



Animando seu aplicativo

Aprendendo os conceitos de
animações

Cronograma

- Porque usar animações?
- Os 3 pilares da animação
- Interligando os pontos
- Como tirar do papel uma animação?

Porque utilizar animações?

Os 3 pilares de uma animação

Animation Controller



Animation



Atributo

Animation Controller

SingleTickerProviderStateMixi
n

```
class _LoginFormState  
extends State<LoginForm>  
with SingleTickerProviderStateMixin{
```

Tick

```
class _LoginFormState  
extends State<LoginForm>  
with TickerProviderStateMixin{
```

Animation

- Tween
- TweenSequence
- SizeTween
- StepTween

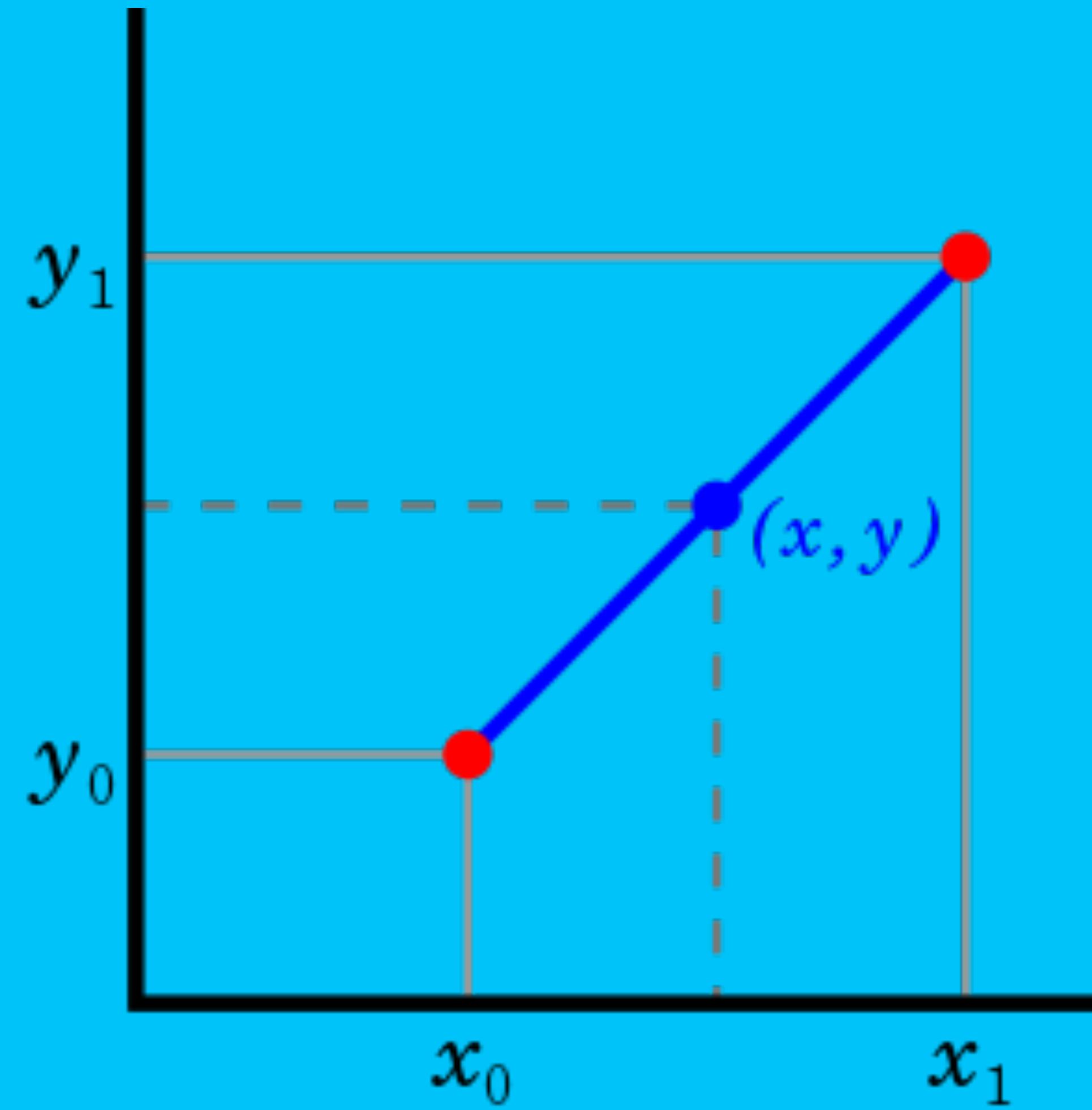
Atributo

- Height, Width, Opacity
- Transform
 - Scale
 - Rotate
 - Translate
- Color.lerp (ColorA, ColorB, animationController)

