

**INSTITUTO FEDERAL
DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA**
Triângulo Mineiro

MATÉRIA: PDM

BALANCEI

Alunos: Carlos Alexandre
Eder Junior
Luiz Folador
Nícollas Ferreira

O PROBLEMA

O PROBLEMA

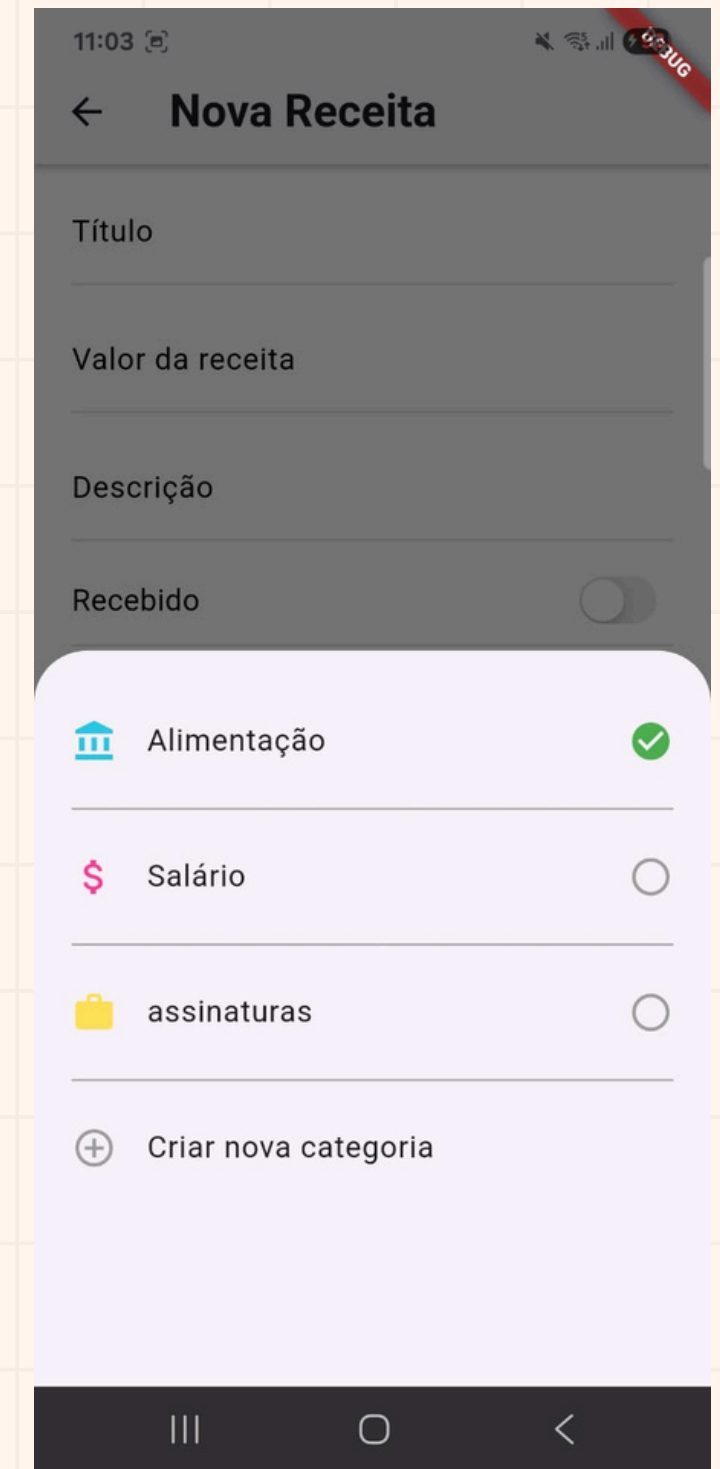
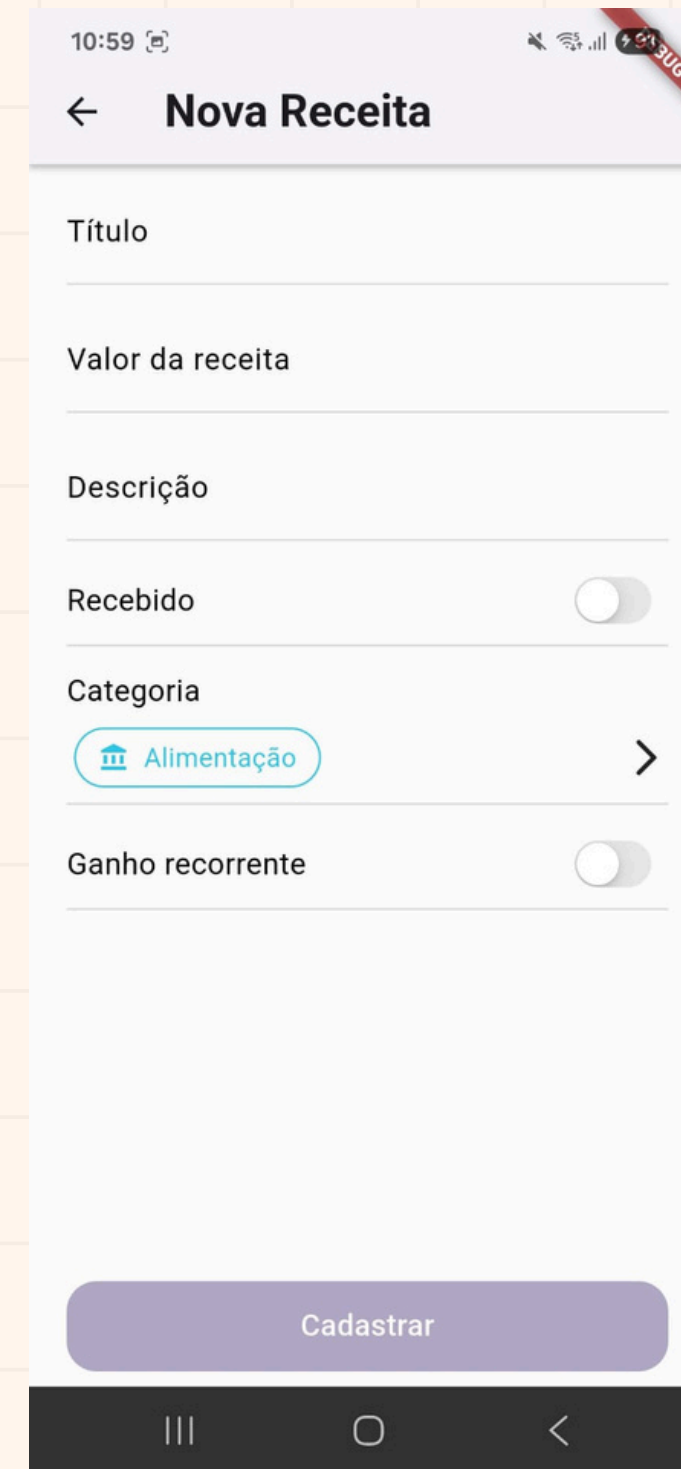
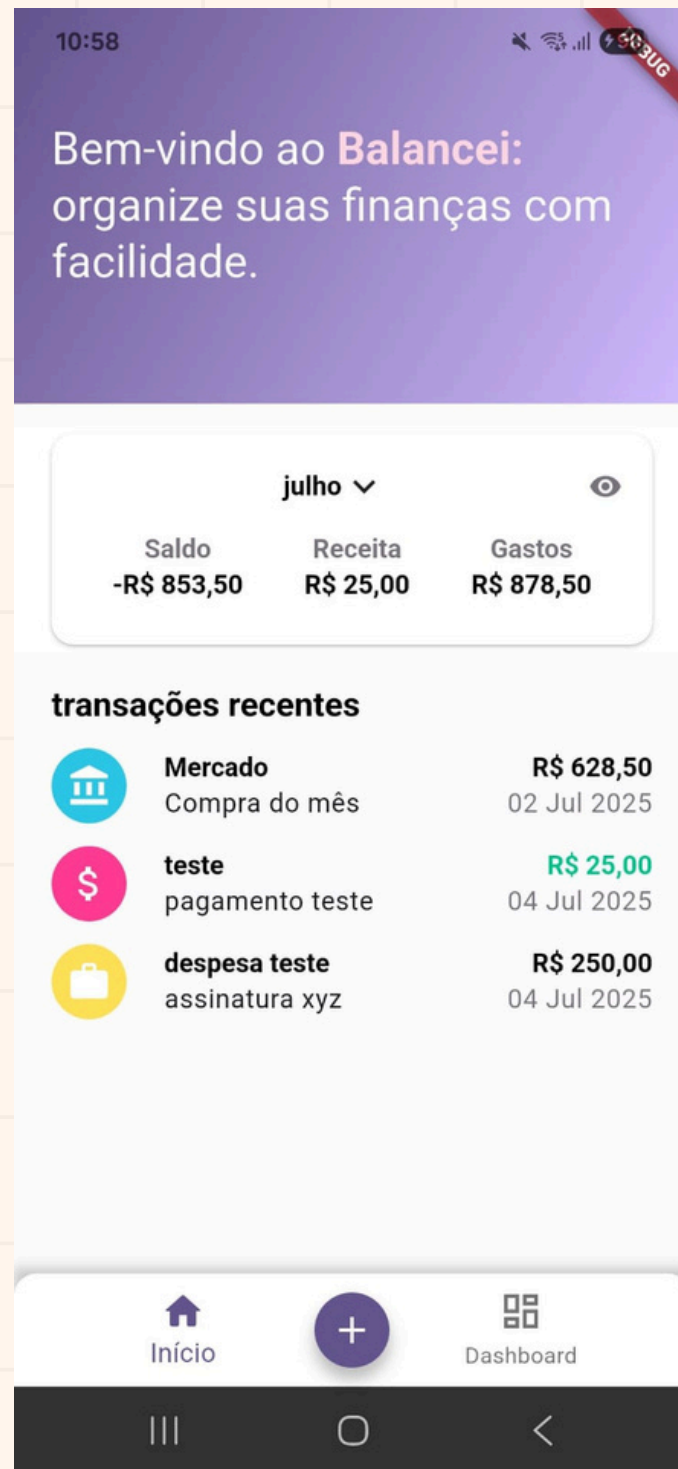
 **DESAFIO:** GESTÃO FINANCEIRA PESSOAL

