

Health Monitor

Documentação Técnica

Victor Hugo Monteiro da Silva

Murilo de Melo

15 de dezembro de 2025

Sumário

1	Visão Geral	2
2	Estrutura do Projeto	2
3	Arquitetura MVVM	2
3.1	Fluxo de Dados	2
4	Telas do Aplicativo	3
5	Modelo de Dados	3
5.1	Entidade Principal	3
5.2	Enumerações	3
6	Fórmulas Implementadas	4
6.1	Índice de Massa Corporal (IMC)	4
6.2	Taxa Metabólica Basal (Mifflin-St Jeor)	4
6.3	Peso Ideal (Fórmula de Devine)	4
6.4	Necessidade Calórica Diária	4
6.5	Percentual de Gordura Corporal (US Navy)	5
7	Componentes UI	5
8	Design System	5
8.1	Paleta de Cores	5
8.2	Tipografia	5
9	Tecnologias Utilizadas	6
10	Validação de Campos	6
11	Uso de LLMs no Desenvolvimento	6

1 Visão Geral

Aplicativo Android desenvolvido em Kotlin com Jetpack Compose para monitoramento de saúde. Oferece cálculos de IMC, Taxa Metabólica Basal, Necessidade Calórica Diária e Percentual de Gordura Corporal.

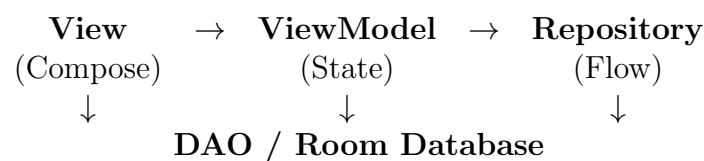
2 Estrutura do Projeto

```
1 app/src/main/java/com/example/calculadoraimc/
2 |-- data/
3 |   |-- local/
4 |     |-- AppDatabase.kt
5 |     |-- dao/IMCHistoryDao.kt
6 |     |-- entity/IMCHistory.kt
7 |     |-- converter/Converters.kt
8 |   |-- repository/
9 |     |-- IMCHistoryRepository.kt
10 |-- datasource/
11 |   |-- Calculations.kt
12 |-- feature/
13 |   |-- home/
14 |   |-- history/
15 |   |-- detail/
16 |   |-- help/
17 |-- ui/theme/
18 |   |-- Color.kt
19 |   |-- Theme.kt
20 |   |-- Type.kt
```

Estrutura de Diretórios

3 Arquitetura MVVM

O projeto segue o padrão MVVM (Model-View-ViewModel) com Repository Pattern:



3.1 Fluxo de Dados

1. Usuário interage com a UI (Compose)
2. ViewModel processa a ação e atualiza o estado
3. Repository faz a mediação com o banco de dados
4. Room persiste/recupera os dados
5. Flow propaga as mudanças de volta para a UI

4 Telas do Aplicativo

Tela	Descrição	ViewModel
Home	Entrada de dados e resultados	HomeViewModel
History	Lista de registros com gráfico	HistoryViewModel
Detail	Detalhes de um registro	DetailViewModel
Help	Explicação das fórmulas	-

Tabela 1: Telas e seus ViewModels

5 Modelo de Dados

5.1 Entidade Principal

```

1 @Entity(tableName = "imc_history")
2 data class IMCHistory(
3     @PrimaryKey(autoGenerate = true) val id: Int = 0,
4     val date: Date,
5     val weight: Double,
6     val height: Int,
7     val imc: Double,
8     val imcClassification: String,
9     val age: Int,
10    val gender: Gender?,
11    val bmr: Int?,
12    val idealWeight: String?,
13    val dailyCaloricNeed: Int?,
14    val bodyFat: Float?
15 )

```

IMCHistory.kt

5.2 Enumerações

```

1 enum class Gender { MALE, FEMALE }
2
3 enum class ActivityLevel(val displayName: String, val factor: Double) {
4     SEDENTARY("Sedentario", 1.2),
5     LIGHTLY_ACTIVE("Levemente Ativo", 1.375),
6     MODERATELY_ACTIVE("Moderadamente Ativo", 1.55),
7     VERY_ACTIVE("Muito Ativo", 1.725),
8     EXTRA_ACTIVE("Extremamente Ativo", 1.9)
9 }