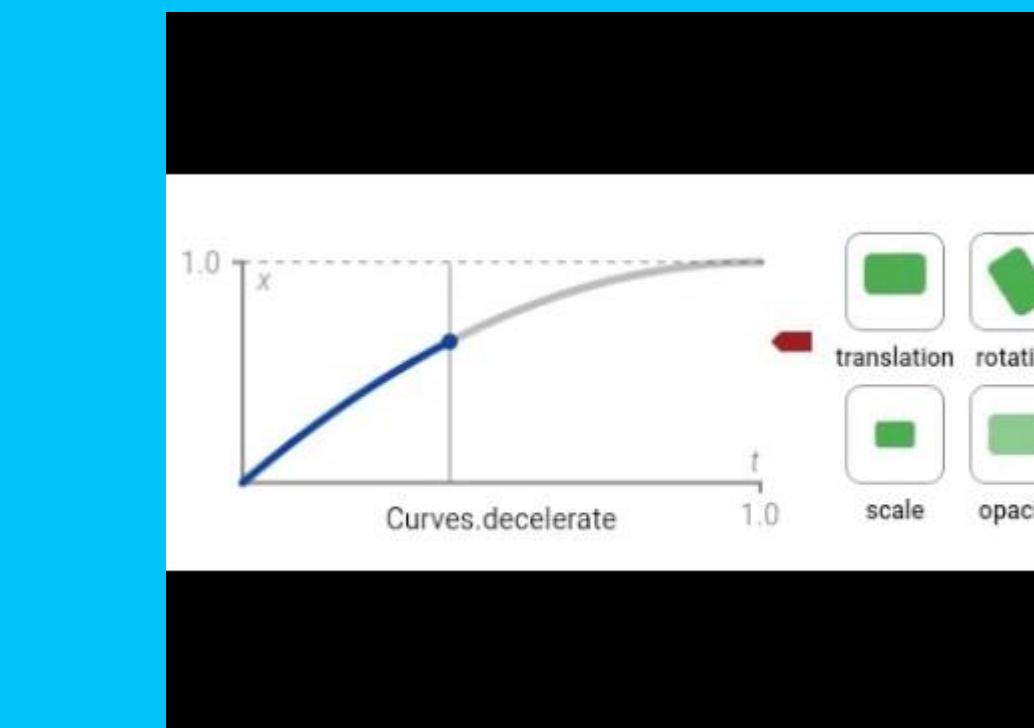
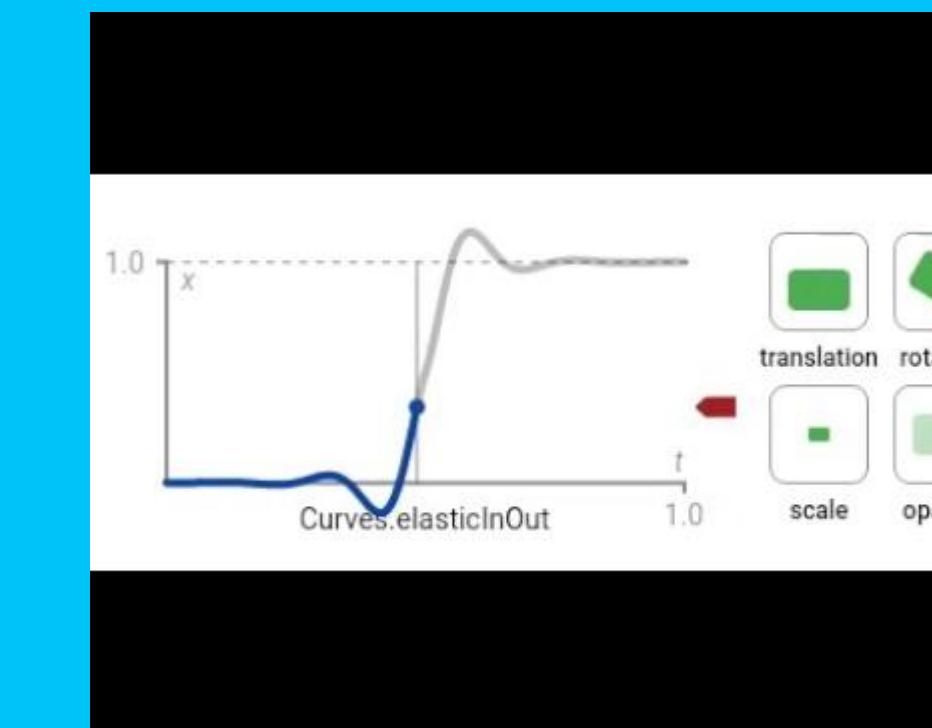
