

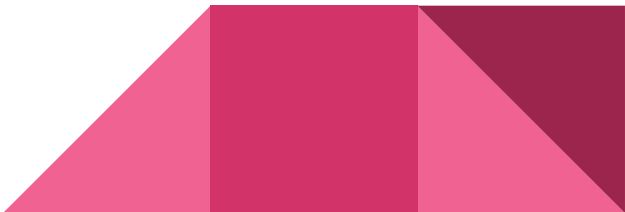
Minicurso de React-Native

com @felipemfp e @chicobentojr



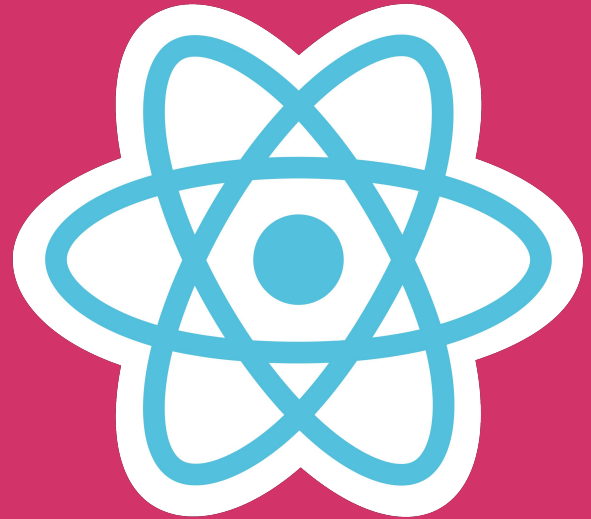
Agenda

- Introdução ao React
- Introdução ao React-Native
- Componentes Inclusos
- Componentes Open-Source
- Expo
- Prática



Introdução ao React

Apenas uma
biblioteca



Tudo são componentes

TfL Tube Tracker

Navigation: < > <Network /> Search:

