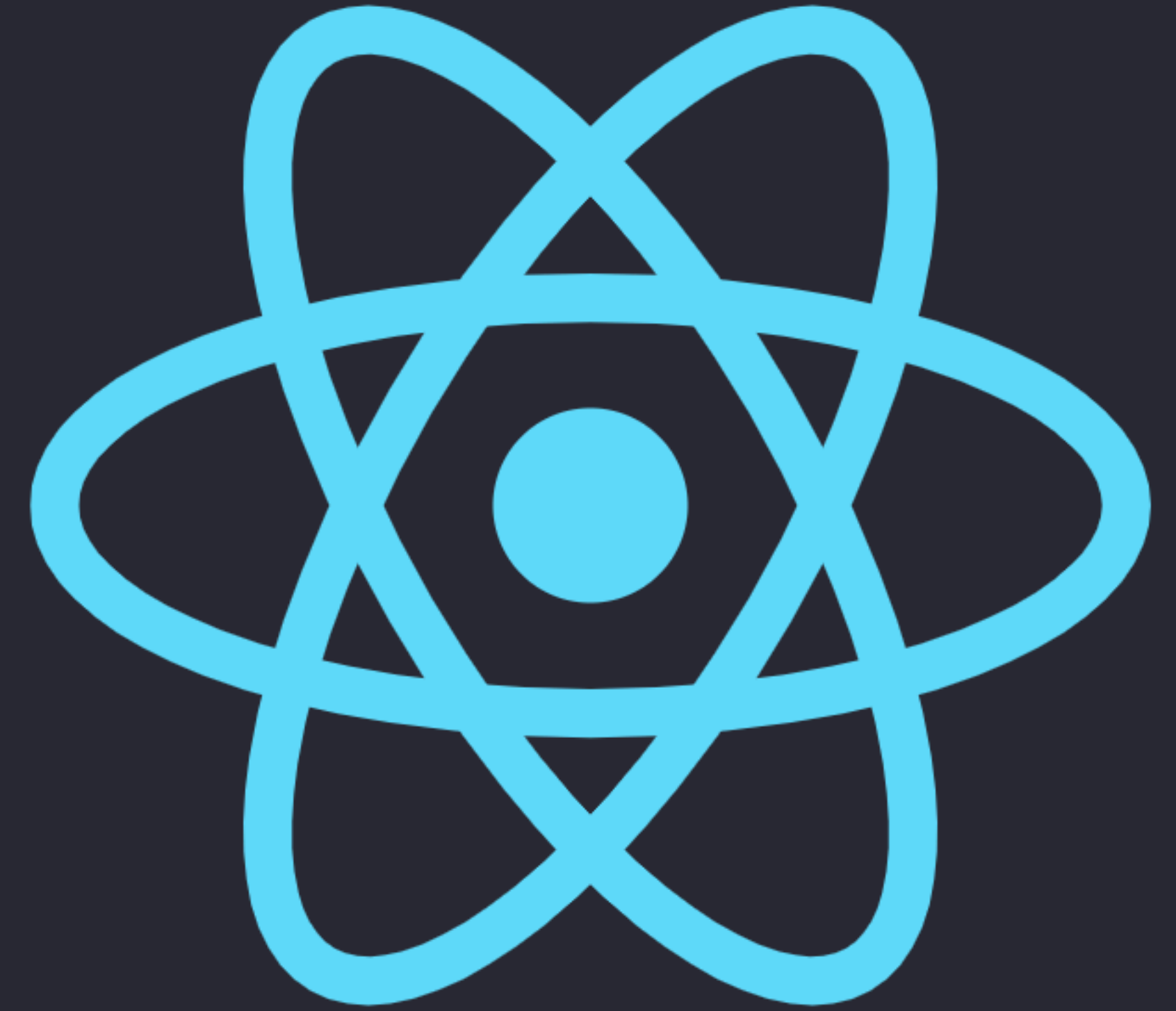