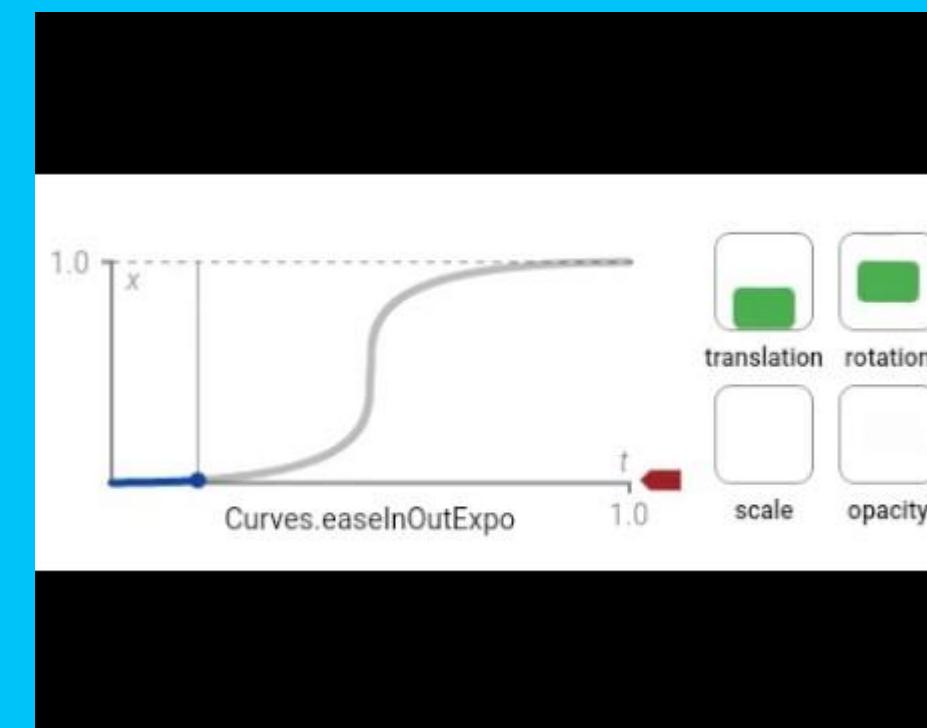
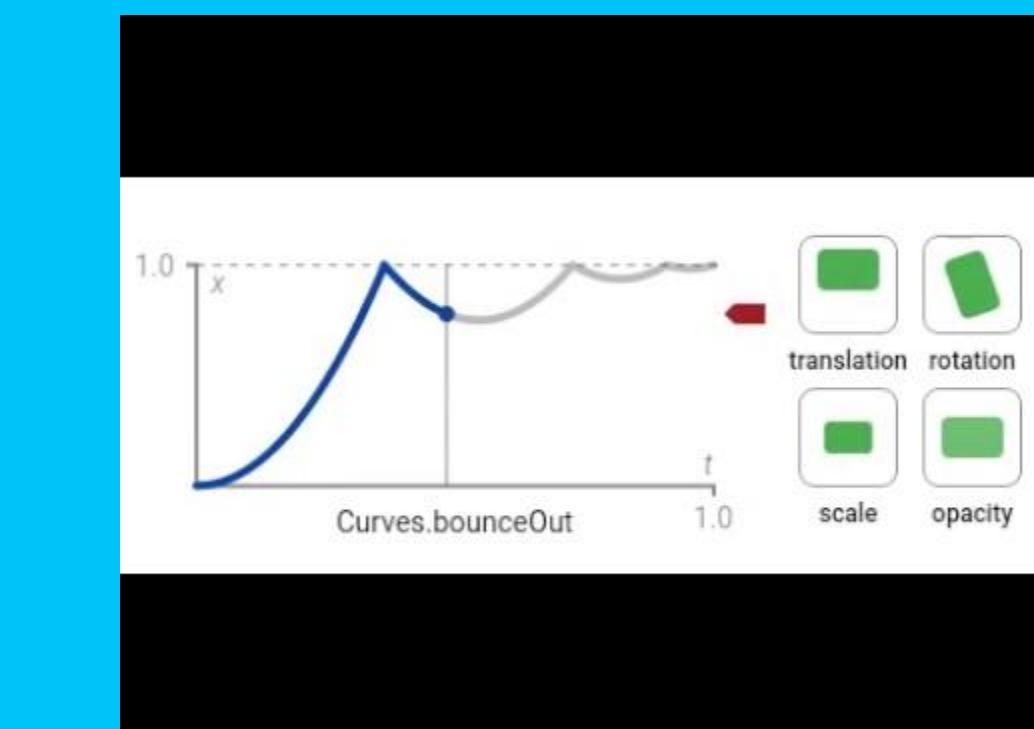
