

[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.

Pessoas que já fazem algum controle financeiro de alguma outra forma, também pode se beneficiar do uso do aplicativo para lhe ajudar nessa tarefa.

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

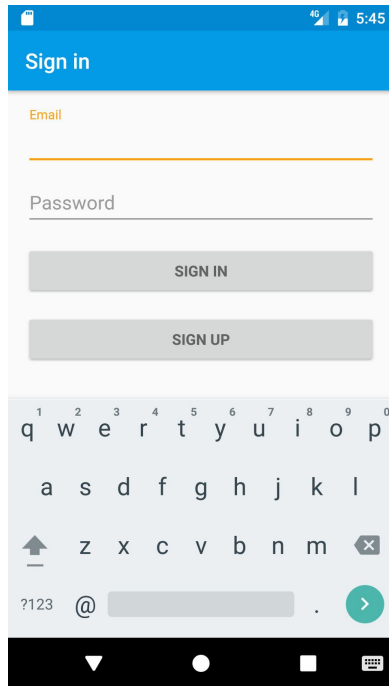
Features futuras

- Gráficos por contas e categorias
- Marcação de movimentações com tags
- Reconhecer movimentação como pagamento de fatura de uma conta de cartão de crédito
- Permitir movimentações fixas recorrentes, diariamente, semanalmente ou mensalmente
- Permitir movimentações parceladas
- Poder informar se uma receita já foi recebida de fato ou é programada
- O mesmo para despesas, poder informar se uma despesa foi paga ou se encontra-se em aberta, atrasada ou programada.

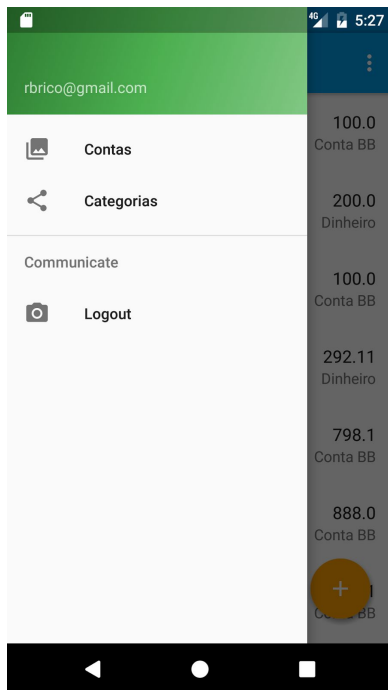
- Paginação por mês
- Exibir movimentações agrupadas por dia na listagem
- Rodapé na lista de movimentações exibindo o saldo do ms quando a funcionalidade de paginação estiver implementada
- Permitir filtros na lista de movimentações
- Busca
- Home de para visão geral
 - com movimentações a pagar e a receber
 - contas atrasadas
 - saldos das contas