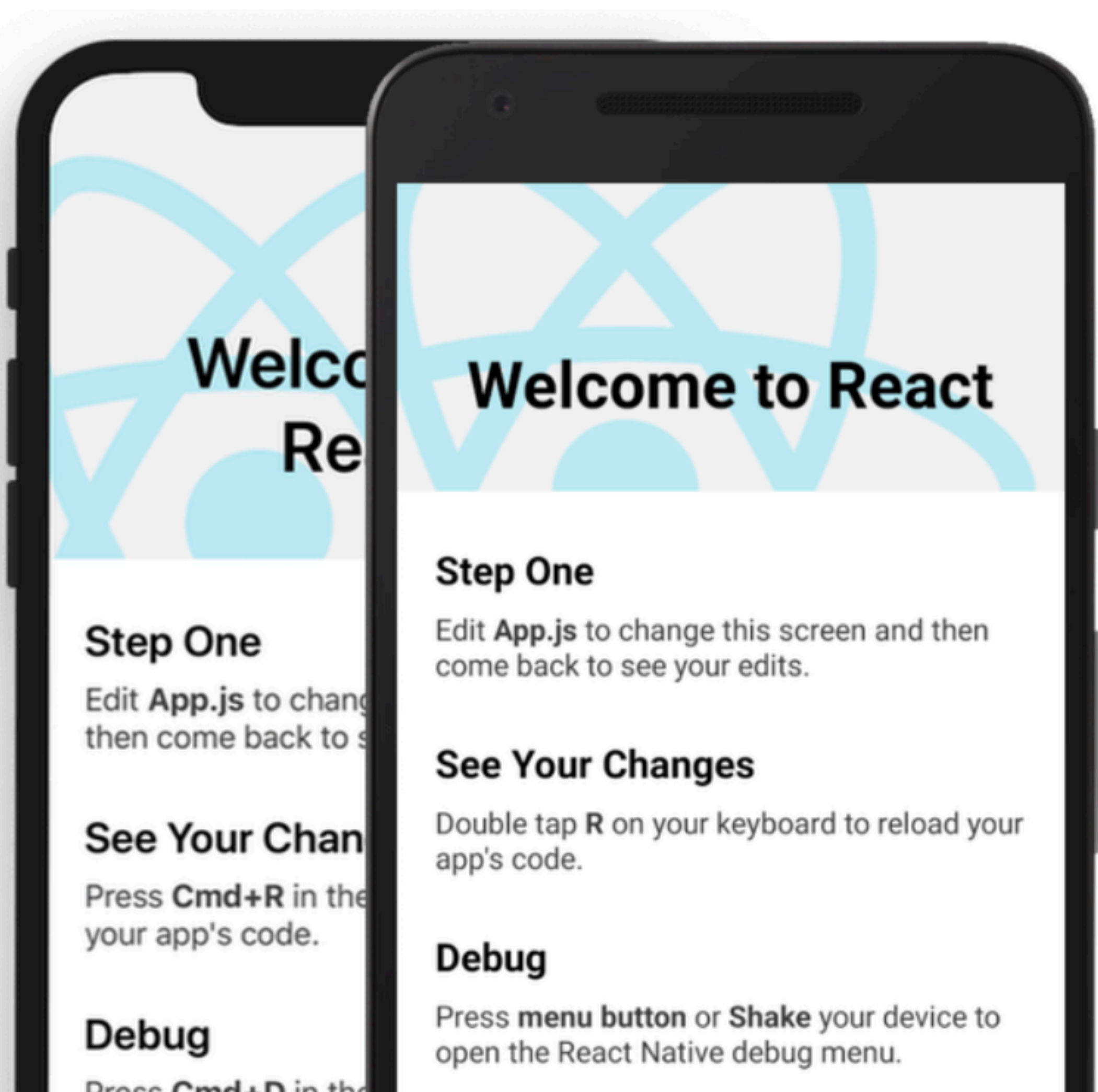