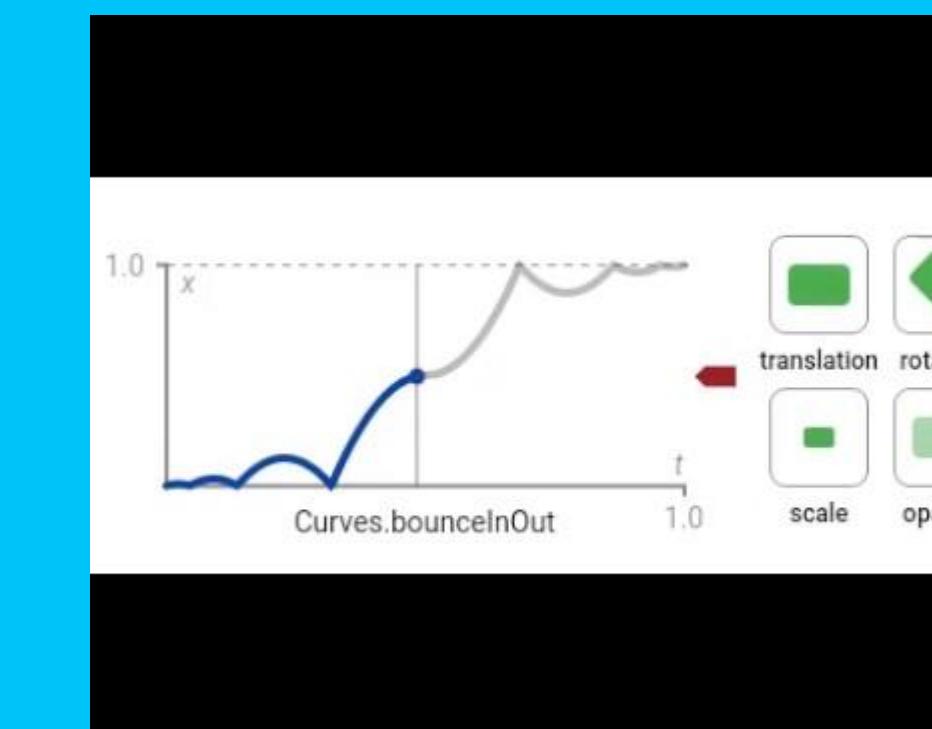
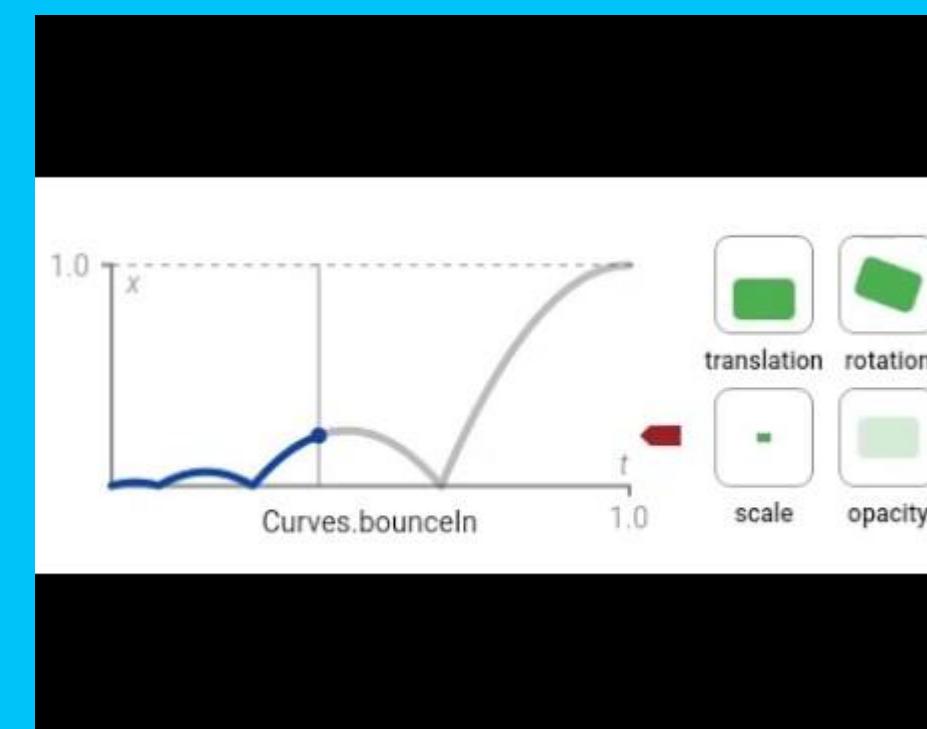
Animated Card



