

[Description](#)

[Intended User](#)

[Features](#)

[User Interface Mocks](#)

[Login](#)

[Menu](#)

[Cadastro de contas \(poupança, conta corrente, cartão de crédito, dinheiro/carteira\)](#)

[Cadastro de categorias](#)

[Registro de receitas/despesas/transferência](#)

[Frases motivacionais sobre finanças após registros de receitas/despesas](#)

[Key Considerations](#)

[How will your app handle data persistence?](#)

[Describe any corner cases in the UX.](#)

[Describe any libraries you'll be using and share your reasoning for including them.](#)

[Describe how you will implement Google Play Services.](#)

[Next Steps: Required Tasks](#)

[Task 1: Project Setup](#)

[Task 2: Implement UI for Each Activity and Fragment](#)

[Task 2.1: Create layout for each activity and fragment](#)

[Task 3: Criar AsyncTask](#)

[Task 4: Criar DataProvider](#)

[Task 5: Criar Widget](#)

GitHub Username: [ricardosbarbosa](#)

Where Is My Money

Description

Aplicativo para ajudar ao usuário ter um melhor controle financeiros de suas despesas e receitas, sabendo onde seu dinheiro é gasto, e como ele está distribuído em suas contas bancárias.

Intended User

Pessoas que tenham dificuldade em saber como seu dinheiro é gasto, que desejem ter um controle de registros de despesas e receitas, para concluírem com o tempo como podem ter uma vida financeira mais saudável, sabendo encontrar pontos de gastos excessivo que precisam ser melhor controlado.

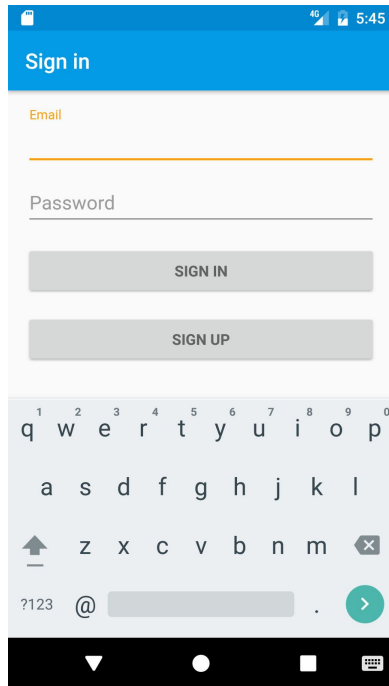
Features

Lista das principais features do app:

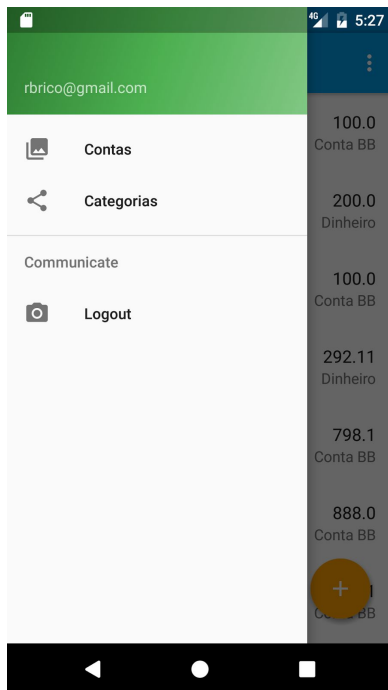
- Login
- Cadastro de usuário
- Cadastro de contas (poupança, conta corrente, cartão de crédito, dinheiro/carteira)
- Registro de despesas de uma categoria em uma determinada conta
- Registro de receitas de uma categoria em uma determinada conta
- Registro de transferência de valores de uma conta para outra, como saques da conta corrente, depósitos etc.
- Saldos das contas
- Frases motivacionais sobre finanças após registros de receitas/despesas