```

Enums

6 Fórmulas Implementadas

6.1 Índice de Massa Corporal (IMC)

$$IMC = \frac{peso \text{ (kg)}}{altura^2 \text{ (m)}} \quad (1)$$

Faixa	Classificação
< 18.5	Abaixo do peso
18.5 – 24.9	Peso normal
25.0 – 29.9	Sobrepeso
30.0 – 34.9	Obesidade Grau I
35.0 – 39.9	Obesidade Grau II
≥ 40.0	Obesidade Grau III

Tabela 2: Classificação do IMC

6.2 Taxa Metabólica Basal (Mifflin-St Jeor)

Homens:

$$TMB = (10 \times peso) + (6.25 \times altura) - (5 \times idade) + 5 \quad (2)$$

Mulheres:

$$TMB = (10 \times peso) + (6.25 \times altura) - (5 \times idade) - 161 \quad (3)$$

6.3 Peso Ideal (Fórmula de Devine)

Homens:

$$PI = 50 + 2.3 \times (altura_{pol} - 60) \quad (4)$$

Mulheres:

$$PI = 45.5 + 2.3 \times (altura_{pol} - 60) \quad (5)$$

6.4 Necessidade Calórica Diária

$$NCD = TMB \times Fator_{atividade} \quad (6)$$

Nível de Atividade	Fator
Sedentário	1.2
Levemente ativo	1.375
Moderadamente ativo	1.55
Muito ativo	1.725
Extremamente ativo	1.9

Tabela 3: Fatores de Atividade

6.5 Percentual de Gordura Corporal (US Navy)

Homens:

$$GC = \frac{495}{1.0324 - 0.19077 \times \log_{10}(cintura - pescoco) + 0.15456 \times \log_{10}(altura)} - 450 \quad (7)$$

Mulheres:

$$GC = \frac{495}{1.29579 - 0.35004 \times \log_{10}(cintura + quadril - pescoco) + 0.22100 \times \log_{10}(altura)} - 450 \quad (8)$$

7 Componentes UI

Componente	Função
MainCard	Card principal com IMC e métricas
MetricCard	Cards menores para métricas individuais
IMCGraphic	Gráfico circular animado do IMC
IMCHistoryChart	Gráfico de evolução (Vico)
IMCCalculatorContainer	Formulário de entrada
IconTag	Ícone circular reutilizável

Tabela 4: Componentes UI Principais

8 Design System

8.1 Paleta de Cores

Cor	Hex	Uso
Primary Dark	#312E81	Fundo escuro
Accent Color	#7C3AED	Cor principal
Green Level	#10B981	Status normal
Orange Level	#F59E0B	Status atenção
Red Level	#EF4444	Status crítico

Tabela 5: Paleta de Cores (baseada em Tailwind CSS)

8.2 Tipografia

- **Fonte:** IBM Plex Sans
- **Pesos:** ExtraLight, Thin, Light, Regular, Medium, SemiBold, Bold

Tecnologia	Uso
Kotlin	Linguagem principal
Jetpack Compose	Interface declarativa
Room	Persistência local (SQLite)
Navigation Compose	Navegação entre telas
Vico	Gráficos de evolução
Material 3	Design system

Tabela 6: Stack Tecnológico

9 Tecnologias Utilizadas

10 Validação de Campos

Campos obrigatórios:

- Altura, Peso, Idade
- Gênero, Nível de Atividade
- Cintura, Pescoço

Campo condicional:

- Quadril (obrigatório apenas para mulheres)

11 Uso de LLMs no Desenvolvimento

Ferramenta	Uso
Claude	Lógica de cálculos, arquitetura, Tarefas gerais
Gemini	Componentes UI, temas, validações
ChatGPT	Documentação, boilerplate

Tabela 7: Ferramentas de IA utilizadas