 **PROBLEMAS** IDENTIFICADOS:

- APPS COMPLEXOS DEMAIS PARA USO COTIDIANO
- DEPENDÊNCIA DE INTERNET/CLOUD
- PREOCUPAÇÕES COM PRIVACIDADE DE DADOS
- FALTA DE SOLUÇÕES SIMPLES E GRATUITAS

DEMO

TELAS



TELAS

11:03

93 JUL

←

Criar Categoria

Descrição da categoria

Color

✓

Outros

Icon

🏠

\$

\$

Outros

Criar

11:04

94 JUL

←

Criar Categoria

Descrição da categoria

Freela

Color

✓

Outros

Icon

\$

🏠

\$

Outros

Criar

11:05

94 JUL

←

Nova Receita

Título

salário freela

Valor da receita

R\$ 1.500,00

Descrição

salario freelancer |

Recebido

Categoria

\$ Freela

>

Ganho recorrente

Cadastrar

11:10

96 JUL

Bem-vindo ao **Balancei**:
organize suas finanças com
facilidade.

julho ▾

🗖

Saldo

Receita

Gastos

transações recentes

🏠

Mercado
Compra do mês

02 Jul 2025

\$

teste
pagamento teste

04 Jul 2025

👛

despesa teste
assinatura xyz

04 Jul 2025

\$

salário freela
salario freelancer

04 Jul 2025

🏠

+

🗖

Início

Dashboard




ARQUITETURA TÉCNICA

CLEAN ARCHITECTURE + MVVM

- **UI LAYER**
 - SCREENS ▪ VIEWMODELS ▪ COMPONENTS
- **DOMAIN LAYER**
 - ENTITIES ▪ DTOS (DATA TRANSFER OBJECTS) ▪ VALIDATIONS
- **DATA LAYER**
 - DATABASE ▪ DAOS (DATA ACCESS OBJECTS) ▪ REPOSITORIES
- **DTOS** TRANSPORTAM DADOS ENTRE CAMADAS SEM LÓGICA DE NEGÓCIO.
- **DAOS** ENCAPSULAM ACESSO DIRETO AO BANCO SQLITE.