User Interface Mocks

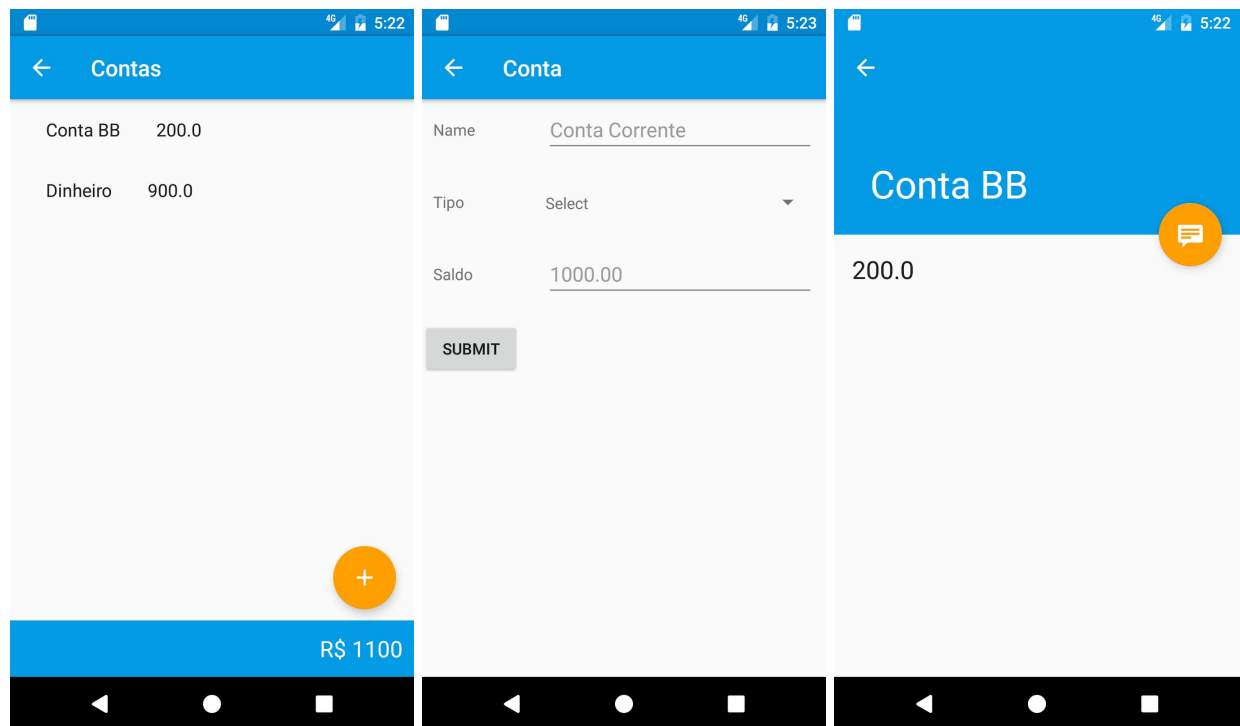
Login



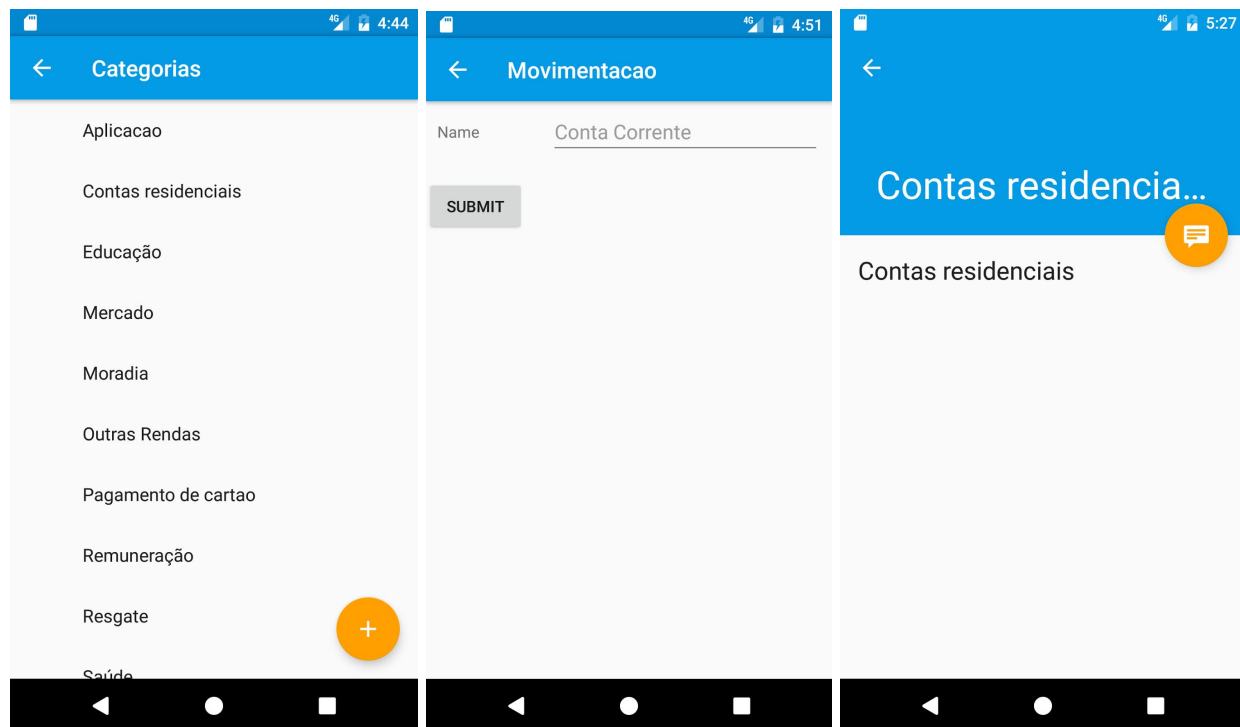
Menu



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



Cadastro de categorias



Registro de receitas/despesas/transferência

← Movimentacao Detail

Valor 1000.00

Conta Select ▼

Categoria Select ▼

Descrição Restaurante

Data

SUBMIT

← Transferência

Valor 1000.00

Conta Origem Select ▼

Conta Destino Select ▼

Categoria Select ▼

Descrição Restaurante

Data

SUBMIT

Frases motivacionais sobre finanças após registros de receitas/despesas

OBS: Essa feature foi criada para suprir os requisitos de capstone, num app real nao colocaria.

