



Desenvolvimento
Mobile 2
Aula 05

Prof. Me Daniel Vieira

SENAI

Agenda

- 1- Animações com flutter
- 2 - Exemplo
- 3-SnackBar
- 4- Exemplo
- 5 - Exercícios

Tipos de animações no Flutter

No Flutter existe suporte para diversos tipos de animações, facilitando a implementação delas em seu projeto, em geral existem algumas categorias de animações, como as:

- Animações implícitas: São a primeira coisa que procuramos quando precisamos de animações, elas são animações *prontas*, o Flutter possui uma lista de *widgets* que pode ser animado dessa forma;
- Animações implícitas personalizadas: Quando o que precisamos não está na lista de *widgets* de animação implícita de Flutter, é hora de criar as animações implícitas personalizadas;
- Animações explícitas: O Flutter também disponibiliza uma série de *widgets* desse tipo, elas são uma opção quando precisamos obter mais controle sobre as animações ou controlar mais de uma animação ao mesmo tempo, algo que não é possível com animações implícitas;

Tipos de animações no Flutter

- Animações explícitas personalizadas: Quando não existem *widgets* de animação explícita para o que precisamos, é novamente o momento de criar os seus próprios *widgets*, e para isso personalizamos as animações com duas classes (AnimatedBuilder e AnimatedWidget);
- Animações com pacotes externos: Então, se nenhuma das opções acima por possível, se sua animação se parece mais com um desenho, se for muito complexa para transformar em código, existem pacotes para te auxiliar a inserir elas na sua aplicação.

Tipos de animações no Flutter

- **Animações implícitas**
- Os **widgets de animação implícita** do Flutter geralmente são a primeira opção procurada por nós pessoas programadoras quando queremos adicionar animações nos aplicativos, pois facilitam a criação de uma animação de forma menos complexa. Eles são **widgets** prontos, com animações que possuem início e fim, e que não se repetem.
- No Flutter existem animações de **widgets** já existentes, elas **são chamadas AnimatedFoo**, em que o **Foo é o nome do widget que ao ter suas propriedades alteradas é animado automaticamente**, veja abaixo alguns dos **widgets** de animações implícitas:
 - .

Tipos de animações no Flutter

- **AnimatedAlign:** Versão animada do *widget Align*, em que quando o alinhamento é alterado, é feita uma animação para a nova posição;
- **AnimatedContainer:** Versão animada do *widget Container*, em que as suas propriedades ao serem alteradas, é feita uma animação da mudança;
- **AnimatedCrossFade:** Esse *widget* realiza a mudança entre dois *widgets* filhos de tamanhos iguais (caso sejam de tamanhos diferentes, o maior é “recortado”), animando a transição entre eles;
- **AnimatedDefaultTextStyle:** Versão animada do *DefaultTextStyle* que faz a animação da mudança de estilo do texto sempre que algo é alterado;
- **AnimatedList:** Esse *widget* possui uma lista de itens e realiza uma animação sempre que um item é adicionado ou removido;

Tipos de animações no Flutter

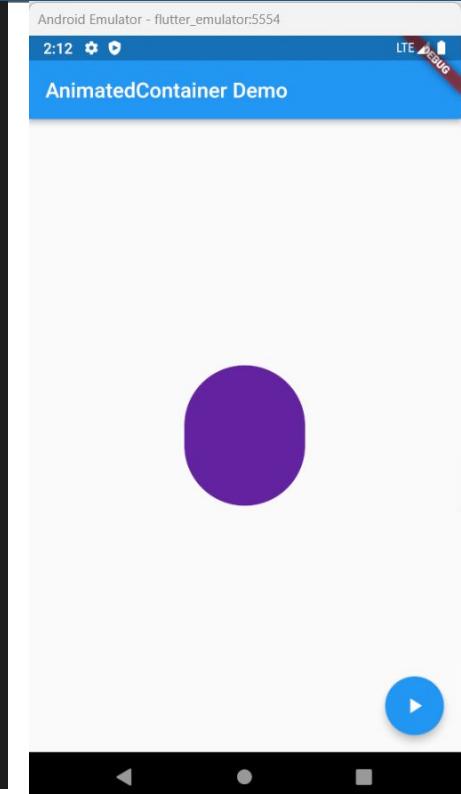
- **AnimatedAlign:** Versão animada do *widget Align*, em que quando o alinhamento é alterado, é feita uma animação para a nova posição;
- **AnimatedContainer:** Versão animada do *widget Container*, em que as suas propriedades ao serem alteradas, é feita uma animação da mudança;
- **AnimatedCrossFade:** Esse *widget* realiza a mudança entre dois *widgets* filhos de tamanhos iguais (caso sejam de tamanhos diferentes, o maior é “recortado”), animando a transição entre eles;
- **AnimatedDefaultTextStyle:** Versão animada do *DefaultTextStyle* que faz a animação da mudança de estilo do texto sempre que algo é alterado;
- **AnimatedList:** Esse *widget* possui uma lista de itens e realiza uma animação sempre que um item é adicionado ou removido;

