

Formação Flutter

Desafio Fase 2 - App Lista de Compras



Termo de uso

Todo o conteúdo deste documento é propriedade da Growdev. O mesmo pode ser utilizado livremente para estudo pessoal.

É proibida qualquer utilização desse material que não se enquadre nas condições acima sem o prévio consentimento formal, por escrito, da Growdev. O uso indevido está sujeito às medidas legais cabíveis.

Objetivo do documento

Este material tem como objetivo descrever o desafio final da fase que deverá ser construído para aprovação da mesma.

Vamos ao desafio!

Chegou a hora de aplicar o conhecimento adquirido nesta fase, lembrando sempre que os desafios serão nossos principais indicadores sobre o conhecimento de vocês, tanto para ajudá-los como na hora do direcionamento para as oportunidades geradas na fase bonus.

Sua missão é desenvolver uma interface intuitiva e responsiva, explorando os conceitos de widgets, componentização, alinhamento de widgets, imagens, modais e entrada de dados. Mostre suas habilidades em Flutter e crie uma experiência única para os usuários. Vamos iniciar essa jornada com muito entusiasmo!

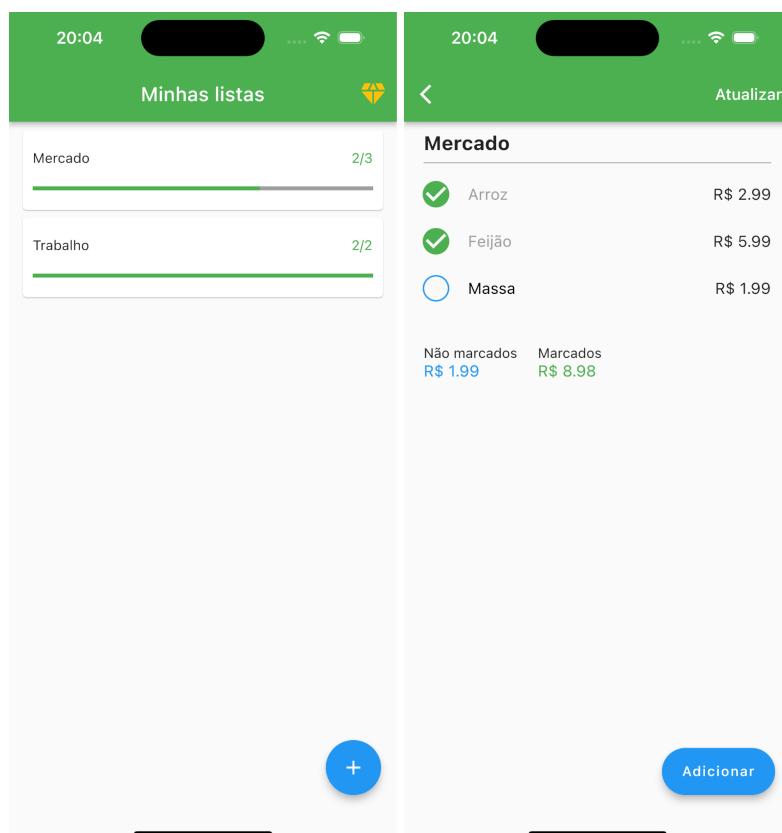
O desafio

O propósito central deste desafio consiste na criação de um aplicativo destinado ao gerenciamento de listas de compras. O aplicativo almejado visa capacitar os usuários a formarem grupos distintos para suas compras, nos quais poderão adicionar produtos acompanhados de seus respectivos valores. O aplicativo deve permitir aos usuários a habilidade de assinalar o status de compra de cada produto na lista, indicando se foi adquirido ou não. Ao término da lista, uma resumo exibindo o montante total dos itens comprados e dos itens que não foram comprados.

