à interface do usuário.

#### Estrutura de Pacotes

- `data` : Entidades do Banco de Dados (Room) e DAOs.
- `ui` : Fragmentos, Activities, ViewModels e Adapters.
- `receiver` : Lógica de agendamento e disparo de notificações.

#### Fluxo de Dados

1. **View (Fragment)**: Observa o `LiveData` do ViewModel.
2. **ViewModel**: Processa regras de negócio (cálculo de cores/status) e chama o DAO.
3. **Repository/DAO**: Executa operações de I/O no Banco de Dados SQLite.
4. **Database**: Fonte da verdade.

### 4. Algoritmos e Regras de Negócio (Core)

O coração do PSA Time é a lógica de determinação de status ( `PsaStatus` ).

#### 4.1. Classificação de Risco (Cores)

A lógica reside em `PsaViewModel` e `PsaHistoryAdapter`.

1.  **Nível Normal (Verde):**
  - Variação anual menor que `0.40 ng/mL`.
  - Primeiro exame com valor absoluto `<= 4 ng/mL`.
2.  **Atenção (Amarelo):**
  - **Regra de Variação:** Aumento anual `>= 0.40 ng/mL`.
  - **Regra de Valor Inicial:** Primeiro registro inserido é `> 4 ng/mL` (mas `<= 10 ng/mL`).
  - **Ação:** Indica necessidade de repetição do exame em 3 meses (Follow-up curto).
3.  **Ação Imediata (Vermelho):**

- **Regra de Valor Absoluto:** Primeiro registro inserido é `> 10 ng/mL`.

- **Regra de 3 Pontos (Confirmação de Risco):**

1. Exame anterior já apresentou salto `>= 0.40` (Era Amarelo).
2. Novo exame realizado em curto período (< 120 dias).
3. Novo valor manteve-se alto ou subiu em relação ao anterior.

## 4.2. Tratamento de Visualização (Adapter)

- **Precisão:** Cálculos de delta exibidos com 2 casas decimais (`.2f`) para evitar arredondamentos enganosos.
- **Highlight do Ano:** A data é exibida no formato `AAAA | DD de MMM, AAAA`.
- **Negrito Inteligente:** O valor numérico (`2.5`) é renderizado em negrito e fonte maior, enquanto a unidade (`ng/mL`) permanece em fonte normal e menor.

## 5. Sistema de Notificações Inteligentes

O app atua proativamente para engajar o usuário através de agendamentos via `AlarmManager`.

### 5.1. Tipos de Notificação

1. **Novembro Azul (Fixo):** Dispara todo dia **1º de Novembro às 09:00**, lembrando da campanha mundial.
2. **Follow-up Anual (Dinâmico):** Calcula 1 ano (365 dias) após a data do **último exame registrado** e agenda um lembrete para manter o delta atualizado.
3. **Confirmação de Alerta Amarelo (Crítico):**
  - **Gatilho:** Se o último exame gerou um status **Amarelo** (variação alta).
  - **Ação:** Agenda automaticamente um lembrete para **3 meses** depois.
  - **Cancelamento Inteligente:** Se o usuário inserir um novo exame antes do prazo (seja verde ou vermelho), esse alarme específico é cancelado automaticamente para não gerar ruído.

## 6. Internacionalização (i18n)

O app está totalmente localizado para dois idiomas, com troca dinâmica nas configurações sem necessidade de reiniciar o app.

- **Idiomas Suportados:**

- Inglês (EUA) - Padrão (`values/strings.xml`)

- Português (Brasil) - ( `values-pt/strings.xml` )
- **Mecanismo:** Utiliza `AppCompatDelegate.setApplicationLocales()` para forçar a troca de idioma em tempo de execução.

## 7. Persistência e Backup

### 7.1. Banco de Dados Local (Room)

- Tabela: `psa_results`
- Colunas: `id` (PK), `year` (Int), `value` (Float), `timestamp` (Long).

### 7.2. Backup e Restore (JSON)

Implementado via **Storage Access Framework (SAF)**.

- **Exportar:** Gera arquivo `.json` com timestamp no nome.
- **Importar:** Lê, valida e restaura o banco de dados.

## 8. Interface e Design System

Design inspirado na interface "New UI" do Android Studio (Otter 2025).

- **Temas Dinâmicos:** Suporte completo a **Light Mode** e **Dark Mode**.
- **Paleta de Cores Customizada:**
  - *Primary:* Azul Studio ( `#3574F0` ).
  - *Surface:* Branco Puro (Light) e Cinza Profundo `#1E1F22` (Dark).
  - *Controles:* Componentes nativos ( `MaterialRadioButton` ) forçados a usar a cor primária, eliminando cores de acento padrão (lilás) do Android.
- **Layout:** Uso de `MaterialCardView` sem bordas e `NestedScrollView` para rolagem fluida.

## 9. Apoio e Doação

Integração nativa para apoio ao desenvolvedor via Bitcoin.

- **Funcionalidade:** Clique no QR Code ou endereço copia para o clipboard e tenta abrir o app de carteira instalado via `Intent` com URI `bitcoin:<addr>?message=....`

**Desenvolvido por:** Fabio Ferrante

**Gerado com apoio de:** Gemini AI Assistant