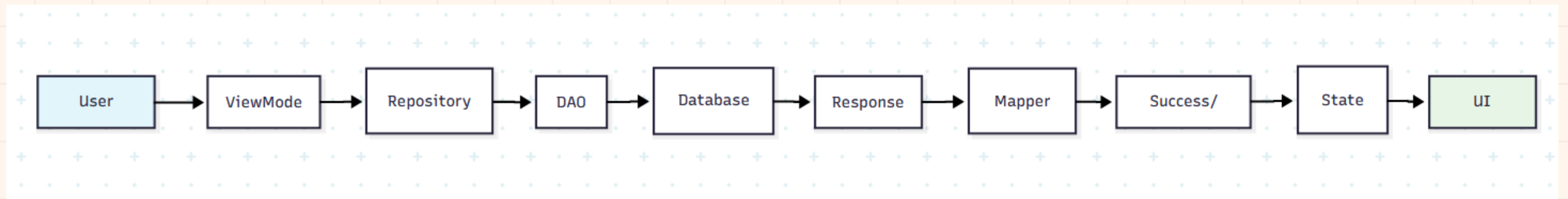
STACK TECNOLÓGICO

TECNOLOGIAS UTILIZADAS

-  FRONTEND:
 - FLUTTER 3.0+ (UI FRAMEWORK)
 - MATERIAL DESIGN 3 (DESIGN SYSTEM)
 - GOROUTER (NAVEGAÇÃO TYPE-SAFE)
-  ESTADO:
 - RIVERPOD (STATE MANAGEMENT)
 - FREEZED (DATA CLASSES)
 - RESULT DART (ERROR HANDLING)
-  PERSISTÊNCIA:
 - DRIFT (SQLITE ORM)
 - PATH PROVIDER (STORAGE)

FLUXO DE DADOS

DATA FLOW PATTERN



CÓDIGO EXEMPLO:

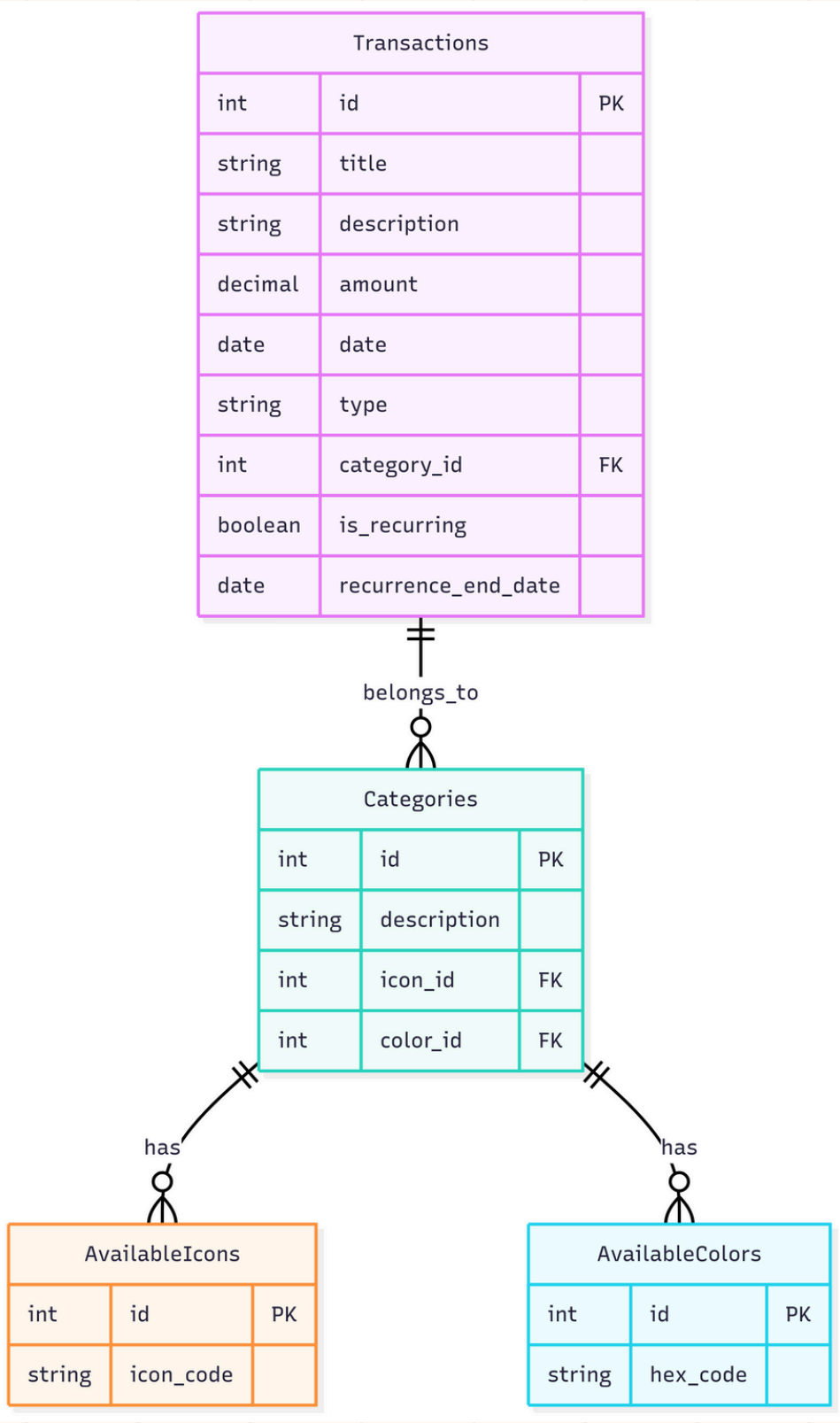
```
// HomeViewModel
Future<void> fetchTransfers() async {
  state = state.copyWith(summary: AsyncValue.loading());
  final result = await _transferRepository.getFinancialSummary(
    dateFilter: state.selectedDateFilter,
  );

  state = result.fold(
    (success) => state.copyWith(summary: AsyncValue.data(success)),
    (failure) => state.copyWith(summary: AsyncValue.error(failure)),
  );
}
```