User Interface Mocks

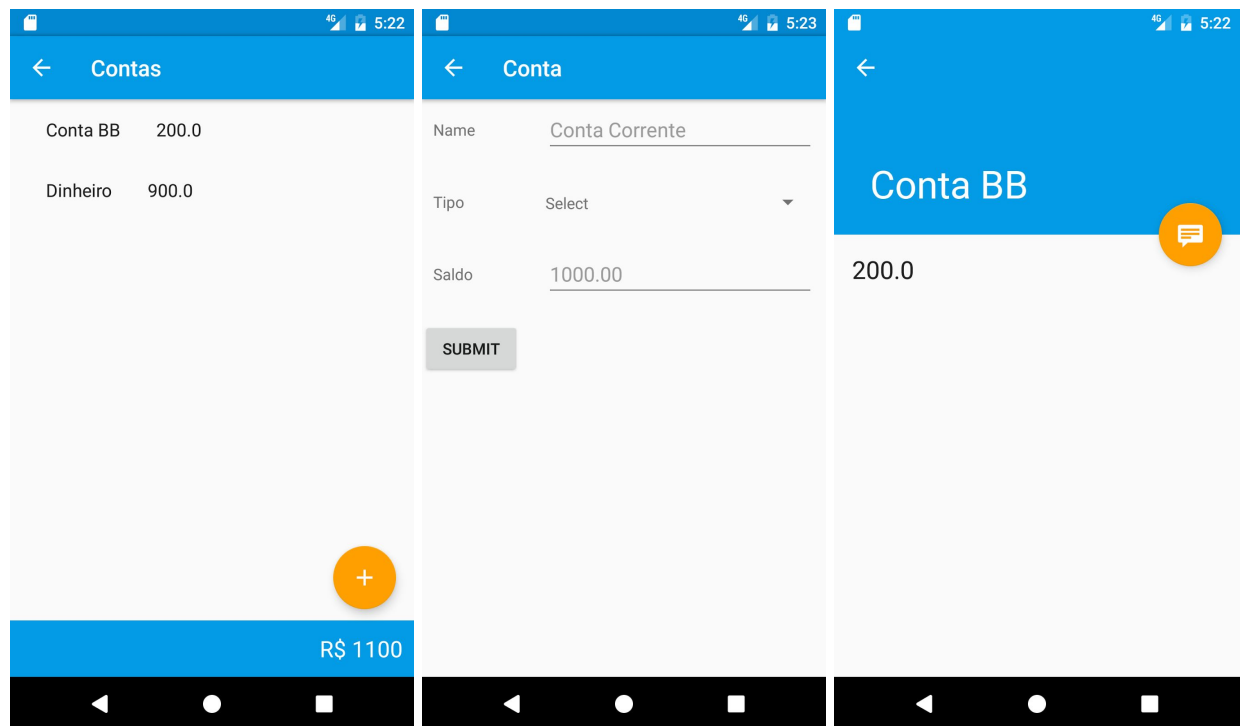
Login



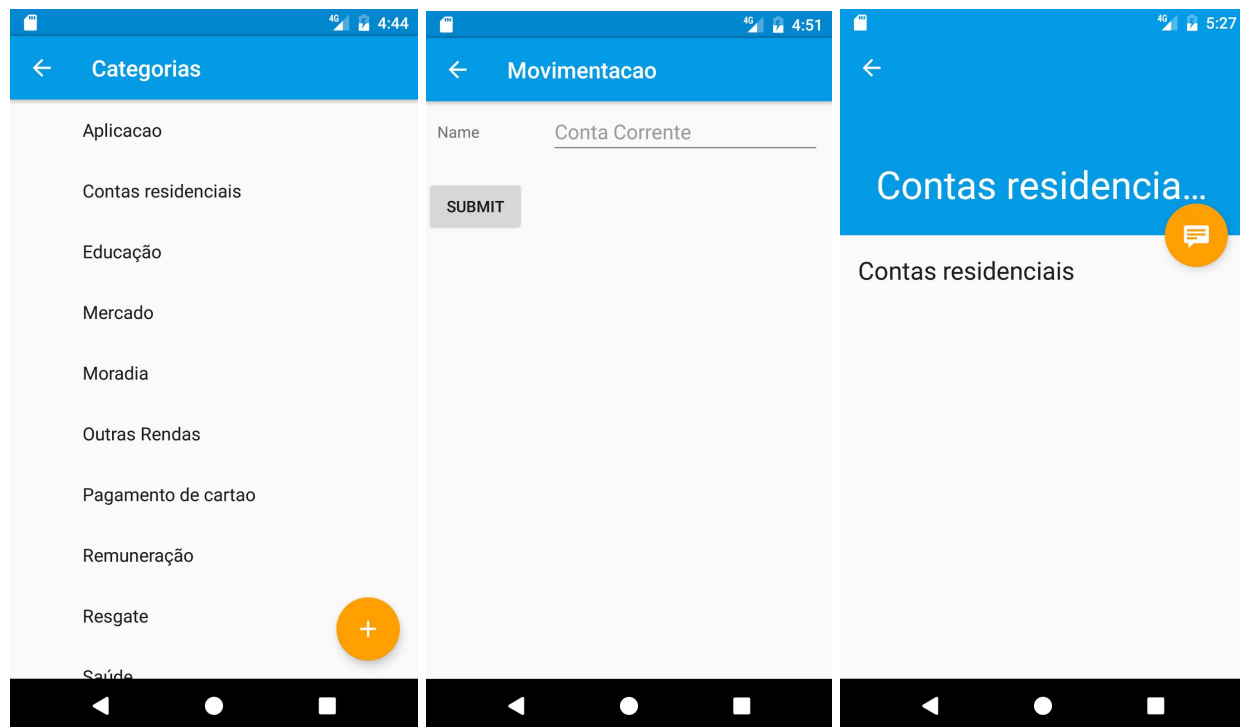
Menu



Cadastro de contas (poupança, conta corrente, cartão de crédito, dinheiro/carteira)



Cadastro de categorias



Registro de receitas/despesas/transferência

The application consists of three main screens for transaction management:

- Movimentacoes:** A list of transactions. Each entry includes a date, a description, and a value. Floating action buttons (FABs) are overlaid on the list: a red '+' for 'Despesas', a green '+' for 'Receitas', a blue '+' for 'Transferência', and an orange 'x' for deleting an entry.
- Movimentacao Detail:** A form to add or edit a transaction. It includes fields for 'Valor' (1000.00), 'Conta' (Select), 'Categoria' (Select), 'Descrição' (Restaurante), and 'Data'. A 'SUBMIT' button is at the bottom.
- Transferência:** A form to record a transfer. It includes fields for 'Valor' (1000.00), 'Conta Origem' (Select), 'Conta Destino' (Select), 'Categoria' (Select), 'Descrição' (Restaurante), and 'Data'. A 'SUBMIT' button is at the bottom.