Como vai funcionar:

- Sempre, após o cadastro de despesa/receita, essa tela com frases "motivacionais" será exibida.
- Funcionará como um "brinde" por ele estar registrando suas receitas/despesas e "incentivar" que continue.



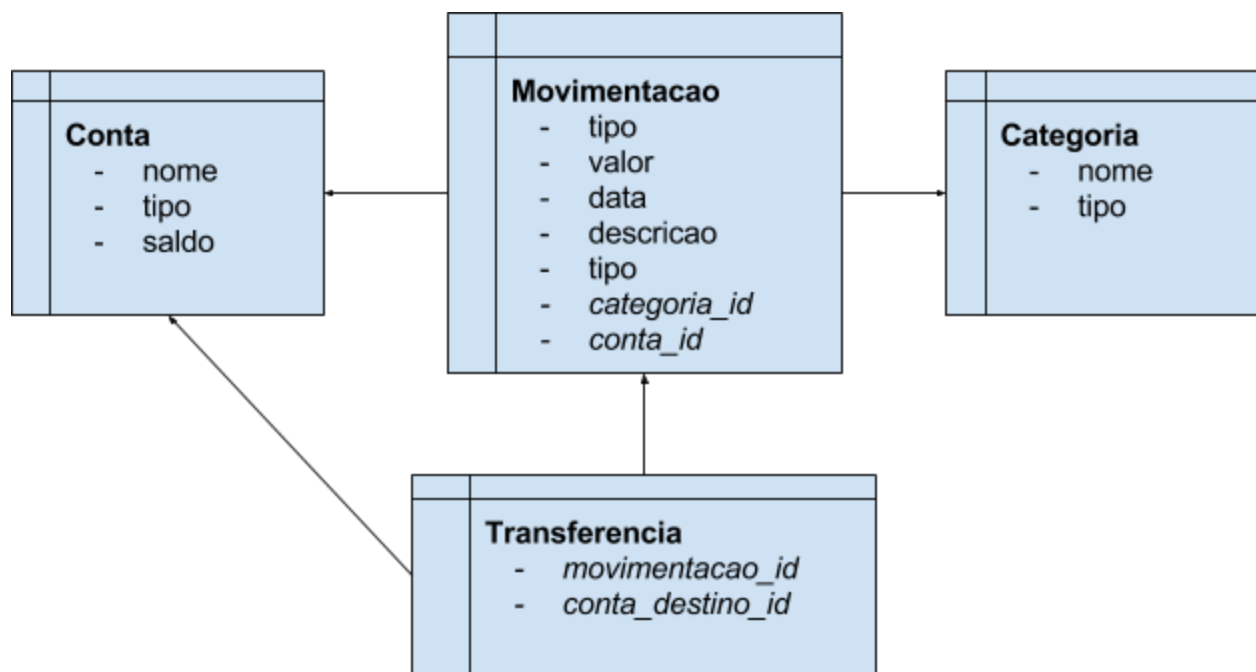
Key Considerations

How will your app handle data persistence?

O Aplicativo manterá os dados num banco de dados sqllite.

Utilizarei as APIs nativas do Android para implementação, criando um content provider para manipulá o database. (Ressalto que não pretendo usar nessa versão o firebase realtime database).

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.