Bakerloo Line

Stations **<Line />** Go

Central Line

Stations **<Line />** Go

Circle Line

Stations **<Line />** Go

District Line

Stations **<Line />** Go

Hammersmith & City Line

Stations **<Line />** Go

Jubilee Line

Stations **<Line />** Go

Metropolitan Line

Stations **<Line />** Go

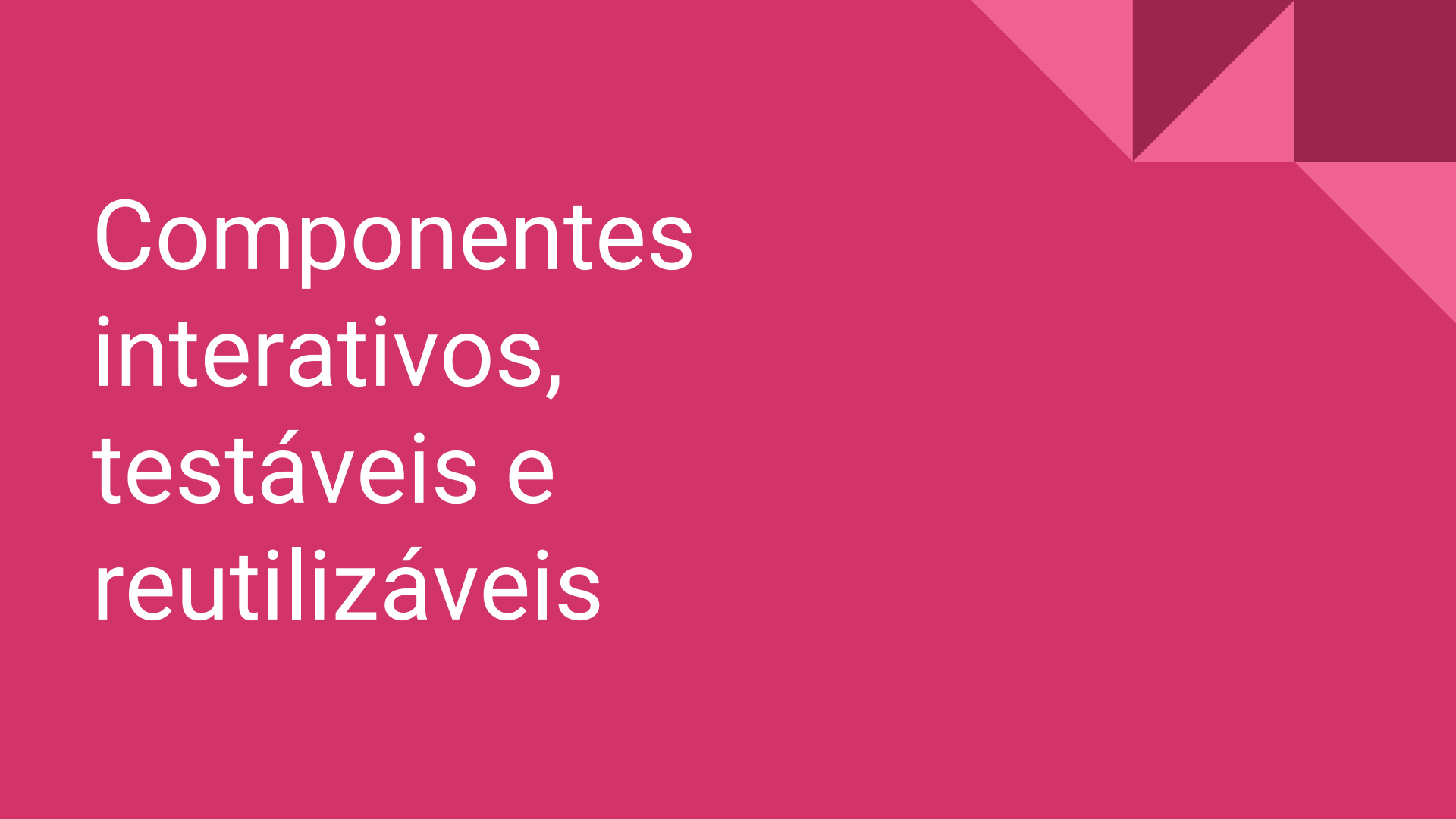
Station Name **<Predictions />**

Platform 1 **<DepartureBoard />**

Destination	Due	Current location
White City	0:00	At Platform
Ealing Broadway	<Trains />	Holland Park
West Ruislip	4:30	Notting Hill Gate
Ealing Broadway	6:00	Queensway