ESTRUTURA DO BANCO



DATABASE SCHEMA (SQLITE)



PADRÕES DE CÓDIGO

CÓDIGO EXEMPLO:

```
// Repository Pattern
abstract interface class TransferRepository {
    AsyncResult<List<TransactionEntity>> getTransfers({DateFilterEntity?
filter});
    AsyncResult<Unit> addTransfer(TransferDTO transfer);
    AsyncResult<FinancialSummaryEntity> getFinancialSummary({required
DateFilterEntity filter});
}

// Validation Pattern
class TransferValidator extends BaseValidator<TransferDTO> {
    @override
    ValidationResult validate(TransferDTO dto) {
        return LucidValidation.phrase(
            .requiredField('title', dto.title)
            .greaterThan('amount', dto.amount, 0)
            .validate();
        )
    }
}
```

IMPLEMENTAMOS REPOSITORY PATTERN PARA ABSTRAÇÃO DE DADOS E UM SISTEMA DE VALIDAÇÃO ROBUSTO. TODA REGRA DE NEGÓCIO FICA ISOLADA NA DOMAIN LAYER.

PRINCIPAIS DESAFIOS







🚧 DESAFIOS TÉCNICOS SUPERADOS

- 🏗️ CONFIGURAÇÃO DO DRIFT
 - GERAÇÃO DE CÓDIGO AUTOMÁTICO
 - MIGRAÇÕES DE SCHEMA
 - TYPE-SAFETY COM SQL
- 🎨 DESIGN SYSTEM CONSISTENTE
 - MATERIAL DESIGN 3 IMPLEMENTATION
 - COMPONENTES REUTILIZÁVEIS
 - RESPONSABILIDADE
- ⚡ STATE MANAGEMENT COMPLEXO
 - ESTADOS ASSÍNCRONOS
 - CACHE DE DADOS
 - NAVEGAÇÃO ENTRE TELAS

ENFRENTAMOS DESAFIOS PRINCIPALMENTE NA CONFIGURAÇÃO INICIAL DO ORM E NA IMPLEMENTAÇÃO DE UM DESIGN SYSTEM CONSISTENTE. A CURVA DE APRENDIZADO DO RIVERPOD TAMBÉM FOI SIGNIFICATIVA.