Tipos de animações no Flutter

- **AnimatedOpacity:** Versão animada do *widget* Opacity, ele realiza a animação da mudança do valor de opacidade do *widget* filho;
- **AnimatedPositioned:** Versão animada do *widget* Positioned, em que ao ser alterada a posição é feita uma animação da mudança; e
- **AnimatedSize:** Esse *widget* anima a mudança das propriedades de borderRadius e elevation do filho.

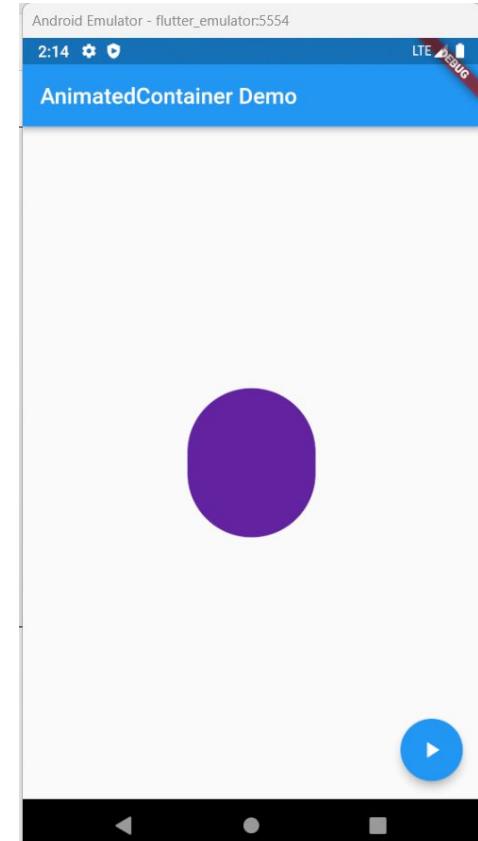
Animated Container

```
1 ✓ import 'dart:math';
2
3 import 'package:flutter/material.dart';
4
5 Run | Debug | Profile
6 void main() => runApp(const AnimatedContainerApp());
7
8 class AnimatedContainerApp extends StatefulWidget {
9   const AnimatedContainerApp({super.key});
10
11   @override
12   State<AnimatedContainerApp> createState() => _AnimatedContainerAppState();
13 }
14
15 class _AnimatedContainerAppState extends State<AnimatedContainerApp> {
16   // Define the various properties with default values. Update these properties
17   // when the user taps a FloatingActionButton.
18   double _width = 50;
19   double _height = 50;
20   Color _color = Colors.green;
21   BorderRadiusGeometry _borderRadius = BorderRadius.circular(8);
```



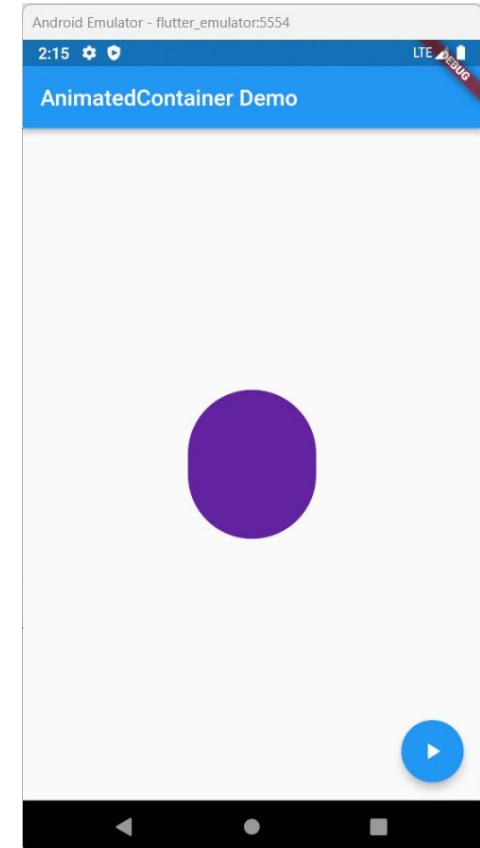
Animated Container

```
22 @override
23 Widget build(BuildContext context) {
24   return MaterialApp(
25     home: Scaffold(
26       appBar: AppBar(
27         title: const Text('AnimatedContainer Demo'),
28       ), // AppBar
29       body: Center(
30         child: AnimatedContainer(
31           // Use the properties stored in the State class.
32           width: _width,
33           height: _height,
34           decoration: BoxDecoration(
35             color: _color,
36             borderRadius: _borderRadius,
37           ), // BoxDecoration
38           // Define how long the animation should take.
39           duration: const Duration(seconds: 1),
40           // Provide an optional curve to make the animation feel smoother.
41           curve: Curves.fastOutSlowIn,
42         ), // AnimatedContainer
43       ), // Center
```