O aplicativo precisa conter:

- Uma página principal (ao abrir é ela que o usuário verá).
- Na página principal o usuário poderá ter duas visualizações, sendo elas:
 - Uma visualização que mostra uma imagem centralizada com um texto indicando ao usuário para criar uma lista de compras.
 - Uma visualização que deve mostrar as listas de compras cadastradas.

- Uma página para o usuário digitar e cadastrar uma lista de compras
- Uma página para o usuário visualizar a lista de produtos e interagir com eles.
- Um bottom sheet para poder cadastrar um produto dentro de uma lista de compras.



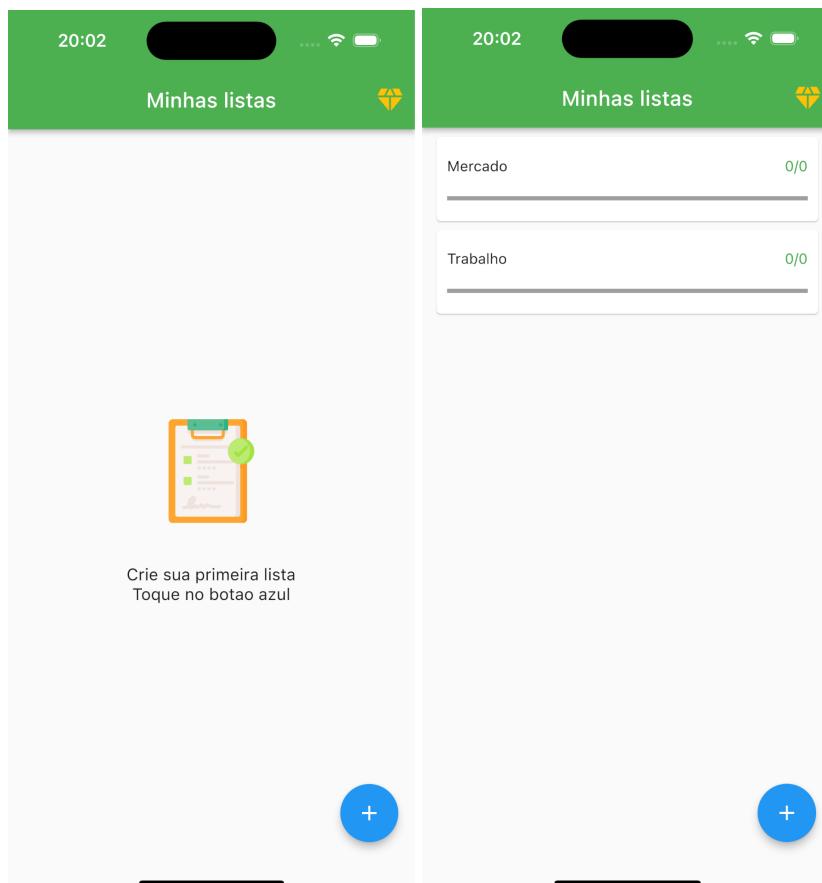
Todos os arquivos que você precisará para construir a página estão anexados na plataforma, inclusive as imagens das telas do aplicativo.

Você deve seguir fielmente o protótipo da imagem para o seu desafio ser aceito e para a automação de testes executar corretamente.

Além da construção da tela seguir fielmente o protótipo, algumas coisas são importantes para dar tudo certo. Você deve inserir um algumas keys (chaves) importantes em alguns widgets para a automação de teste

conseguir localizar e executar os testes corretamente. Você deve adicionar as keys nos seguintes componentes:

- Página de listagem de listas de compras



- Deve ser adicionado uma key dentro do componente do Text no title do AppBar => Key("appBarTitle")
- Deve ser adicionado uma key no componente do FloatingActionButton => Key("addListBtn")
- O ícone do FloatingActionButton é => Icons.add
- O ícone do diamante no AppBar é => Icons.diamond

- Visualização que mostra quando o usuário não tem nenhuma lista de compras cadastrada