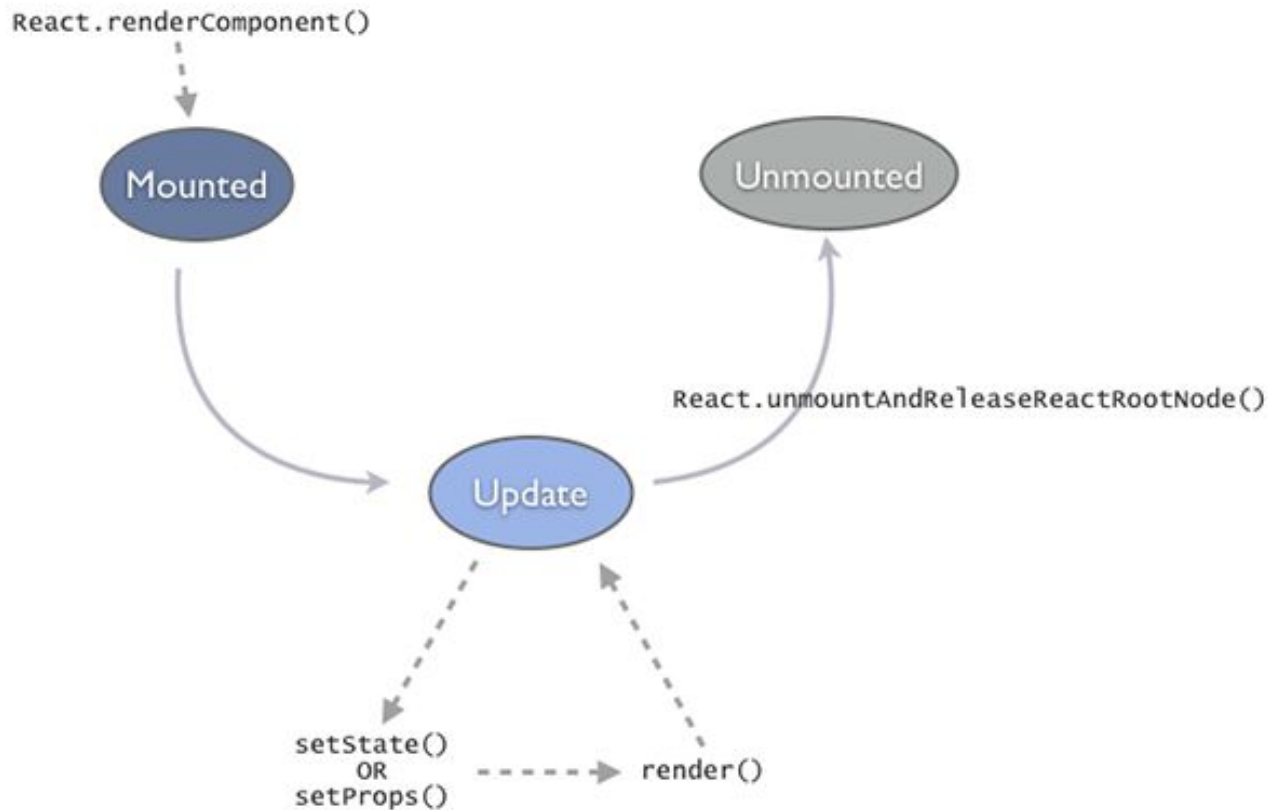
Platform 2 **<DepartureBoard />**

Destination	Due	Current location
White City	0:00	At Platform
Ealing Broadway	<Trains />	Holland Park
West Ruislip	4:30	Notting Hill Gate
Ealing Broadway	6:00	Queensway

