**Dart Com Flutter**

Sumário

[DART 2](#_Toc87949425)

[Listas 2](#_Toc87949426)

[Widgets – USANDO MATERIAL COMPONENTS WIDGETS 2](#_Toc87949427)

[Column: Alinhamento De Seus Filhos 2](#_Toc87949428)

# DART

## Listas

final List<Transfer> \_transfer = [];//Lista vazia

# Widgets – USANDO MATERIAL COMPONENTS WIDGETS

## Column: Alinhamento De Seus Filhos

**Mexendo nos eixos Y e X:**

Column(

        //Espaço máximo entre os widgets dentro da Column, no eixo 'Y':

        mainAxisAlignment: MainAxisAlignment.spaceBetween,

        //Coloca os filhos da Column no início da posição:

        crossAxisAlignment: CrossAxisAlignment.start,

//LOGO DA EMPRESA:

        children: <Widget>[

Vimos que no Column conseguimos alinhar os Widgets filhos a partir do alinhamento de eixo main ou cross.

Esse padrão também é comum no Widget Row, utilizado para ordenar Widgets em linha. A diferença no Row é que o main representa o **eixo horizontal** e o cross o eixo vertical.

## InkWell: Atribuindo eventos à Widgets

[...]

Padding(

            padding: const EdgeInsets.all(8.0),

            //'GestureDetector' - Atribui ao widget capacidade de responder a todo tipo de eventos.

            //'InkWell' - Atribui ao widget capacidade de responder a todo tipo de eventos, mas é próprio do MaterialDesign, por isso é mais adequado. Permite efeitos.

            //Para o efeito funcionar, necessário envolver InkWell com 'Material' e responsabilizar ele pela cor do seu filho.

            child: Material(

              //Cor do container envolvido pelo Material:

              color: Theme.of(context).primaryColor,

              child: InkWell(

                onTap: () {

                  Navigator.of(context).push(

                    MaterialPageRoute(

                      builder: (context) => ListContacts(),

                    ),

                  );

                },

                child: Container(

                  //Padding atribuído aos filhos do container:

                  padding: const EdgeInsets.all(8.0),

                  height: 80,

                  width: 130,

[...]

## SizedBox: Mudando as dimensões de quaisquer widgets

[...]

//BOTÃO SUBMIT:

              SizedBox(

                height: 56.0,

                //Máximo de largura para o ElevatedButton:

                width: double.maxFinite,

                child: Padding(

                  padding: const EdgeInsets.only(top: 16.0),

                  child: ElevatedButton(

                    onPressed: () {},

                    child: Text(

                      'Add',

                      style: TextStyle(

                        fontSize: 20.0,

                      ),

[...]

# THEMA DO APP

    return MaterialApp(

      //Define tudo sobre o tema geral do App.

      theme: ThemeData(

        primaryColor: Colors.green.shade900,

        colorScheme: ColorScheme.fromSwatch().copyWith(

          primary: Colors.green.shade900,

          secondary: Colors.blueAccent.shade700,

        ),

        //Cor para botões:

        elevatedButtonTheme: ElevatedButtonThemeData(

          style: ButtonStyle(

            backgroundColor:

                MaterialStateProperty.all<Color>(Colors.blueAccent.shade700),

          ),

        ),

      ),

      home: const Dashboard(),

    );

## Como Usar A Cor Do Tema Definido Em Um Widget Qualquer:

color: Theme.of(context).primaryColor,

Prop color de um Container.