INSTITUTO FEDERAL
Ceará

Campus
Crato

Projeto de Sistemas Web 1

Frontend com React Native

Professor: Harley Macêdo de Mello

harley.mello@ifce.edu.br

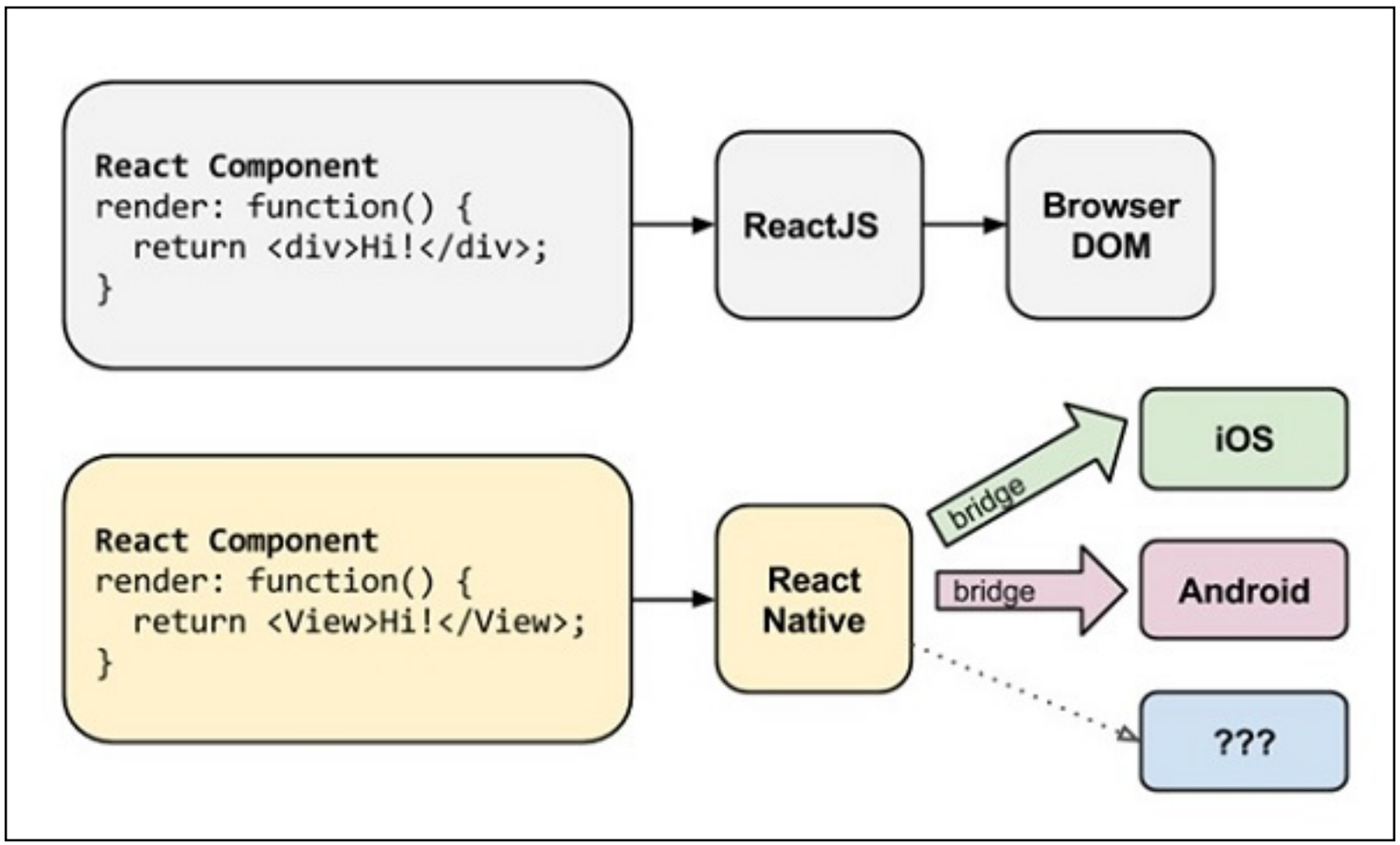
Roteiro

- Conceitos
- Ambiente de desenvolvimento
- Estrutura de App no React Native
- Componentes React Native
- Props
- State
- Events
- Estilo de componentes
- Aplicação de busca

