

PROJETO

A plataforma alvo deve ser o sistema operacional no **Android** ou navegador **Web**

O layout e design é atribuído para ser implementado em Flutter.

O aplicativo implementado deve contar com 5 telas. Sendo assim, o seu grupo deve selecionar 5 telas para utilizar como referência no link correspondente ao leiaute selecionado.

Link: <https://www.uistore.design/items/online-groceries-free-app-ui-kit-for-figma/>

Na atividade, serão utilizados as seguintes avaliações:

- Uso de recursos apresentados
- Fidelidade de implementação de layout e requisitos
- Não precisa mudar o estado do aplicativo.
- Atender a orientação de design.

Implementação de Recursos Flutter

Os recursos (indicado como RC) que devem ser implementados, e seu código para identificação, estão listados a seguir:

- [RC1] Especifique uma estrutura de pastas (arquitetura) para o projeto do aplicativo, semelhante ao que foi apresentado na Aula 3 - Rotas e Navegação. Busque respeitar a arquitetura definida, alocando os arquivos de código fonte entre eles. (até 0,5 ponto)
- [RC2] Utilize Rotas e Navegação, aplicando a técnica com MaterialApp apresentada em Aula 3 - Rotas e Navegação. (até 1,0 ponto)
- [RC3] Passe dados entre as páginas utilizando as estratégias descritas na Aula 3 - Rotas e Navegação. (até 1,0 ponto)
- [RC4] Defina pelo menos um formulário no seu aplicativo, como apresentado na Aula 4 (PDMI6) - Widgets de Input e Dialog, Buttons e Material Widgets. (até 1,0 ponto)
- [RC5] Utilize a classe LayoutBuilder ou MediaQuery para tornar o aplicativo responsivo à rotação de tela, tendo como base a aula Aula 5 (PDMI6) - Criação de Layouts com Flutter. (até 1,0 ponto)
- [RC6] Defina o estilo do app em um ThemeData, como apresentado na Aula 5 (PDMI6) - Criação de Layouts com Flutter. (até 0,5 ponto) Dicas:
 - Monte o layout do aplicativo utilizando os Widgets de layout (Column, Row, ListView, CheckBox, Scaffold, etc). Não é necessário se limitar aos widgets da aula. Veja opções de layouts em <https://docs.flutter.dev/development/ui/widgets/layout>
 - Para lidar com problemas de overflow de componentes (quando um widget filho ultrapassa os limites espaciais do widget pai):

Utilize os widgets SingleChildScrollView ConstrainedBox para resolver o problema. Veja em: e

- <https://api.flutter.dev/flutter/widgets/SingleChildScrollView-class.html>
- <https://api.flutter.dev/flutter/widgets/ConstrainedBox-class.html>

Fidelidade de Layout

Para a avaliação da implementação do layout, será verificado o quanto a sua implementação é fiel aos leiautes e designs selecionados para implementação. A pontuação será atribuída por tela implementada.

- Tela 1 - Tela de boas-vindas - até 0.5 ponto;
- Tela 2 - Tela que se adequa ao uso de ListView - até 1 ponto;
- Tela 3 - Tela que se adequa ao uso de formulário - até 1 ponto;
- Tela 4 - Tela que se adequa ao uso de GridView - até 1 ponto;
- Tela 5 - Tela que apresenta um item selecionado - até 1 ponto;
- Uso de Widgets Material (AppBar, Drawer, ButtonNavigationBar, FloatingActionButton) sem funcionalidades.

Obs1: Se o layout não contiver elementos que possam ser mapeados diretamente para Widgets Material, deve utilizá-los mesmo assim, ajustando o estilo para estar em conformidade ao layout selecionado;

Obs2: A aplicação implementada não terá alteração de estado, portanto não será necessário o uso de banco de dados;

Obs3: Se desejar, mais telas do que as especificadas poderão ser implementadas.