React Native

Aprenda uma vez, escreva em qualquer lugar.



React Native



Crie aplicativos nativos para Android e iOS usando React

React Native combina as melhores partes do desenvolvimento nativo com React, a melhor biblioteca JavaScript da classe para construir interfaces de usuário.

Use um pouco - ou muito . Você pode usar React Native hoje em seus projetos Android e iOS existentes ou pode criar um aplicativo totalmente novo do zero.

React Native

Escrito em JavaScript - renderizado com código nativo

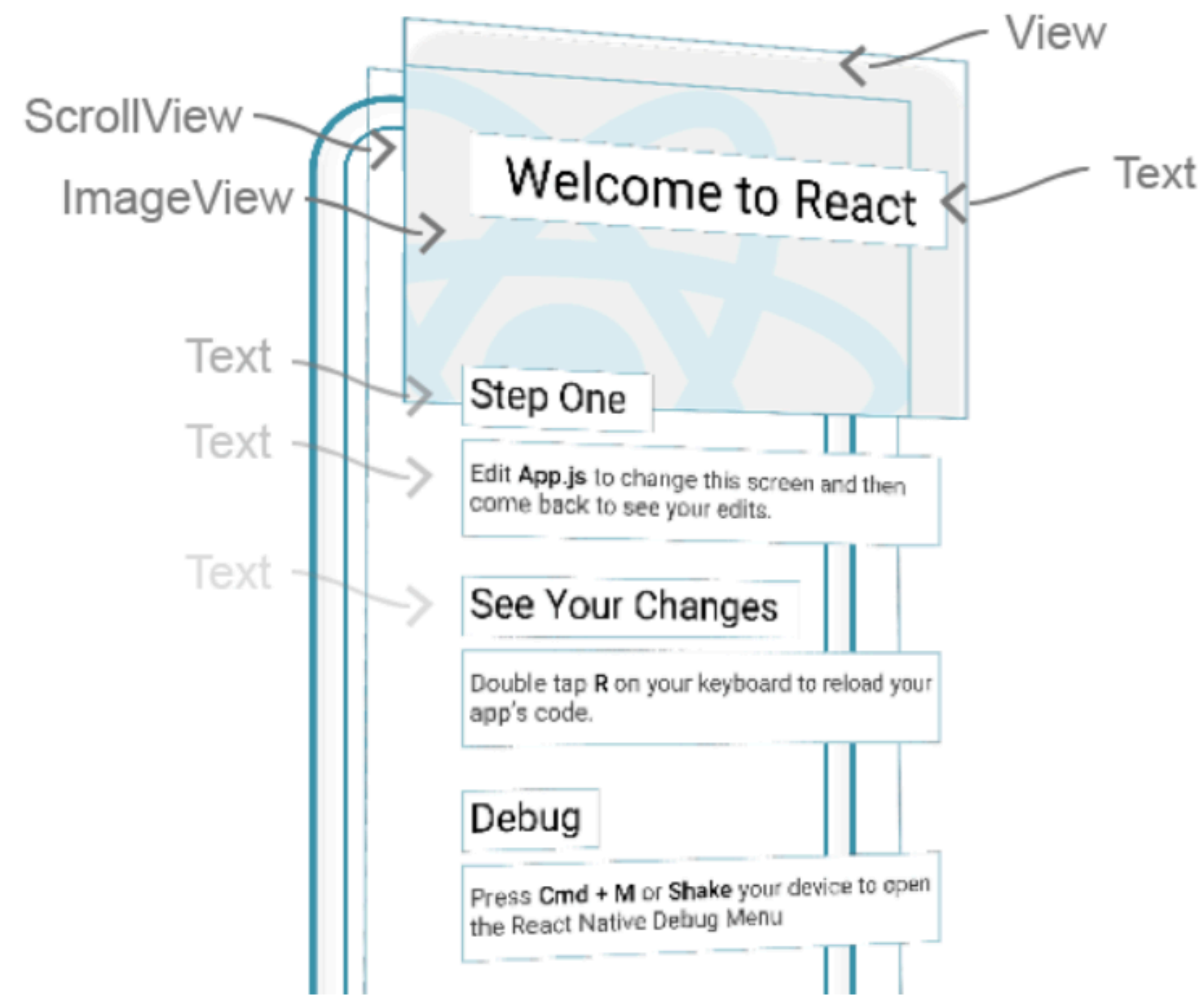
As primitivas do React são renderizadas para a IU da plataforma nativa, o que significa que seu aplicativo usa as mesmas APIs da plataforma nativa que outros aplicativos usam.