Interpolação linear



Curvas de animação



[Link das curvas](#)

Tween

```
opacity = Tween(begin: 0.0, end: 1.0).
```

TweenSequence

```
TweenSequence(<TweenSequenceItem>[  
    TweenSequenceItem(tween: Tween(begin: 1.0, end: 0.0), weight: 0.4),  
    TweenSequenceItem(tween: Tween(begin: 0.0, end: 0.5), weight: 0.6)  
]); // <TweenSequenceItem>[] // TweenSequence
```

CurvedAnimation

```
CurvedAnimation(  
  parent: widget.animationController,  
  curve: Interval(0.5, 1.0))); //  
}  
}
```

Curves

```
animate(  
    CurvedAnimation(  
        parent: widget.animationController,  
        curve: Interval(0.5, 1.0, curve: Curves.easeInSine)));
```

Interval



```
CurvedAnimation(  
  parent: widget.animationController,  
  curve: Interval(0.5, 1.0))); //  
}
```

AnimationController

```
AnimationController animationController;  
void initAnimation(){  
    animationController = AnimationController  
        (vsync: this,duration: Duration(seconds: 2));  
}
```

Controlando sua animação

```
animationController.forward();
```

```
animationController.reverse();
```

Animated Builder

```
AnimatedBuilder(  
    animation: controller,  
    builder: (context, animation){  
        return Container(  
            height: height.value,  
        ); // Container  
    },  
); // AnimatedBuilder
```

Como analisar uma animação?