FUNCIONALIDADES NÃO IMPLEMENTADAS




ROADMAP VS. REALIDADE

-  NÃO IMPLEMENTADAS (POR PRIORIZAÇÃO):
 -  GRÁFICOS E RELATÓRIOS (CHARTS_FLUTTER)
 -  CONTROLE DE INVESTIMENTOS
 -  INTEGRAÇÃO OPEN FINANCE
 -  ANÁLISES COM IA
-  FOCO NO MVP:
 - CORE FEATURES SÓLIDAS E TESTADAS
 - EXPERIÊNCIA OFFLINE COMPLETA
 - BASE ARQUITETURAL PARA EXPANSÕES

POR QUESTÕES DE TEMPO E COMPLEXIDADE, PRIORIZAMOS ENTREGAR FUNCIONALIDADES CORE BEM IMPLEMENTADAS AO INVÉS DE TENTAR ABRANGER TUDO.

LIÇÕES APRENDIDAS

KEY LEARNINGS

-  PLANEJAMENTO DE ESCOPO
 - MVP BEM DEFINIDO > FEATURES INCOMPLETAS
 - VALIDAÇÃO DE COMPLEXIDADE TÉCNICA
 - TIMEBOXING DE FUNCIONALIDADES
-  DESENVOLVIMENTO FLUTTER
 - STATE MANAGEMENT PATTERNS
 - DATABASE ORM INTEGRATION
 - TYPE-SAFE DEVELOPMENT
-  TRABALHO EM EQUIPE
 - GIT FLOW COLABORATIVO
 - CODE REVIEW PROCESS
 - DOCUMENTAÇÃO TÉCNICA

RESULTADOS ALCANÇADOS




✓ ENTREGÁVEIS CONCLUÍDOS

- 📱 APLICATIVO FUNCIONAL
 - 100% OFFLINE CAPABILITY
 - UI/UX POLIDA E RESPONSIVA
 - PERFORMANCE OTIMIZADA
- 🏗️ ARQUITETURA ROBUSTA
 - CLEAN ARCHITECTURE
 - DESIGN PATTERNS
- 📖 DOCUMENTAÇÃO COMPLETA
 - README TÉCNICO
 - ANÁLISE EXPECTATIVAS VS. REALIDADE
 - RELATÓRIO DE CRONOGRAMA

ENTREGAMOS UM PRODUTO FUNCIONAL, COM ARQUITETURA SÓLIDA E DOCUMENTAÇÃO COMPLETA.
O APP ESTÁ PRONTO PARA USO REAL E EXPANSÕES FUTURAS.

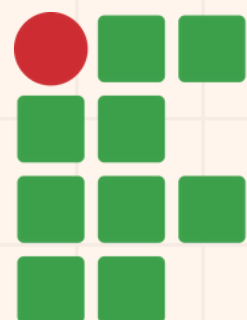
PRÓXIMOS PASSOS

ROADMAP FUTURO

-  VERSÃO 1.1 - VISUALIZAÇÕES
 - GRÁFICOS DE GASTOS POR CATEGORIA
 - RELATÓRIOS MENSAIS/ANUAIS
 - EXPORTAÇÃO DE DADOS
-  VERSÃO 1.2 - INVESTIMENTOS
 - CARTEIRA DE ATIVOS
 - ACOMPANHAMENTO DE RENTABILIDADE
 - METAS FINANCEIRAS
-  VERSÃO 2.0 - CLOUD INTEGRATION
 - SINCRONIZAÇÃO ENTRE DISPOSITIVOS
 - BACKUP AUTOMÁTICO
 - OPEN FINANCE INTEGRATION

TEMOS UM ROADMAP CLARO PARA EVOLUIR O APP.

A BASE SÓLIDA QUE CRIAMOS PERMITIRÁ IMPLEMENTAR ESSAS FUNCIONALIDADES DE FORMA INCREMENTAL E SUSTENTÁVEL.



**INSTITUTO FEDERAL
DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA**
Triângulo Mineiro

OBRIGADO