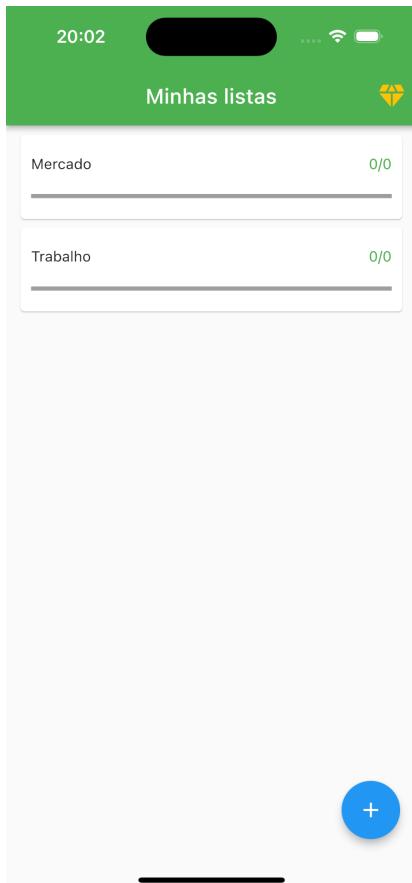
- Deve ser adicionado uma key dentro do componente da imagem => Key("emptyListImage")

- Página de adicionar uma lista de compras



- Deve ser adicionado uma key dentro do componente do TextField => Key("listNameInput")
- Deve ser adicionado uma key dentro do componente do botão de voltar => Key("backToListsBtn")
- Deve ser adicionado uma key dentro do componente do botão de criar => Key("createListBtn")

- Card da lista de compras



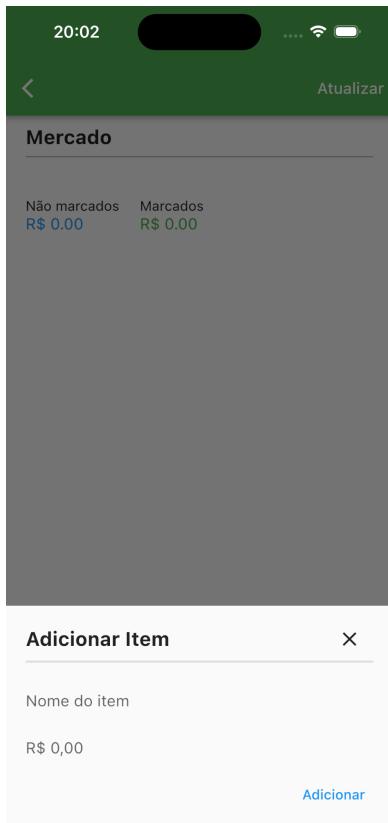
- Deve ser adicionado uma key dentro do componente clicável que desenha cada lista de compra na tela => Key("shoppingListCard")

- Página de lista de produtos



- Deve ser adicionado uma key dentro do componente do FloatingActionButton => Key("addNewItemBtn")
- Deve ser adicionado uma key dentro do componente do botão de atualizar => Key("updateListBtn")
- No elemento que representa o produto na lista
 - Deve ser adicionado uma key dentro do CheckBox => Key("productCheckbox")

- BottomSheet para adicionar um produto



- Deve ser adicionado uma key dentro do TextField do nome do produto => Key("inputItem")
- Deve ser adicionado uma key dentro do TextField do valor do produto => Key("inputValue")
- Deve ser adicionado uma key dentro do botão de adicionar => Key("addItemBtn")
- O ícone de fechar o modal é => Icons.close

- Algumas validações são feitas através dos textos em tela, então para o seu aplicativo passar nos testes, além das keys serem definidas conforme mencionado acima, os textos precisam ser iguais aos do protótipo.

Instruções de entrega

Você deve subir o desafio em um repositório público no Github com o código do projeto já com as definições das keys adicionadas e em seguida responder ao formulário disponibilizado informando o mesmo e-mail cadastrado na plataforma, o link do repositório e a fase a qual o desafio pertence.