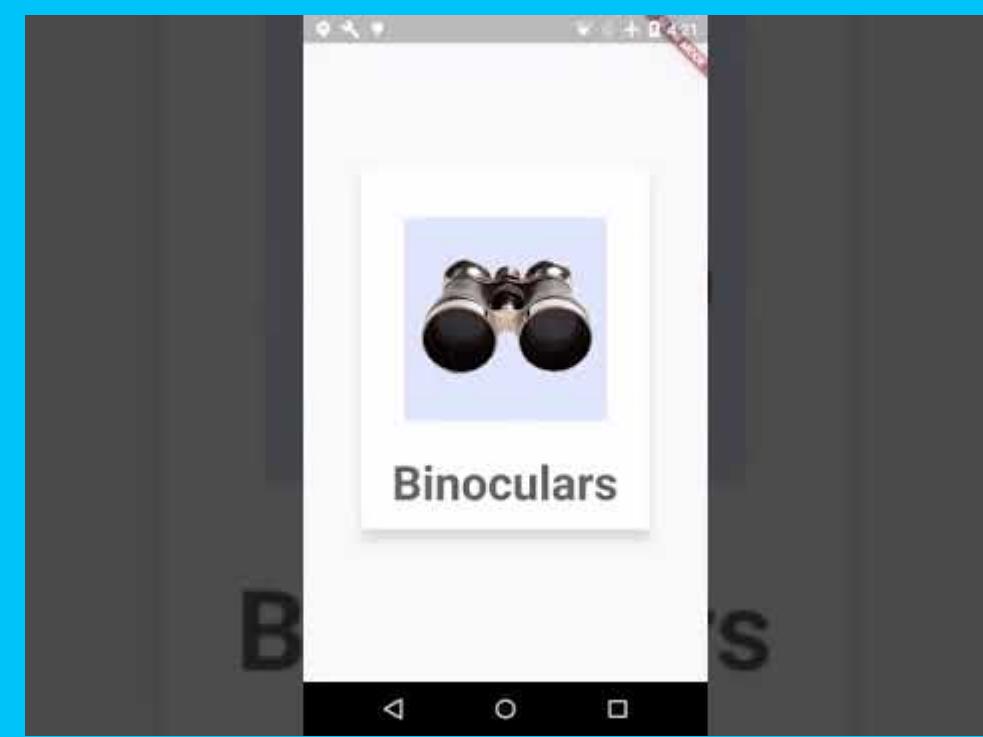
- Observar o fluxo,
analisar quantas animações compõem o conjunto
- Descrever todas as animações separadas
- Implementar cada animação
- Juntar todas e trabalhar com o Interval

Como analisar uma animação?

