



## Curso de Especialização em Programação para Dispositivos Móveis

### Trabalho Final da Disciplina de Interfaces Gráficas em Aplicações para Dispositivos Móveis

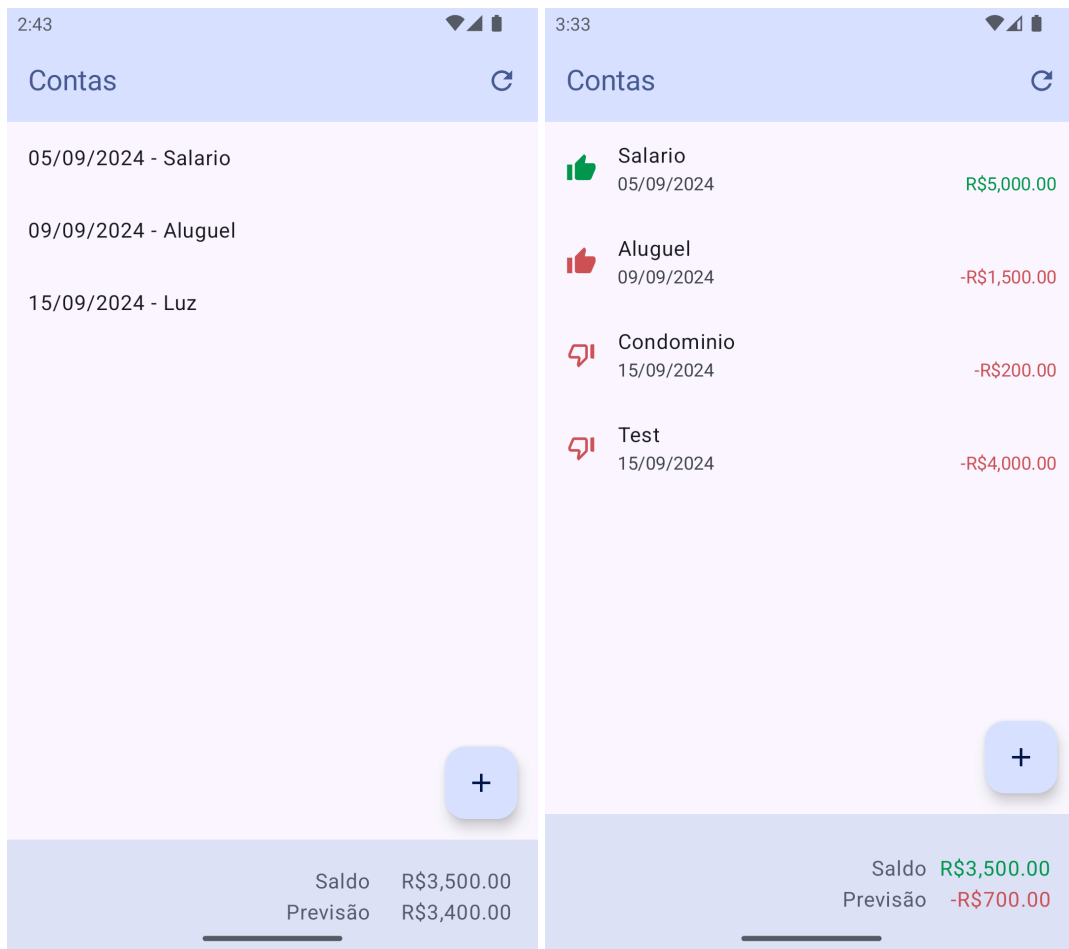
#### **Objetivo do trabalho:**

Avaliar a capacidade dos acadêmicos em criar Interfaces Gráficas utilizando o [Jetpack Compose](#).