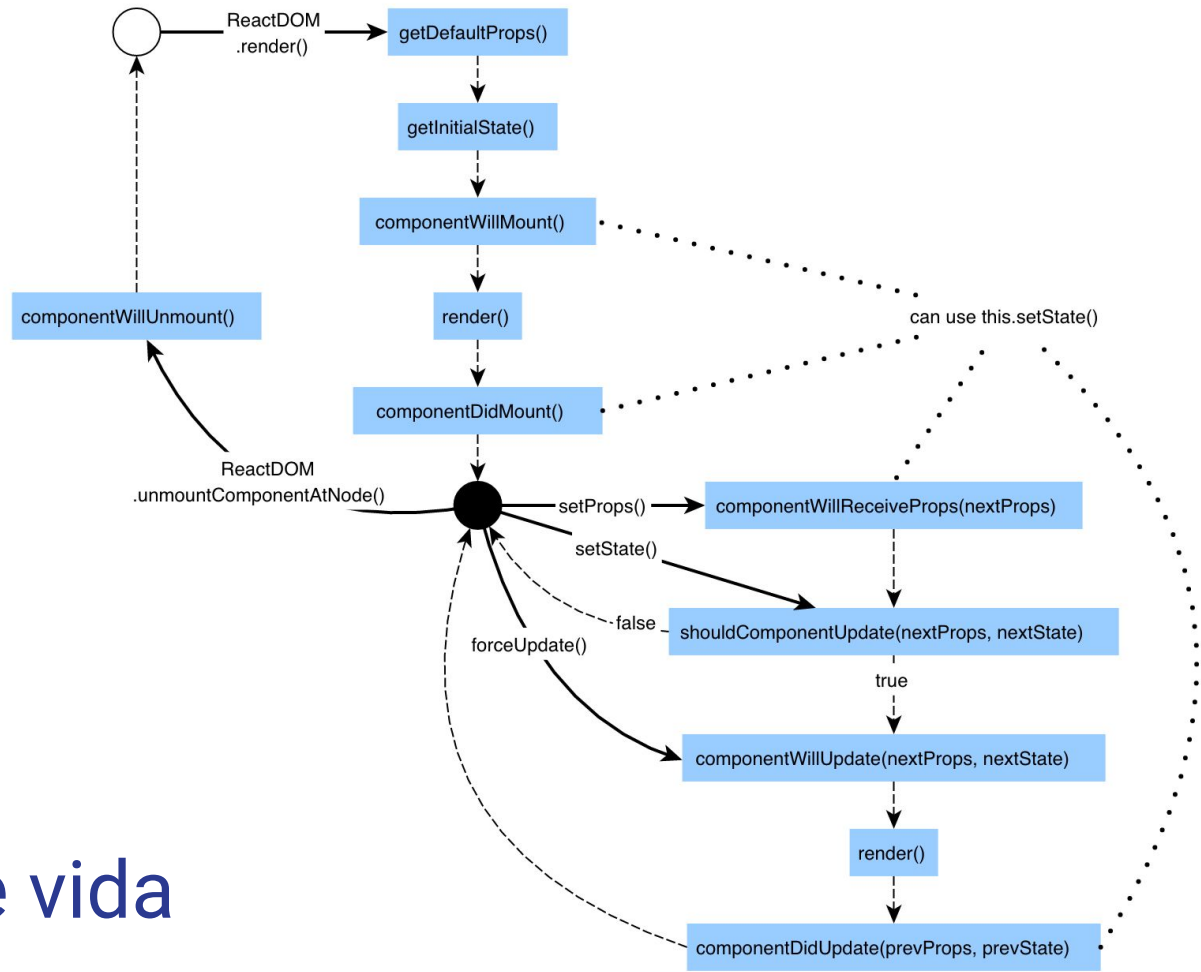


Componentes
interativos,
testáveis e
reutilizáveis

Ciclo de vida simplificado

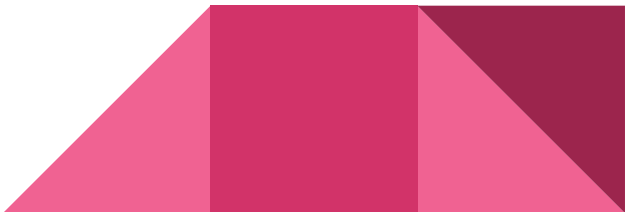


Ciclo de vida



Como escrevemos componentes em React?

- Componentes são objetos
- `render()` e JSX

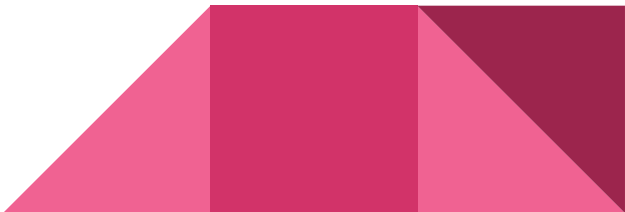


JSX

```
var Hello = React.createClass({  
  render: function() {  
    return (  
      <div>  
        <h1>Hello from Kode</h1>  
        <p>Maybe you prefer a hello from Adele</p>  
      </div>  
    );  
  }  
});  
  
ReactDOM.render(<Hello/>, document.getElementById('react-target'));
```

Como escrevemos componentes em React?

- Possuem *props* e *state*
- Possuem métodos:
 - *setState()*
 - *getInitialState()*
 - *getDefaultProps()*
 - [...]




props

```
1 import React from 'react'
2 import ReactDOM from 'react-dom'
3
4 var ComplimentMachine = React.createClass({
5   render: function () {
6     return (
7       <div>
8         <h1>What's your name?</h1>
9         {this.props.compliment}
10      </div>
11    )
12  }
13 })
14
15 ReactDOM.render(
16   <ComplimentMachine compliment="You're a great listener" />,
17   document.getElementById('root')
18 )
```

state

```
var CommentBox = React.createClass({
  loadFromServer: function() {
    return [
      {id: 1, author: "Stephan Schmidt", text: "This is one comment"},
      {id: 2, author: "Jens Schumann", text: "This is *another* comment"}
    ];
  },
  getInitialState: function() {
    return {data: []};
  },
  componentDidMount: function() {
    // Ajax code ....
    this.setState({data: this.loadFromServer()});
  },
  render: function() {
    return (
      <div className="commentBox">
        <h1>Comments</h1>
        <CommentList data={this.state.data} />
      </div>
    );
  }
});
```



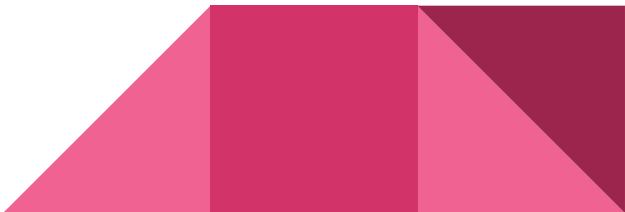
The diagram illustrates the state management process in the provided code. A bracket on the left side of the `loadFromServer` function and the `componentDidMount` function points to the `this.setState` call, indicating that `loadFromServer` is used to populate the state. Another bracket on the right side points from the `this.state.data` property access in the `render` method back to the `setState` call, showing that the state is used to render the UI.