Frases motivacionais sobre finanças após registros de receitas/despesas

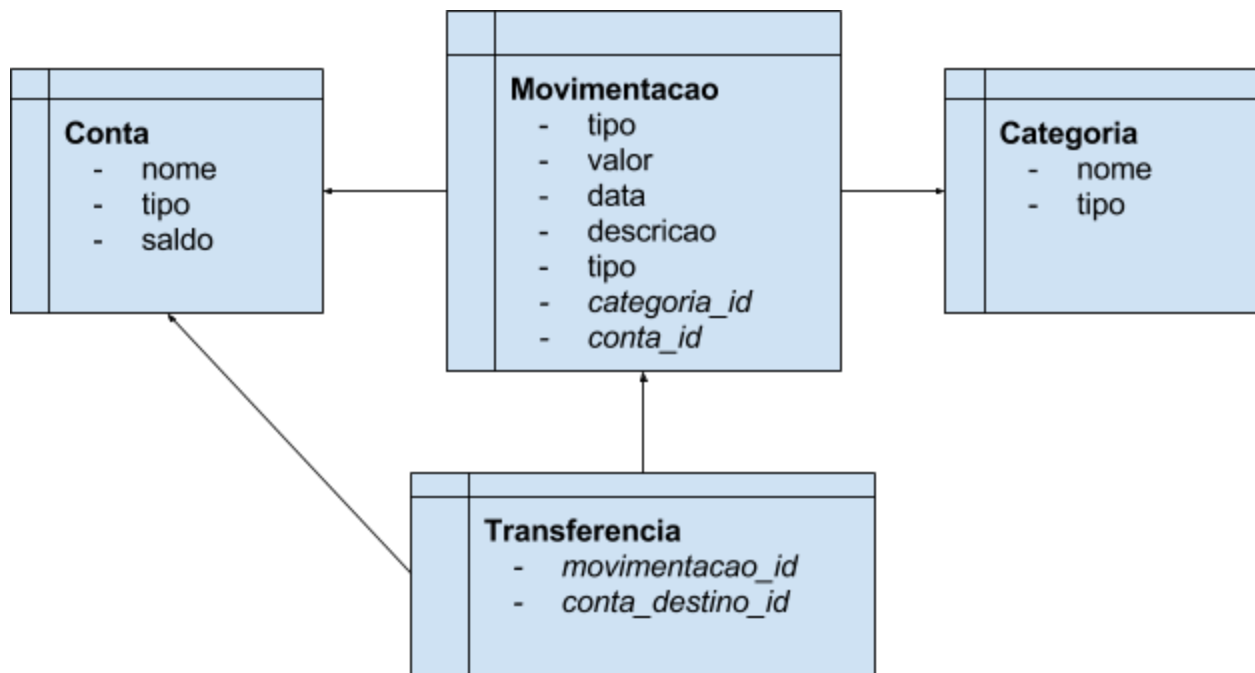


Key Considerations

How will your app handle data persistence?

O Aplicativo manterá os dados num banco de dados sqllite. Será criado um data provider para manipulá-lo.

Abaixo segue uma estrutura das tabelas do banco.



Describe any corner cases in the UX.

O usuário retorna para tela de movimentações clicando no botão de voltar no canto superior à esquerda nas tela de cadastro despesas/receitas/movimentações.

O usuário Acessa o menu clicando no ícone de *sanduíche* no canto superior à esquerda na tela de movimentações.

Describe any libraries you'll be using and share your reasoning for including them.

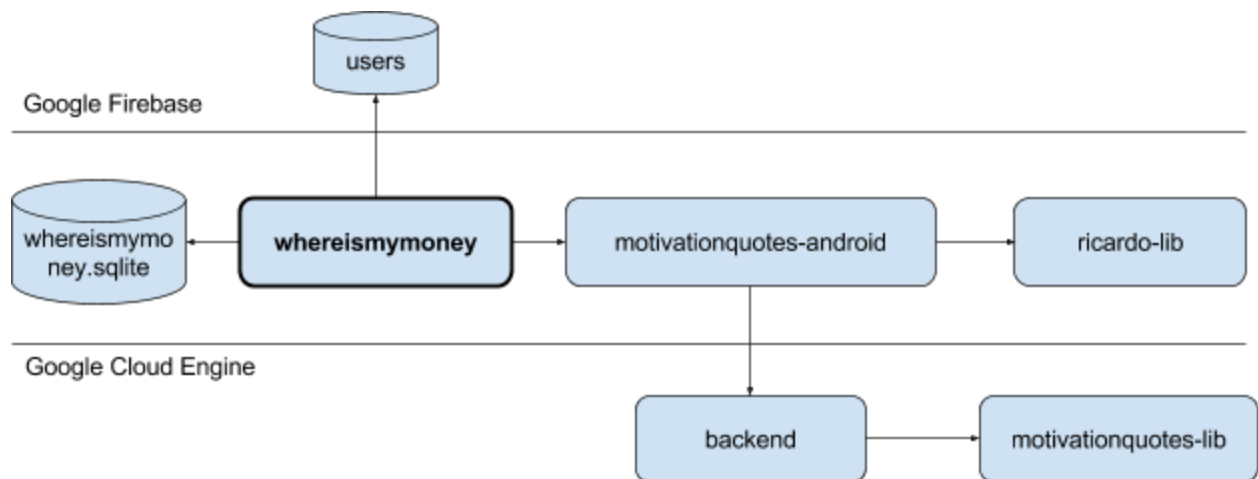
- Picasso (com.squareup.picasso:picasso:2.5.2)
 - Na tela de mensagens motivacionais é exibido uma imagem de fundo, o picasso carregará essa imagem a partir de uma url dada
- Picasso Transformation (p.wasabeef:picasso-transformations:2.1.0)
 - aplicar efeito de blur
 - trabalha em conjunto com o picasso
- OkHttp3 (com.squareup.okhttp3:okhttp:3.6.0')