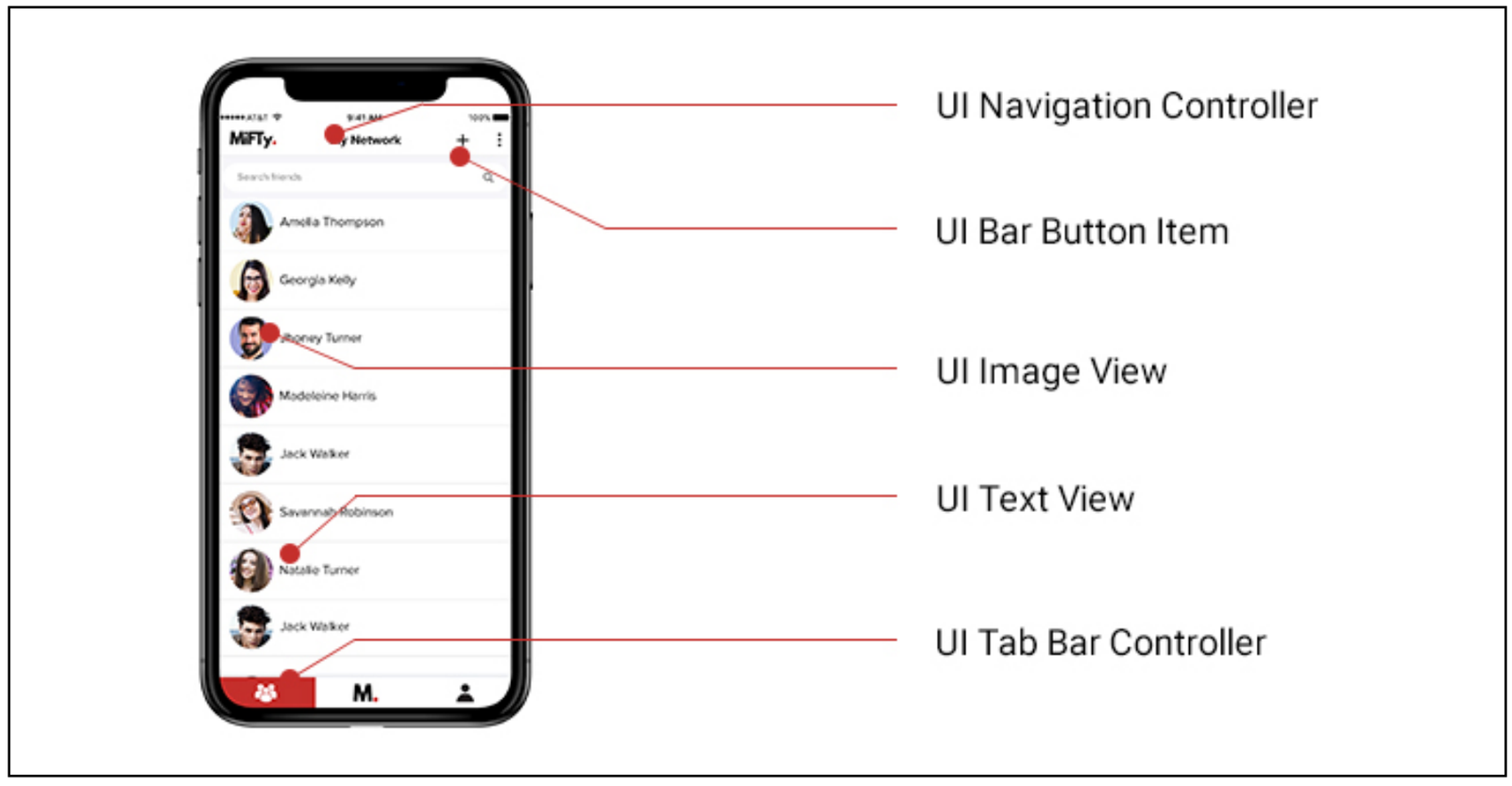
Conceitos

- Construir aplicações para plataformas mobile
- Mantido pelo Facebook
- Usa mesmos conceitos do Reactjs
- Gera uma aplicação nativa, não apenas simula
- Pouca diferença na programação para um S.O. e outro
- Pode agilizar o desenvolvimento