Componentes *stateless*

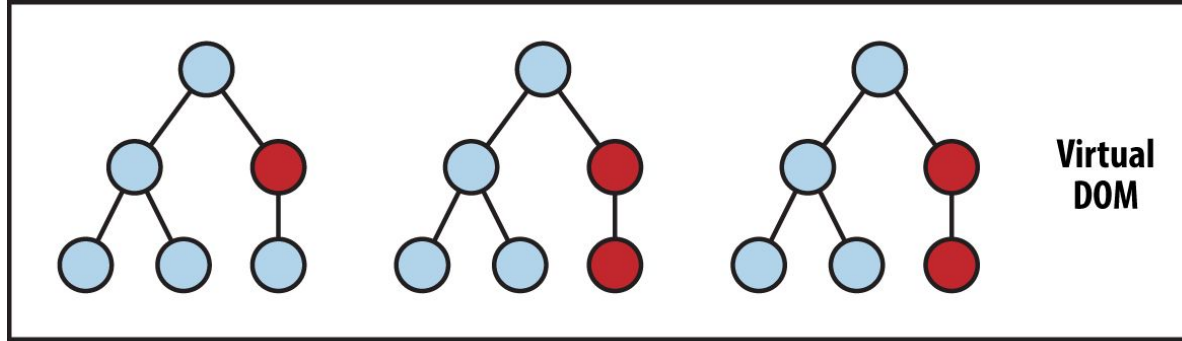
```
const Intro = () => {  
  return <p>Hi there...</p>  
}
```

Como escrevemos componentes em React?

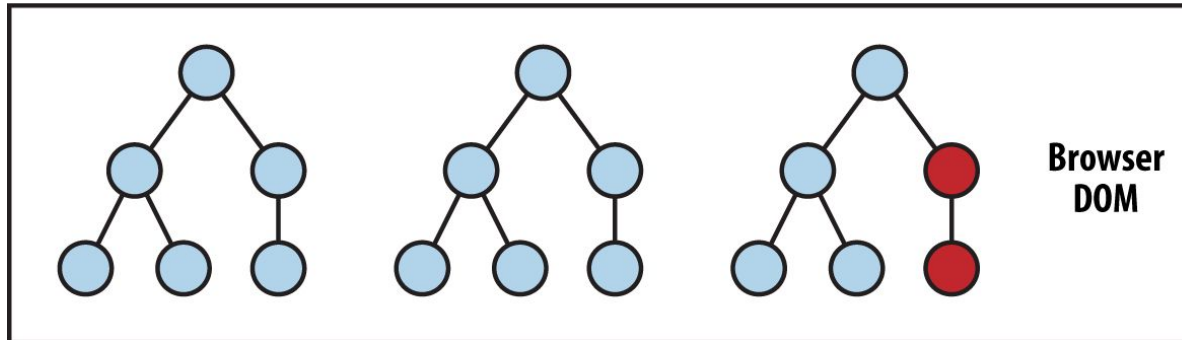
- Dados fluem numa única direção
- Renderização conforme necessário (Virtual DOM)



Virtual DOM



State Change → Compute Diff → Re-render



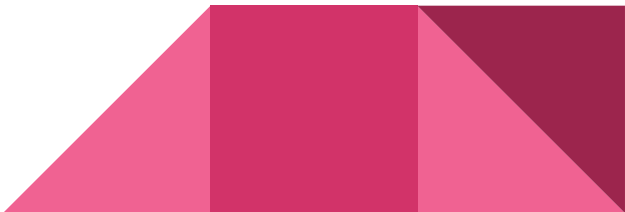
Introdução ao React-Native



React-Native é React

Introdução ao React-Native

- Componentes são renderizados como componentes nativos
- Os mesmos conceitos são utilizados para construir aplicações
- Suporte oficial para iOS e Android
- Suporte ainda para Universal Windows