- facilitar a manipulação de requisições http
- Facebook Stetho (com.facebook.stetho:stetho:1.4.2)
 - para analisar o banco de dados quando debugava
- FloatingActionButton (com.github.clans:fab:1.6.4)
 - criar um menu com vários fabs na tela de movimentações

Describe how you will implement Google Play Services.

O login da aplicação usará o firebase auth.

A imagem abaixo explica a estrutura do projeto em módulos:



Next Steps: Required Tasks

Task 1: Project Setup

Adicionar as dependências ao projeto:

1. Baixar o projeto
 - a. <https://github.com/ricardosbarbosa/WheresMyMoney/>
2. Google cloud plataform <https://console.cloud.google.com>
 - a. Criar um projeto para receber o módulo **backend**
 - b. Habilitar o API Engine no menu lateral do projeto criado
3. Upload do backend
 - a. No android studio publicar o módulo **backend** no projeto criado no *google cloud plataform* do passo anterior
4. Google firebase
 - a. No menu: *Authentication*
 - b. Na aba: *Método de login*
 - c. Habilitar o Provedores de login: *email/senha*

Task 2: Implement UI for Each Activity and Fragment

- Build UI para MovimentacoesListActivity
- Build UI para MovimentacaoDetailActivity
- Build UI para o SideNavDrawer
- Build UI para CategoriasListActivity
- Build UI para CategoriaDetailActivity
- Build UI para ContaListActivity
- Build UI para ContaDetailActivity
- Build UI para MovimentacaoFormFragment
- Build UI para ContaFormFragment
- Build UI para CategoriaFormFragment
- Build UI para TransferenciaFormFragment

Task 2.1: Create layout for each activity and fragment

- layout para a tela de cadastro de receita/despesa
- layout para a tela de cadastro de transferência
- layout para a lista de movimentações
- layout para visualizar os detalhes de movimentações
- layout para o menu lateral
- layout para a lista de categorias
- layout para tela de cadastro de uma conta
- layout para o detalhe de uma categoria

- layout para tela de cadastro de uma categoria
- layout para a lista de contas
- layout para o detalhe de uma conta

Task 3: Criar AsyncTask

Duas AsyncTasks serão usadas no projeto:

1. A primeira é responsável por buscar uma url de uma imagem aleatória para ser usada como background na tela de mensagens motivacionais.
2. A segunda é responsável por solicitar um uma frase motivacional ao módulo hospedado no google cloud plataform.

Task 4: Criar DataProvider

Criar as classes de contrato das tabelas do banco.

As classes de apoio para manipular criação do banco de dados.

Implementar o data provider.

Task 5: Criar Widget

O widget do app exibe apenas tem a função de exibir as contas cadastradas e seus saldos