Muitas plataformas, um React. Crie versões de componentes específicas da plataforma para que uma única base de código possa compartilhar o código entre as plataformas. Com o React Native, uma equipe pode manter duas plataformas e compartilhar uma tecnologia comum - React.

```
import React from 'react';
import {Text, View} from 'react-native';
import {Header} from './Header';
import {heading} from './Typography';

const WelcomeScreen = () => (
  <View>
    <Header title="Welcome to React Native"/>
    <Text style={heading}>Step One</Text>
    <Text>
      Edit App.js to change this screen and turn it
      into your app.
    </Text>
    <Text style={heading}>See Your Changes</Text>
    <Text>
      Press Cmd + R inside the simulator to reload
      your app's code.
    </Text>
  </View>
);
```

React Native



Desenvolvimento nativo para todos

O React Native permite que você crie aplicativos verdadeiramente nativos e não compromete a experiência dos usuários. Ele fornece um conjunto de plataformas componentes nativos agnósticos como **View**, **Text** e **Image** que o mapeia diretamente para blocos de UI construção nativas da plataforma.

Com suporte do Facebook, orientado pela comunidade

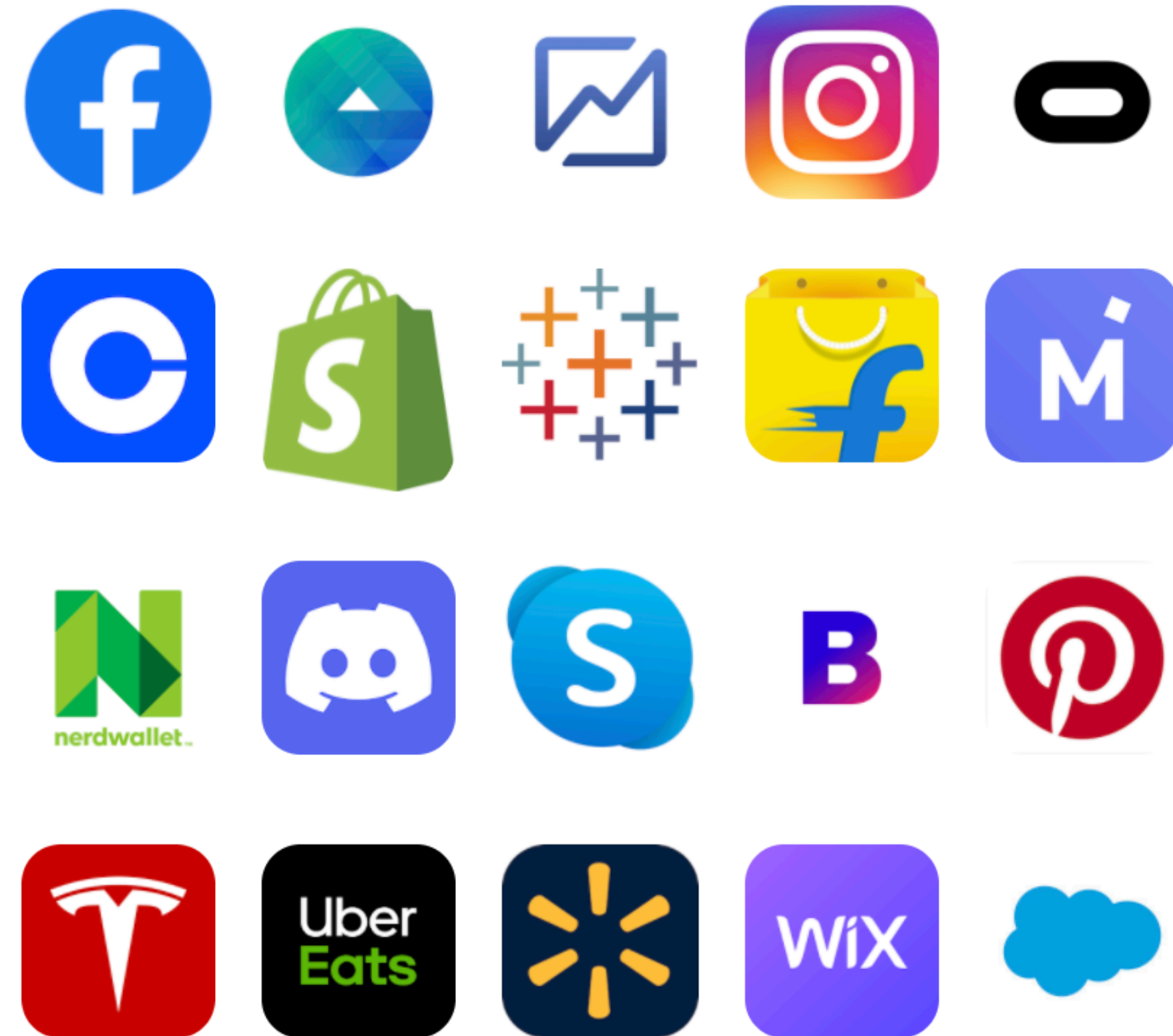


O Facebook lançou o React Native em 2015 e o mantém desde então.

Em 2018, o React Native tinha o segundo maior número de colaboradores para qualquer repositório no GitHub. Hoje, React Native é apoiado por contribuições de indivíduos e empresas ao redor do mundo, incluindo Callstack , Expo , Infinite Red , Microsoft e Software Mansion .

Nossa comunidade está sempre enviando novos projetos interessantes e explorando plataformas além do Android e iOS com repositórios como React Native Windows , React Native macOS e React Native Web .

O React Native está sendo usado em milhares de aplicativos, mas é provável que você já o tenha usado em um destes aplicativos:



Configuração do Ambiente

React Native

Expo CLI Quickstart

Quickstart do React Native CLI

Supondo que você tenha o Node 12 LTS ou superior instalado, você pode usar o npm para instalar o utilitário de linha de comando Expo CLI:

acima do nível do mar

Fio

```
npm install -g expo-cli
```

Em seguida, execute os seguintes comandos para criar um novo projeto React Native chamado "AwesomeProject":

acima do nível do mar

Fio

```
expo init AwesomeProject  
  
cd AwesomeProject  
npm start # you can also use: expo start
```