Introdução ao React-Native

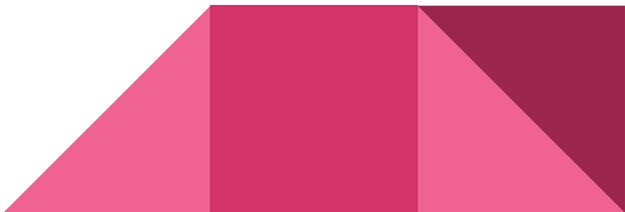
- Ciclo de vida (*props*, *state*...) do componentes são o mesmo
- Continuamos usando JSX



CSS é um pouco
diferente

Adicionando estilo aos componentes nativos

- “CSS” como *prop*
- StyleSheet é quase CSS

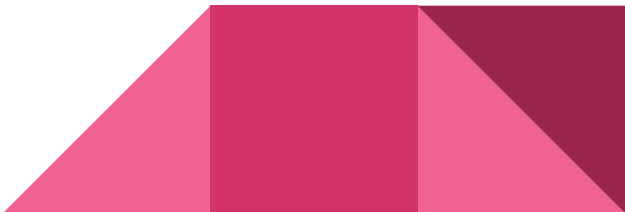


StyleSheet

```
var styles = StyleSheet.create({  
  base: { width: 38, height: 38 },  
  background: { backgroundColor: '#222222' },  
  active: { borderWidth: 2, borderColor: '#00ff00' },  
});  
  
<Text style={styles.base} />  
<View style={styles.background} />
```

Construindo layout com React-Native

- O mesmo Flexbox da web
- Com diferentes *defaults*:
 - flexDirection: column
 - alignItems: stretch



Componentes incluidos

Componentes incluidos

- **ActivityIndicator**
- **Button**
- CheckBox
- DatePickerIOS
- DrawerLayoutAndroid
- **FlatList**
- **Image**
- KeyboardAvoidingView
- ListView
- MaskedViewIOS
- Modal
- NavigatorIOS
- Picker
- Picker.Item
- PickerIOS
- ProgressBarAndroid
- ProgressViewIOS
- RefreshControl
- **ScrollView**
- SectionList
- SegmentedControlIOS
- Slider
- SnapshotViewIOS
- StatusBar

Componentes incluidos

- Switch
- TabBarIOS
- TabBarIOS.Item
- **Text**
- **TextInput**
- ToolbarAndroid
- TouchableHighlight
- TouchableNativeFeedback
- **TouchableOpacity**
- TouchableWithoutFeedback
- **View**
- ViewPagerAndroid
- VirtualizedList
- WebView

API inclusas

- AccessibilityInfo
- ActionSheetIOS
- Alert
- AlertIOS
- Animated
- AnimatedValue
- AnimatedValueXY
- AppRegistry
- AppState
- AsyncStorage
- BackAndroid
- BackHandler
- CameraRoll
- Clipboard
- DatePickerAndroid
- Dimensions
- Easing
- Geolocation
- ImageEditor
- ImagePickerIOS
- ImageStore
- InteractionManager
- Keyboard
- LayoutAnimation

API inclusas

- Linking
- NetInfo
- PanResponder
- PermissionsAndroid
- PixelRatio
- PushNotificationIOS
- Settings
- Share
- StatusBarIOS
- StyleSheet
- Systrace
- TimePickerAndroid
- ToastAndroid
- Vibration
- VibrationIOS
- Layout Props
- Picker Style Props
- Shadow Props

Componentes Open-Source

<http://www.awesome-react-native.com/>

Navigation for React Native



[Get Started →](#)

[DOCS](#)[GITHUB](#)[CONNECT](#)[START A PROJECT](#)[MARKET](#)

