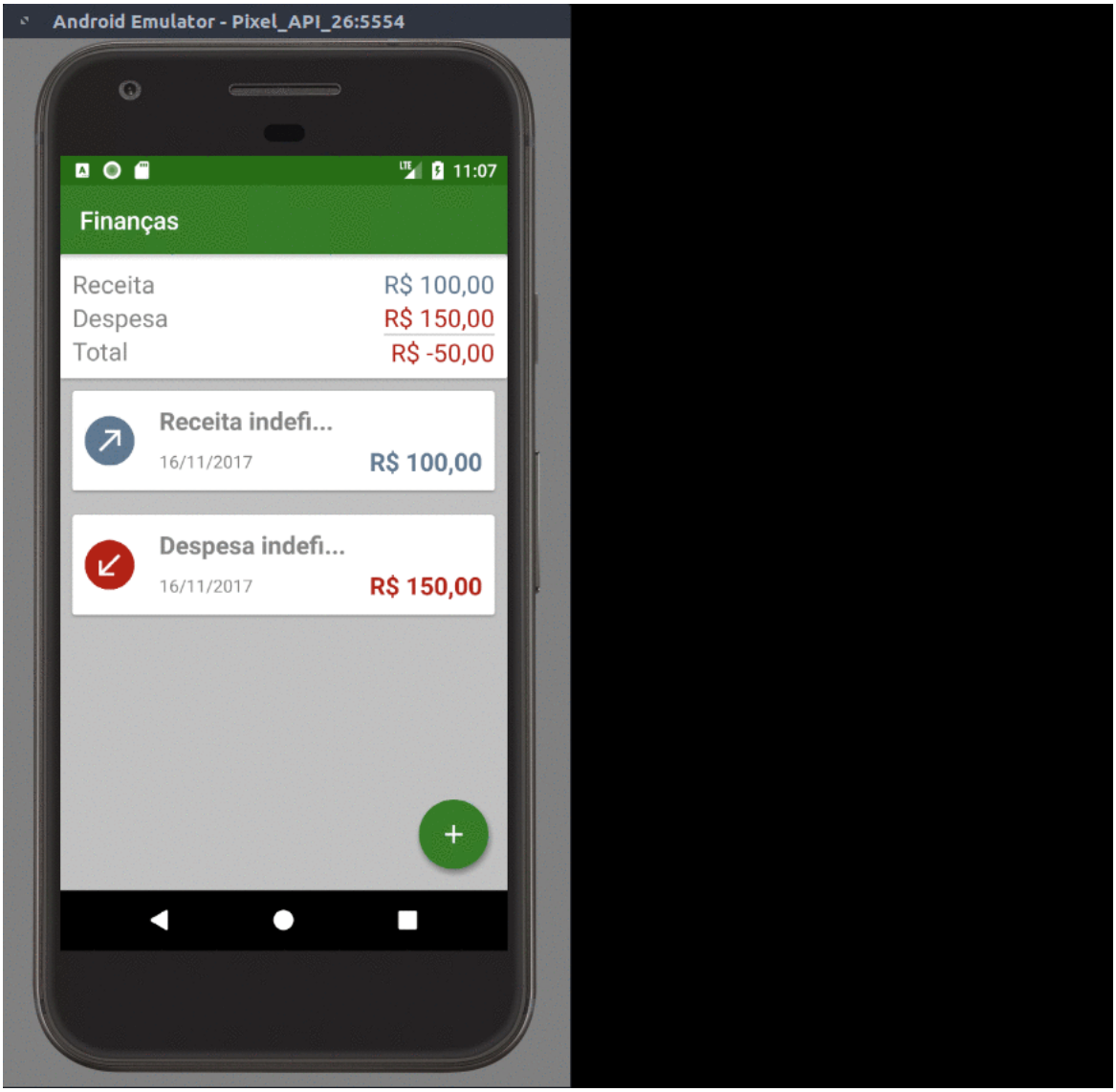
02

Criando o DAO para manter as transações

> Caso você precisar do projeto com os ajustes que foram realizados na aula anterior, você pode baixá-lo por meio deste link (https://github.com/alura-cursos/financas-kotlin-parte3/archive/aula-6.zip).

Finalizamos todas as features da nossa App, permitindo que o usuário adicione, altere e remova transações. Só quando temos algumas transações e o celular é rotacionado, temos o seguinte resultado:



Repare que as transações são perdidas, então faremos um processo de refatoração para evitar esse tipo de situação.

O nosso primeiro passo é delegar toda a responsabilidade de manter as transações salvas para um classe responsável, ou seja, um DAO para as transações.

Criando a class para manter as transações

Sendo assim, crie a classe `TransacaoDAO` no pacote **dao**. Em seguida, crie a property `transacoes` do tipo `MutableList<Transacao>` que servirá como base de dados das transações.

Implementando os comportamentos básico.

Com a lista pronta, implemente as seguintes funções:

- `adiciona()`
- `altera()`
- `remove()`

Ambas as funções devem manter o mesmo comportamento que fizemos na `ListaTransacoesActivity`, a diferença é que não será realizado o processo de atualizar o resumo ou a lista de transações.

Utilizando o DAO na Activity

Após realizar a implementação, utilize a `TransacaoDAO` na Activity: crie um property chamada `dao` e modifique todos os pontos que realizava a adição, alteração e remoção direto na property `transacoes` para que agora utilizem as funções do DAO.

Em seguida, faça com que a referência da property `transacoes` seja inicializada pela property `transacoes` do DAO, para manter as mesmas informações.

Por fim, teste a App e veja se as funcionalidades de adicionar, alterar e remover funcionam como antes.

> É válido lembrar que o problema de rotação ainda vai continuar.