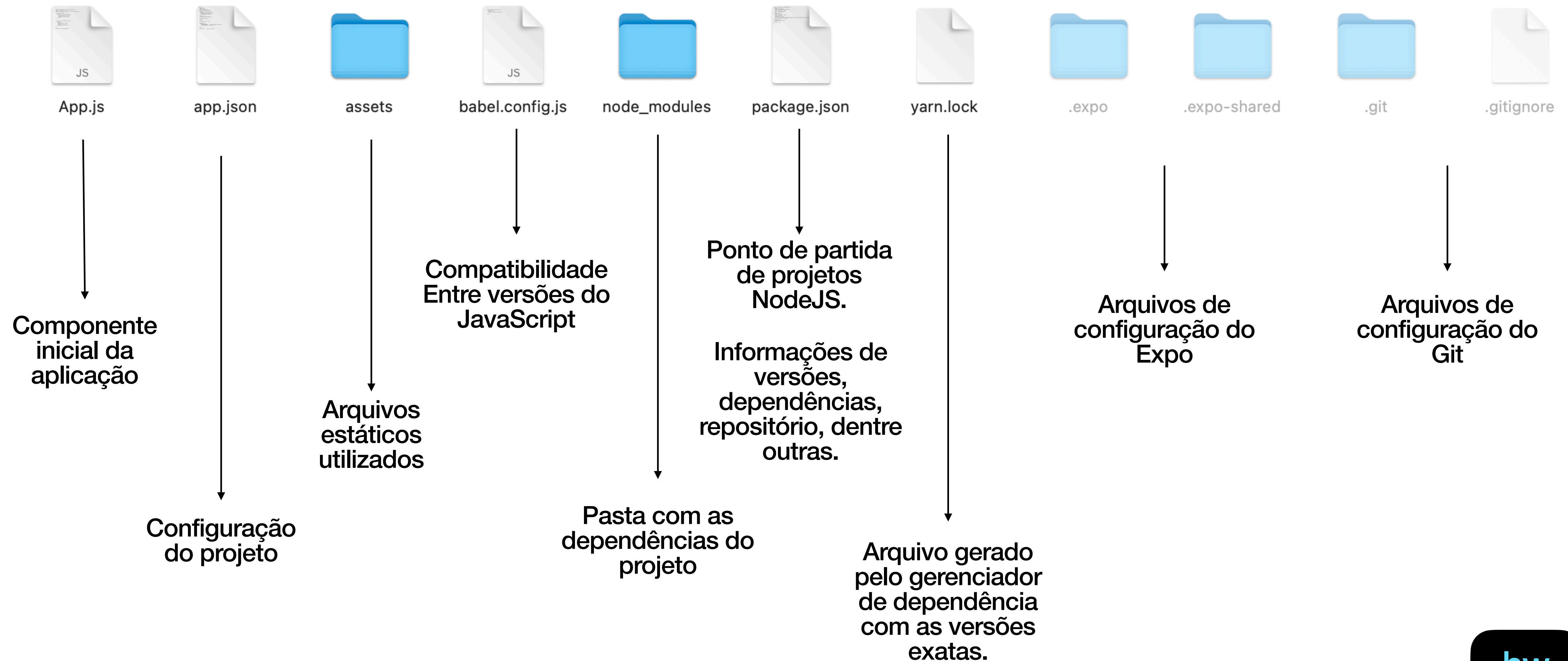
Isso iniciará um servidor de desenvolvimento para você.

Limitações do Expo

- ▶ Nem todas as APIs iOS e Android estão disponíveis
- ▶ O SDK não oferece suporte a todos os tipos de execução de código em segundo plano
- ▶ Se você precisa manter o tamanho do seu aplicativo extremamente reduzido, o fluxo de trabalho gerenciado pode não ser a melhor escolha
- ▶ Bibliotecas nativas para integração com serviços proprietários geralmente não são incluídas no SDK
- ▶ O único serviço de notificação push de terceiros com suporte é o serviço de notificação Expo
- ▶ As versões mínimas de sistema operacional suportadas são Android 5+ e iOS 10+
- ▶ As compilações gratuitas às vezes podem ser enfileiradas

Estrutura do Projeto

React Native



Função vs Classe

React Native

Hello World

Componente de Função

Componente de classe

Hello World Function Component ⓘ ↗

```
import React from 'react';
import { Text, View } from 'react-native';

const HelloWorldApp = () => {
  return (
    <View style={{
      flex: 1,
      justifyContent: 'center',
      alignItems: 'center'
    }}>
      <Text>Hello, world!</Text>
    </View>
  );
}

export default HelloWorldApp;
```

Componente de Função

Componente de classe

Hello World Function Component ⓘ ↗

```
import React, { Component } from 'react';
import { Text, View } from 'react-native';

class HelloWorldApp extends Component {
  render() {
    return (
      <View style={{
        flex: 1,
        justifyContent: "center",
        alignItems: "center"
      }}>
        <Text>Hello, world!</Text>
      </View>
    );
  }
}

export default HelloWorldApp;
```

hw

Linguagem JSX

JSX é uma extensão da sintaxe da linguagem JavaScript que fornece uma maneira de estruturar a renderização de componentes usando uma sintaxe familiar para muitos desenvolvedores. É semelhante ao HTML.