NativeBase

Essential cross-platform UI components for React Native

100% open source

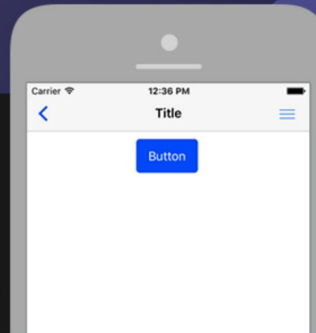
Currently v2.3.2

6,786 stars 731 forks

Without NativeBase

```
var style = StyleSheet.create({
  button: {
    backgroundColor: '#99AAFF',
    borderRadius: 5,
    borderWidth: 1,
    borderColor: '#000033'
  }
});

<TouchableOpacity style={style.button}>
  <Text style={{color: 'white'}}>Click me!</Text>
</TouchableOpacity>
```



With NativeBase

```
<Button>
  <Text>Button</Text>
</Button>
```




React Native Elements UI Toolkit

Home

API ▾

Installation ▾

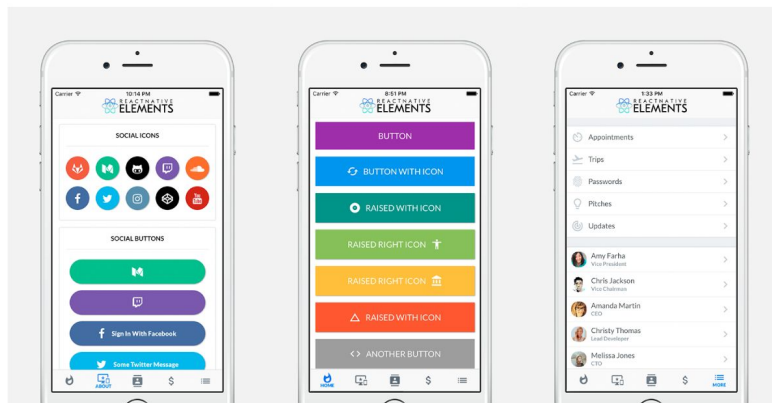
Home



React Native Elements

Cross Platform [React Native](#) UI Toolkitnpm **v0.18.2** downloads **41k/month** cdnjs **v0.17.0** build **passing** slack **7/174**backers **10** sponsors **1** codecov **73%** styled with **prettier**

Table of contents

[Get Started](#)[Installation](#)[Usage](#)[Components Included](#)[Documentation](#)[Demo App](#)[Backers](#)[Sponsors](#)[Roadmap](#)[First Contributors](#)[1.0 Release](#)[Not Started](#)[Contributing](#)

[Introduction](#)

[Android](#)

[iOS/macOS](#)

[React Native](#)

[Web](#)

[After Effects](#)

[Supported After Effects Features](#)

[Other Platforms](#)

[Troubleshooting](#)

[Community Showcase](#)

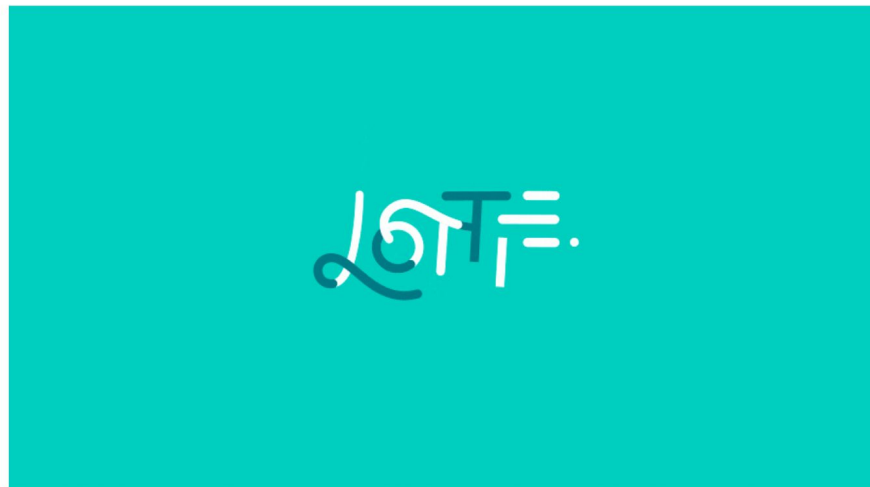
Published with [GitBook](#)

Lottie for [Android](#), [iOS](#), [React Native](#), and [Web](#)



Lottie is a mobile library for Android and iOS that parses [Adobe After Effects](#) animations exported as json with [Bodymovin](#) and renders them natively on mobile and on the web!