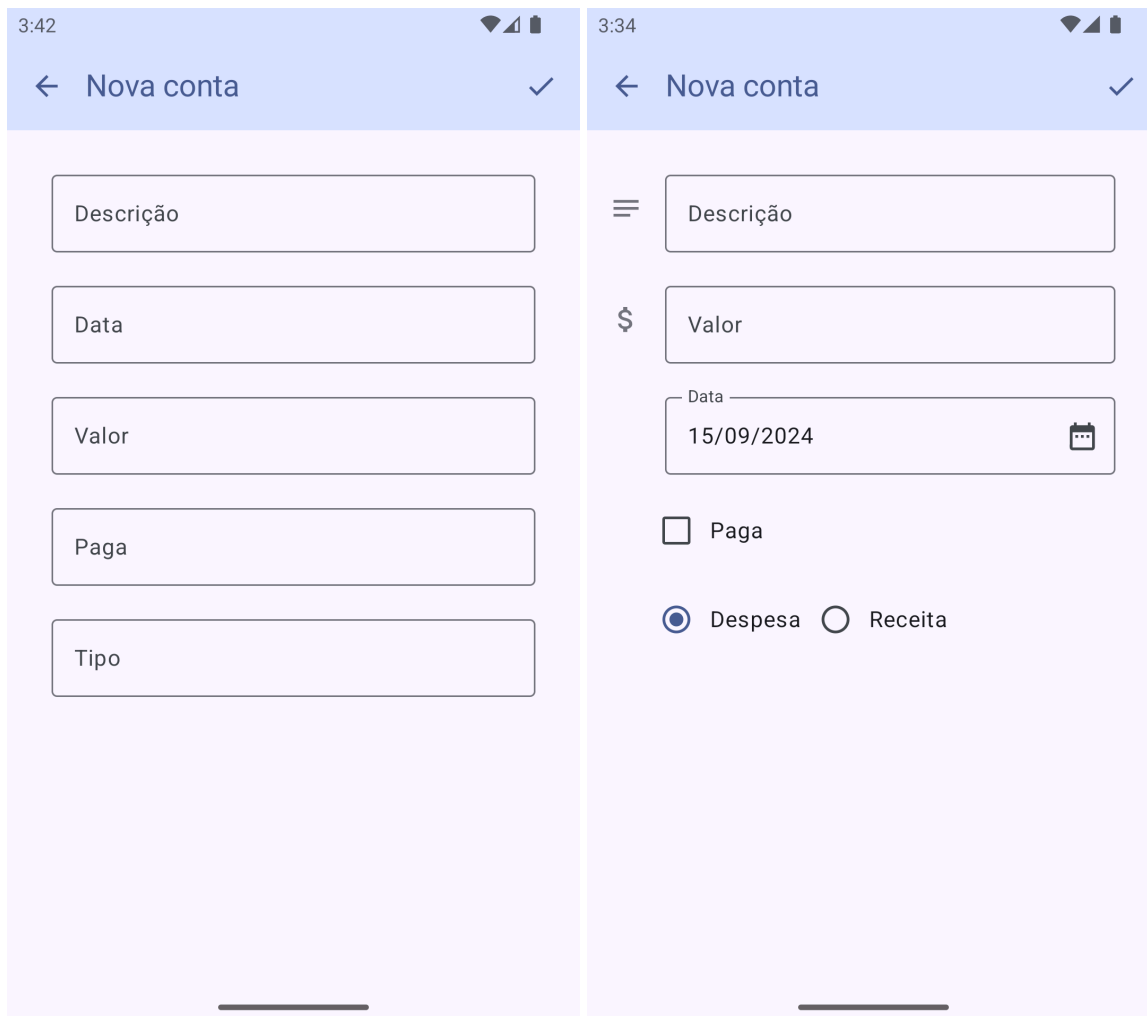
#### **Especificação:**

O trabalho pode ser desenvolvido de forma individual, ou em duplas (definidas pelos próprios acadêmicos). Será compartilhada com os acadêmicos, uma aplicação previamente desenvolvida, que serve para gerenciar contas a pagar e a receber. Dentro dessa aplicação, será necessário alterar partes da interface gráfica para melhorar a experiência do usuário. Essas alterações estão descritas abaixo.

1. A listagem de contas exibe apenas a data e a descrição de cada conta, sem exibir o restante das informações, dessa forma, será necessário realizar os seguintes ajustes:
  - Em cada item da lista, adicionar um ícone alinhado à esquerda, para indicar se a conta foi paga ou não. Além disso, alterar a cor do ícone para indicar se é uma despesa ou receita. (Peso 1,5)
    - Sugestão de ícones: **Icons.Filled.ThumbUp** para contas pagas e **Icons.Filled.ThumbDownOffAlt** para contas não pagas.
    - Sugestão de cores: **Color(0xFFCF5355)** para despesas e **Color(0xFF00984E)** para receitas.
  - Em cada item da lista, deixar a primeira linha apenas para a descrição da conta, movendo a data para a linha de baixo. Além disso, adicionar o valor da conta na mesma linha da data, porém, alinhado à direita. Caso a conta seja uma despesa, o texto do valor deverá ser exibido com sinal negativo. A cor do texto do valor deve ser alterada de acordo com o tipo da conta. A sugestão é utilizar as mesmas cores sugeridas para o ícone de paga/não paga. (Peso 1,5)
  - Modificar os textos dos valores do saldo e da previsão, que estão na BottomBar, para que sigam a mesma regra do texto do valor da conta: usar uma cor para cada tipo de conta e, caso seja uma despesa, exibir o valor com sinal negativo. (Peso 1,0)
  - Exemplo do Antes e Depois para auxiliar no desenvolvimento:



2. O formulário possui campos que permitem cadastrar, editar ou excluir uma conta, porém, todos os campos são de texto, dificultando a entrada de outros valores, como datas e booleanos. Dessa forma, será necessário realizar os seguintes ajustes no formulário:
- Adicionar ícones à esquerda dos campos referentes à descrição e ao valor. Sugestão de ícones: **Icons.AutoMirrored.Filled.Notes** para a descrição e **Icons.Filled.AttachMoney** para o valor. (Peso 1,0)
  - Modificar o campo de data para que seja um DatePicker. (Peso 1,5)
  - Modificar o campo de paga/não paga para que seja um Checkbox. (Peso 1,0)
  - Modificar o campo do tipo da despesa para que sejam dois RadioButtons, um para cada tipo de despesa. (Peso 1,5)
  - Não permitir salvar uma despesa sem a descrição e o valor preenchidos. Além disso, validar para que a aplicação não “quebre” ao digitar caracteres inválidos no campo de valor. (Peso 1,0)
  - Exemplo do Antes e Depois para auxiliar no desenvolvimento:



### **Critérios de Avaliação:**

- Entrega do projeto ajustado através do Moodle. Os projetos devem conter apenas o código fonte, sem os arquivos gerados por builds. Para remover os builds, pode ser utilizado [esse comando](#) Gradle.
  - Caso o trabalho seja feito em duplas, apenas um dos integrantes deve enviar pelo Moodle, deixando um comentário com o nome dos integrantes da dupla.
- Verificação da existência e correto funcionamento dos itens solicitados para desenvolvimento.

### **Prazo para Entrega do Projeto:**

**04/10/2024**