Curly Braces ⓘ ↗

```
import React from 'react';
import { Text } from 'react-native';

const Cat = () => {
  const name = "Maru";
  return (
    <Text>Hello, I am {name}!</Text>
  );
}

export default Cat;
```

Curly Braces ⓘ ↗

```
import React from 'react';
import { Text } from 'react-native';

const getFullName = (firstName, secondName, thirdName) => {
  return firstName + " " + secondName + " " + thirdName;
}

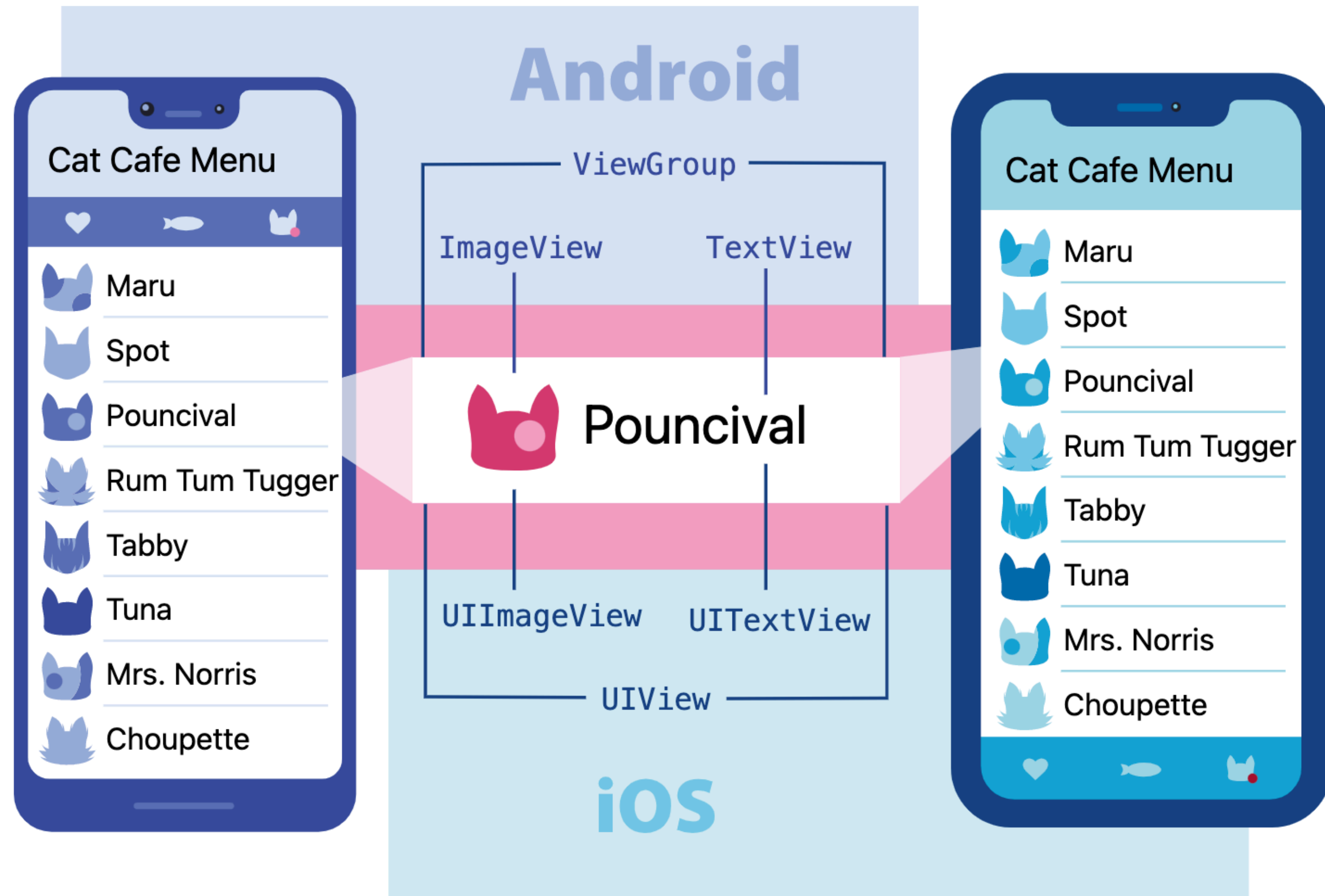
const Cat = () => {
  return (
    <Text>
      Hello, I am {getFullName("Rum", "Tum", "Tugger")}!
    </Text>
  );
}

export default Cat;
```