For the first time, designers can create **and ship** beautiful animations without an engineer painstakingly recreating it by hand. They say a picture is worth 1,000 words so here are 13,000:

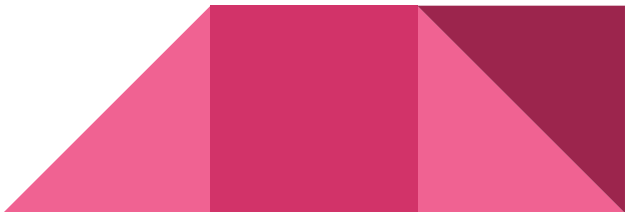




Expo

Expo é React-Native

- Apenas JavaScript
- Fácil prototipagem
- Rápido desenvolvimento



Expo para desenvolvedores e clientes



Expo

Expo Project Produtividade



Expo Client

By Nametag



Prática

github.com/cades-ifrn/minicurso-react-native-gdg-natal

Passo 0 - Iniciar o projeto

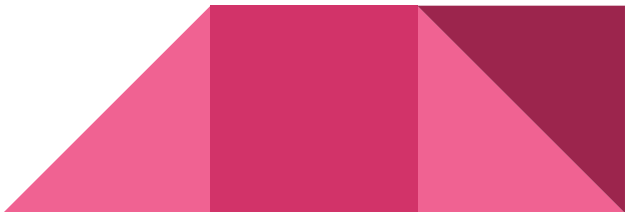
- create-react-native-app pitaco-de-musica
- cd pitaco-de-musica

Com **npm**:

- npm install --save react-navigation axios
- npm run start

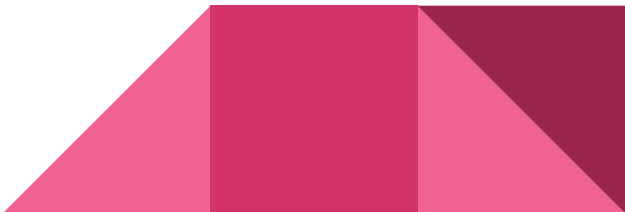
Com **yarn**:

- yarn add react-navigation axios
- yarn start



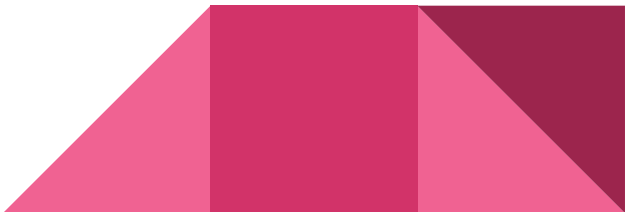
Passo 1 - Adicionar React-Navigation

- Criar componente HomeScreen
- Utilizar componente StackNavigator



Passo 2 - Criar tela de início

- Utilizar componente Image
- Utilizar componente ScrollView
- Utilizar componente Button
- Estilizar componentes

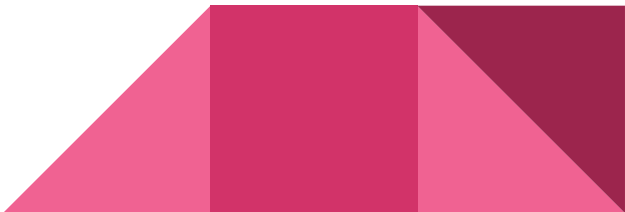


Passo 3 - Navegar entre telas

- Criar componente SearchScreen
- Adicionar SearchScreen no navegador raiz
- Adicionar evento para realizar navegação

Passo 4 - Pesquisar e Apresentar Resultados

- Utilizar componente FlatList
- Criar componente Header
- Criar componente Item
- Carregar resultados



Passo 5 - Melhorar UI e utilizar APIs

- Utilizar Alert API
- Criar componente ResultItem

Passo 6 - Adicionar exploração

- Adicionar evento ao clicar num resultado



Próximos passos

The Complete React Native and Redux Course

<https://goo.gl/2NYB6H>



How to Become a React Native Developer in 2018

<https://goo.gl/ggdZJE>

Obrigado!

Felipe Pontes
@felipemfp

Francisco Bento
@chicobentojr