Animated Container

```
44     ), // Center
45     floatingActionButton: FloatingActionButton(
46       // When the user taps the button
47       onPressed: () {
48         // Use setState to rebuild the widget with new values.
49         setState(() {
50           // Create a random number generator.
51           final random = Random();
52
53           // Generate a random width and height.
54           _width = random.nextInt(300).toDouble();
55           _height = random.nextInt(300).toDouble();
56
57           // Generate a random color.
58           _color = Color.fromRGBO(
59             random.nextInt(256),
60             random.nextInt(256),
61             random.nextInt(256),
62             1,
63           ); // Color.fromRGBO
64
65           // Generate a random border radius.
```



Animated Container

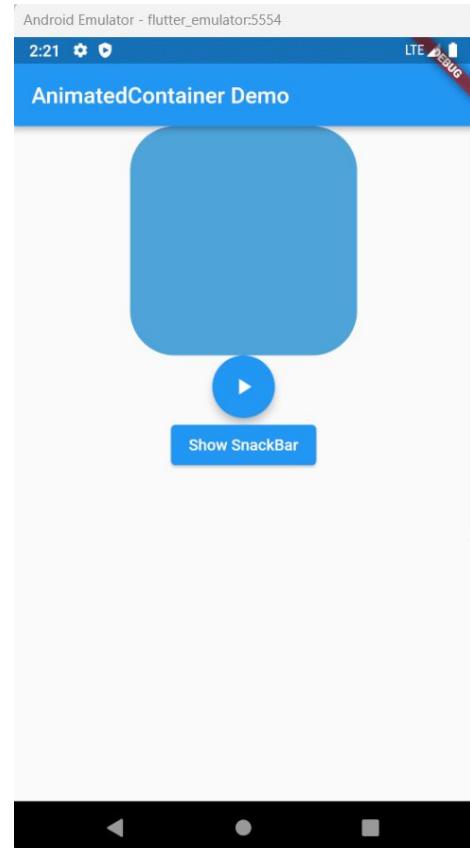
```
64         // Generate a random border radius.
65         _borderRadius =
66             BorderRadius.circular(random.nextInt(100).toDouble());
67     });
68     ),
69     child: const Icon(Icons.play_arrow),
70 ), // FloatingActionButton
71 ), // Scaffold
72 ); // MaterialApp
73 }
74 }
75 }
```

Animated Container

```
class SnackBarPage extends StatelessWidget {
  const SnackBarPage({super.key});

  @override
  Widget build(BuildContext context) {
    return Center(
      child: ElevatedButton(
        onPressed: () {
          final snackBar = SnackBar(
            content: const Text('Yay! A SnackBar!'),
            action: SnackBarAction(
              label: 'Undo',
              onPressed: () {
                // Some code to undo the change.
              },
            ), // SnackBarAction
          ); // SnackBar

          // Find the ScaffoldMessenger in the widget tree
          // and use it to show a SnackBar.
          ScaffoldMessenger.of(context).showSnackBar(snackBar);
        },
        child: const Text('Show SnackBar'),
      ), // ElevatedButton
    ); // Center
  }
}
```



Exercícios

- 1) Você foi contratado pela empresa SM Mobile para desenvolver um aplicativo da empresa SM Consultoria onde cada tela deve ter um tipo de serviço: Projeto, Consultoria, Elaboração de laudo. Colocar imagem do tipo de serviço e valor de cada serviço
- 2) Criar um aplicativo utilizando o Animated Container
- 3) Criar um aplicativo com login que calcule o valor da conta de energia elétrica de acordo com o consumo. Valor do Kw/h 0,65
- 4) Criar um aplicativo com uma lista de máquinas disponíveis em uma indústria como torno, fresa, furadeira. Tempo de utilização, consumo em W e calcular o consumo de energia de cada equipamento

Exercícios

Subir o código no Git Hub e postar o link nesse forms

https://docs.google.com/forms/d/1Aba5_aDYsAaMNXWXWnR13aTeibTSy0h74ILLC5sF4Xw/edit

Obrigado!

Prof. Me Daniel Vieira

Email: danielvieira2006@gmail.com

Linkedin: Daniel Vieira

Instagram: Prof daniel.vieira95