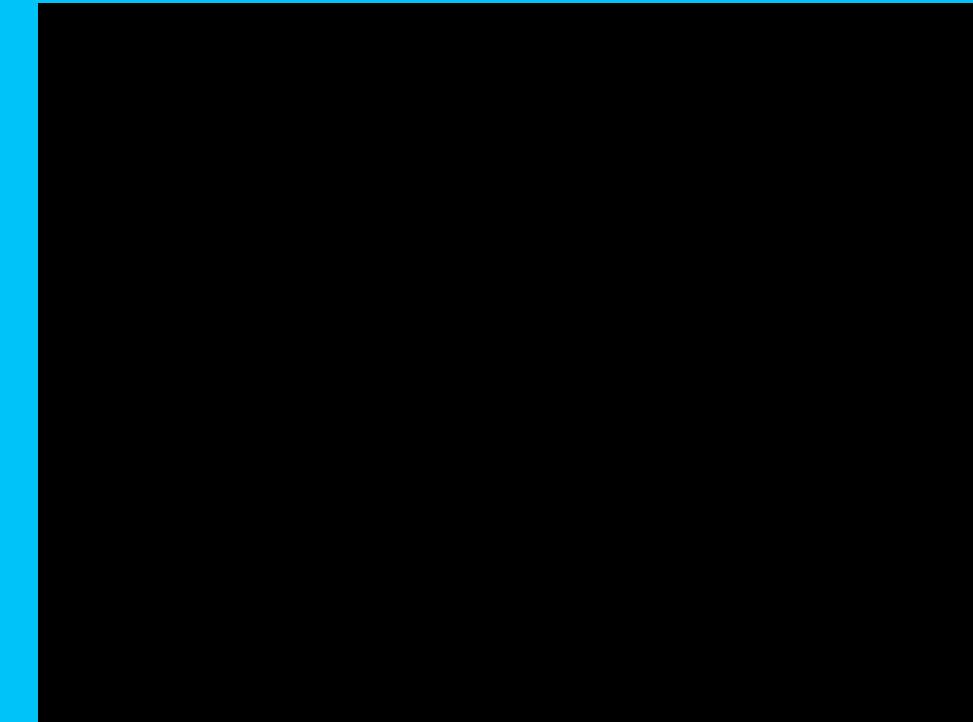


Animações implícitas

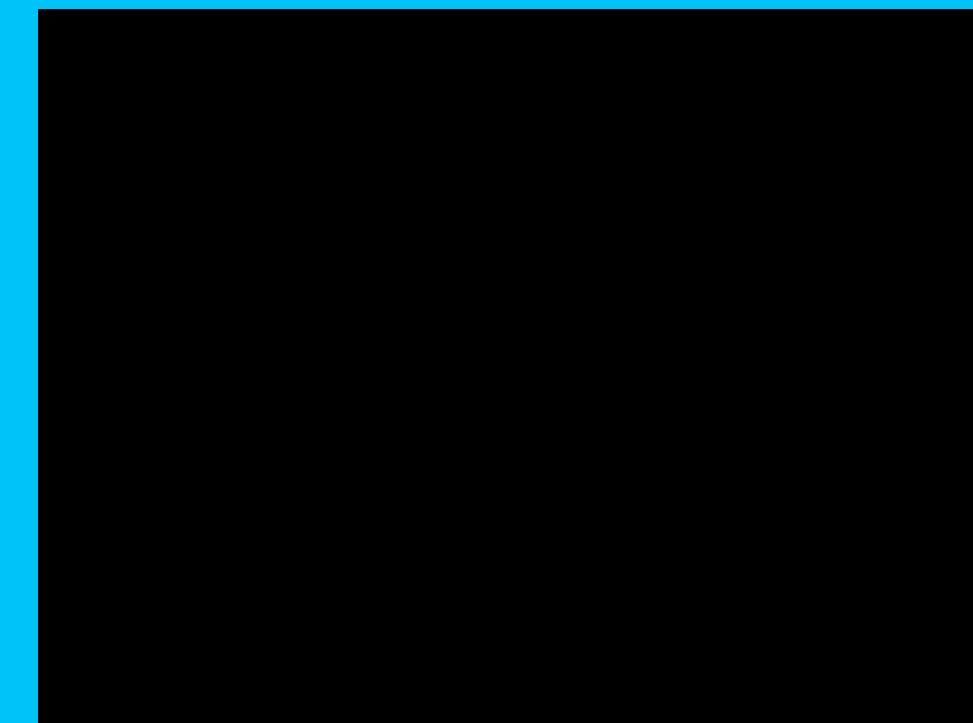
- Hero



- Animated Container

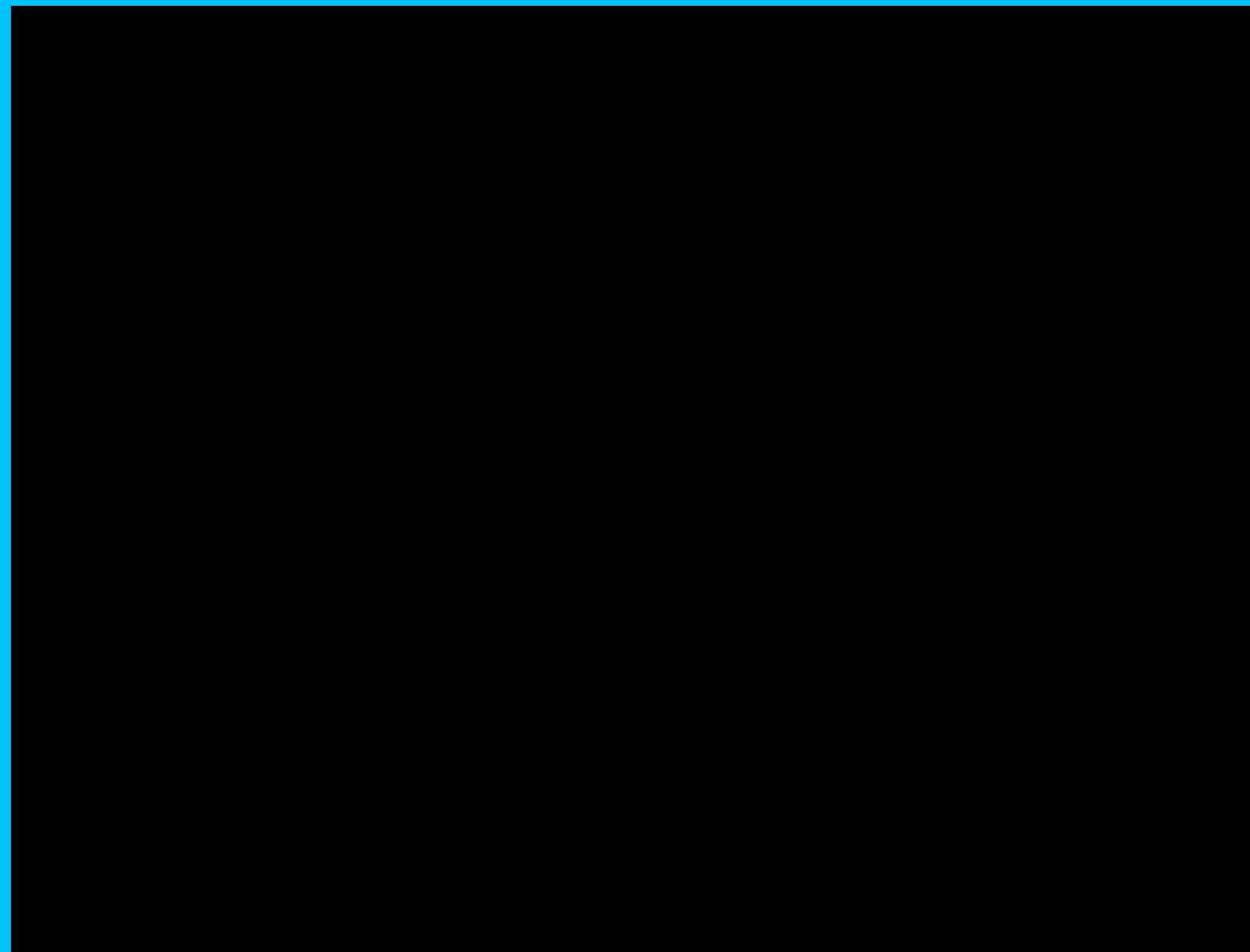


- Animated Opacity



[Link das animações](#)

Animações com Flare



Flutter Talks 2019

Success Check

DESIGN ANIMATE

HIERARCHY DRAW ORDER

Artboard

Guido Rosso's file. To make changes, fork it to Your Files.

FORK

Artwork License: CC BY Attribution

VIEW LICENSE

Artboard

00:00:47

AUTOKEY

Untitled

Check Icon Group

Check

Particle Line Left

Particle Line Right

FPS 60 00:10:00

00:00f 30f 01:00f 30f 02:00f 30f 03:00f

KEY INTERPOLATION

Type

No Key Selected

This screenshot shows a design and animation interface for a Flutter icon. The main view displays a green circular icon with a white checkmark and a small crosshair, surrounded by small green dots, centered on a white artboard with a red border. The top navigation bar includes tabs for 'DESIGN' and 'ANIMATE'. The left sidebar shows 'HIERARCHY' and 'DRAW ORDER' sections, with 'Artboard' selected. Below the main canvas, there is a timeline from 00:00:47 to 00:10:00 at 60 FPS. The timeline shows keyframes for several layers: 'Check Icon Group' (multiple diamond shapes), 'Check' (a single diamond shape), 'Particle Line Left' (multiple diamond shapes), and 'Particle Line Right' (multiple diamond shapes). The bottom right panel shows 'KEY INTERPOLATION' settings with 'Type' set to 'Linear' and a note 'No Key Selected'.



Obrigado!



gabul.dev



[gabuldev](https://gabuldev.in)

GitHub

gabul.dev