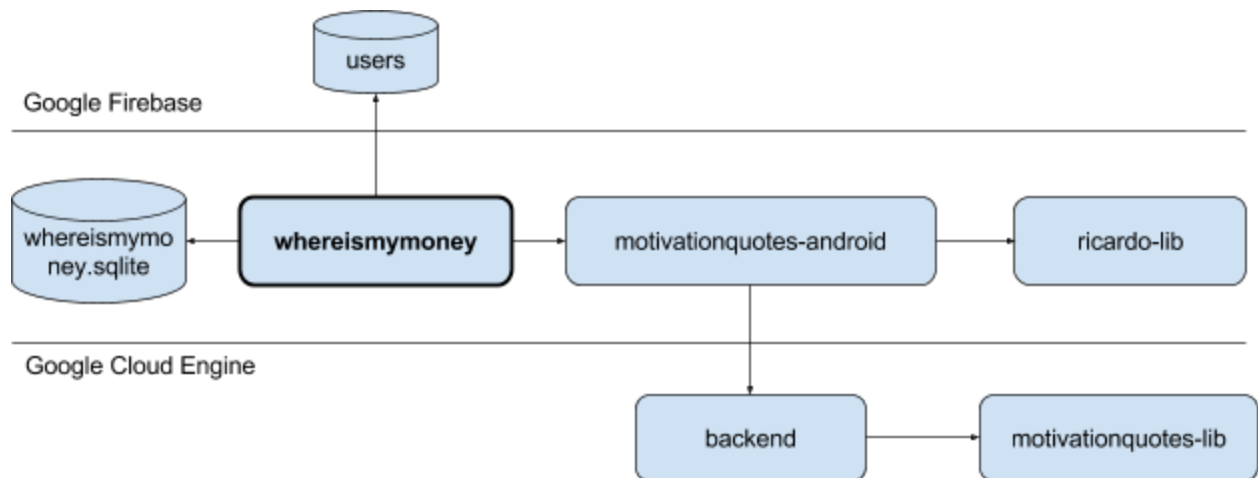
- Picasso (com.squareup.picasso:picasso:2.5.2)
 - Na tela de mensagens motivacionais é exibido uma imagem de fundo, o picasso carregará essa imagem a partir de uma url dada
- Picasso Transformation (p.wasabeef:picasso-transformations:2.1.0)
 - aplicar efeito de blur

- 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

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 2.3: Implementar casos de uso

Task 2.3.1: Caso de uso do cadastro de movimentação de despesa/receita

1. na lista de movimentações o usuário clica no fab que exibe o "submenu de fabs"
2. o usuário clica no fab de despesa ou receita
3. o app exibe a tela configurando o parâmetro DESPESA ou RECEITA de acordo com a seleção do usuário
4. o usuário informa os campos de valor descrição e data
5. o usuário escolhe uma categoria previamente cadastrada através do menu categorias.
6. o usuário escolhe uma conta, previamente cadastrada através do menu contas, onde a movimentação ocorrerá.
7. o usuário clica no botão submit
8. o app inclui o tipo de movimentação baseado no parâmetro do item 3
9. o app valida os dados preenchidos pelo usuário
 - a. todos são obrigatórios
10. o app abre uma transação com o banco de dados
 - a. o app salva a movimentação
 - b. o app atualiza o saldo da conta da movimentacao
 - i. se a movimentação for de receita, o VALOR da movimentação é ACRESCENTADO ao saldo da conta
 - ii. se a movimentação for de despesa, o VALOR da movimentação é REMOVIDO do saldo da conta
 - c. Se as operações forem realizadas com sucesso o app fecha a transação

Task 2.3.2: Caso de uso do cadastro de movimentação do tipo transferência

11. na lista de movimentações o usuário clica no fab que exibe o "submenu de fabs"
12. o usuário clica no fab de transferência
13. o app exibe a tela de cadastro de transferência
14. o usuário informa os campos de valor descrição e data
15. o usuário escolhe uma categoria previamente cadastrada através do menu categorias.
16. o usuário escolhe uma conta de origem - dentre as contas previamente cadastrada através do menu contas - de onde a transferência de valor partirá.
17. o usuário escolhe uma conta de destino - dentre as contas previamente cadastrada através do menu contas - para qual o valor da transferência irá.
18. o usuário clica no botão submit
19. o app valida os dados preenchidos pelo usuário
 - a. todos são obrigatórios
20. o app abre uma transação com o banco de dados
 - a. o app salva uma movimentação com os valores colocando o id da conta origem selecionada na coluna conta_id

- b. o app guarda o id da movimentação cadastrada
- c. o app salva uma transferência com os valores do id da movimentacao cadastrada e o id da conta de destino selecionada
- d. o app atualiza o saldo da conta de ORIGEM, REMOVENDO o VALOR da movimentação do saldo da conta
- e. o app atualiza o saldo da conta de DESTINO, ACRESCENTANDO o VALOR da movimentação ao saldo da conta
- f. Se as todas as operações forem realizadas com sucesso o app fecha a transação

Task 3: Criar AsyncTask

Duas AsyncTask 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 plattaform.

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