Componentes Principais

React Native

Componentes



React Native

COMPONENTE REACT NATIVE	ANDROID VIEW	IOS VIEW	WEB	DESCRIÇÃO
<code><View></code>	<code><ViewGroup></code>	<code><UIView></code>	Um não- scrolling <code><div></code>	Um contêiner que oferece suporte a layout com flexbox, estilo, algum manuseio de toque e controles de acessibilidade
<code><Text></code>	<code><TextView></code>	<code><UITextView></code>	<code><p></code>	Exibe, estiliza e aninha strings de texto e até lida com eventos de toque
<code><Image></code>	<code><ImageView></code>	<code><UIImageView></code>	<code></code>	Exibe diferentes tipos de imagens
<code><ScrollView></code>	<code><ScrollView></code>	<code><UIScrollView></code>	<code><div></code>	Um contêiner de rolagem genérico que pode conter vários componentes e visualizações
<code><TextInput></code>	<code><EditText></code>	<code><UITextField></code>	<code><input type="text"></code>	Permite ao usuário inserir texto

React Native


Hello World ⓘ ↗

^ Expo

```
import React from 'react';
import { View, Text, Image, ScrollView, TextInput } from 'react-native';

const App = () => {
  return (
    <ScrollView>
      <Text>Some text</Text>
      <View>
        <Text>Some more text</Text>
        <Image
          source={{
            uri: 'https://reactnative.dev/docs/assets/p_cat2.png',
          }}
          style={{ width: 200, height: 200 }}
        />
      </View>
      <TextInput
        style={{
          height: 40,
          borderColor: 'gray',
          borderWidth: 1
        }}
        defaultValue="You can type in me"
      />
    </ScrollView>
  );
};
```

Some text
Some more text



You can type in me

Preview ☒

My Device

iOS

Android

Web



multiple

Múltiplos Componentes

React Native

Multiple Components ⓘ ↗

```
import React from 'react';
import { Text, View } from 'react-native';

const Cat = () => {
  return (
    <View>
      <Text>I am also a cat!</Text>
    </View>
  );
}

const Cafe = () => {
  return (
    <View>
      <Text>Welcome!</Text>
      <Cat />
      <Cat />
      <Cat />
    </View>
  );
}

export default Cafe;
```

Definindo vários
componentes no mesmo
arquivo!

Customização para o
reuso de componentes!

props

Customização e Reuso

Utilizando PROPS para passar parâmetros para os componentes

Multiple Components ⓘ ↗

```
import React from 'react';
import { Text, View } from 'react-native';

const Cat = (props) => {
  return (
    <View>
      <Text>Hello, I am {props.name}!</Text>
    </View>
  );
}

const Cafe = () => {
  return (
    <View>
      <Cat name="Maru" />
      <Cat name="Jellylorum" />
      <Cat name="Spot" />
    </View>
  );
}

export default Cafe;
```

Guardando Dados

State ⓘ ↗

0

```
import React, { useState } from "react";
import { Button, Text, View } from "react-native";

const Cat = (props) => {
  const [isHungry, setIsHungry] = useState(true);

  return (
    <View>
      <Text>
        I am {props.name}, and I am {isHungry ? "hungry" : "full"}!
      </Text>
      <Button
        onPress={() => { setIsHungry(false); }}
        disabled={!isHungry}
        title={isHungry ? "Some milk, please!" : "Thank you!"}
      />
    </View>
  );
}

const Cafe = () => {
  return (
    <>
      <Cat name="Munkustrap" />
      <Cat name="Spot" />
    </>
  );
}
```

Utilizando STATE
para guardar
dados no
componentes