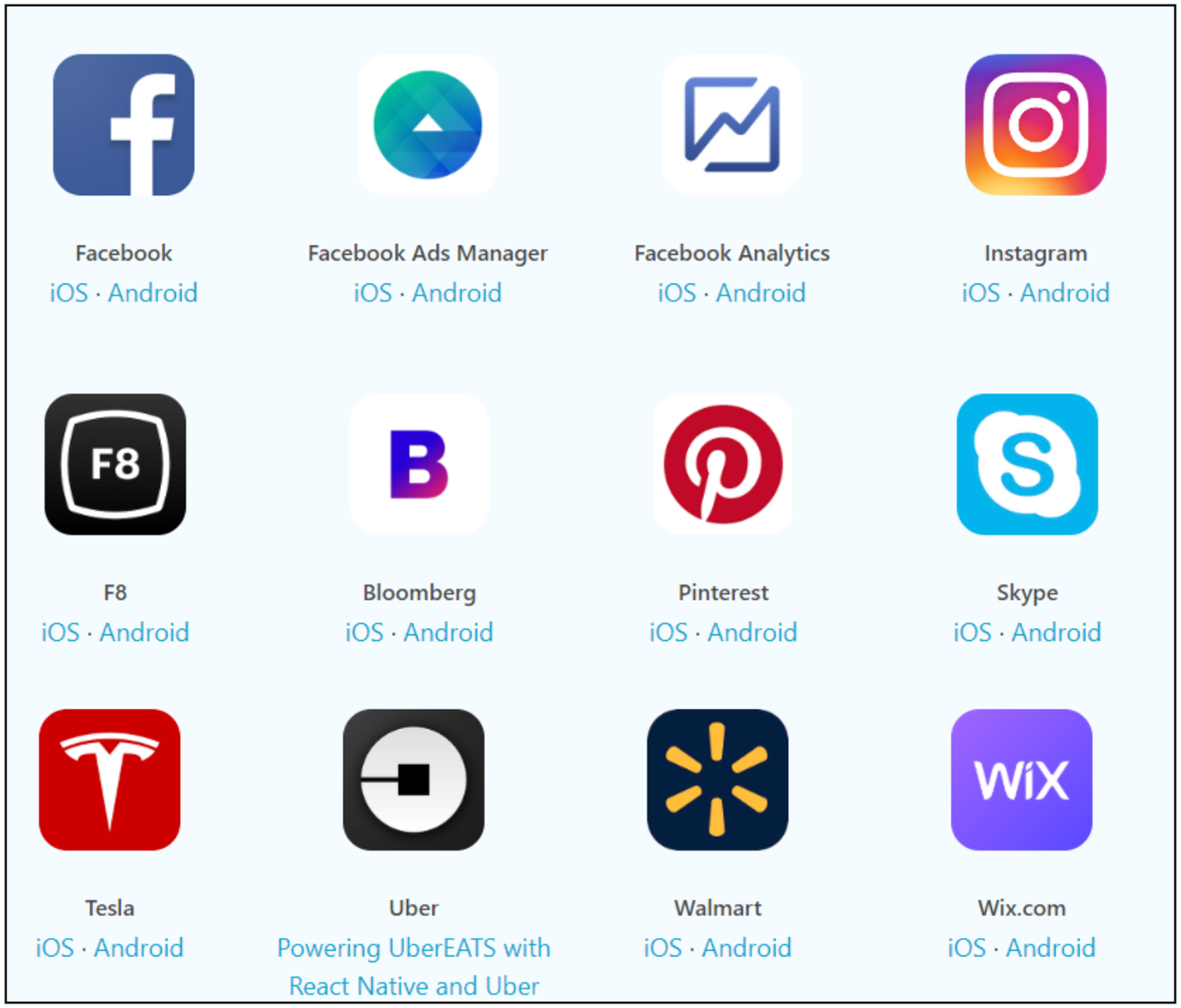
Conceitos


React Component
render: function() {
return Hi! ;
}
ReactJS
Browser DOM
React Component
render: function() {
return <View>Hi!</View>;
}
React Native
bridge
iOS
bridge
Android
???

Conceitos


MiFTy.
Search friends
Amelia Thompson
Georgia Kelly
Rooney Turner
Madeleine Harris
Jack Walker
Savannah Robinson
Natalie Turner
Jack Walker
M.
UI Navigation Controller
UI Bar Button Item
UI Image View
UI Text View
UI Tab Bar Controller

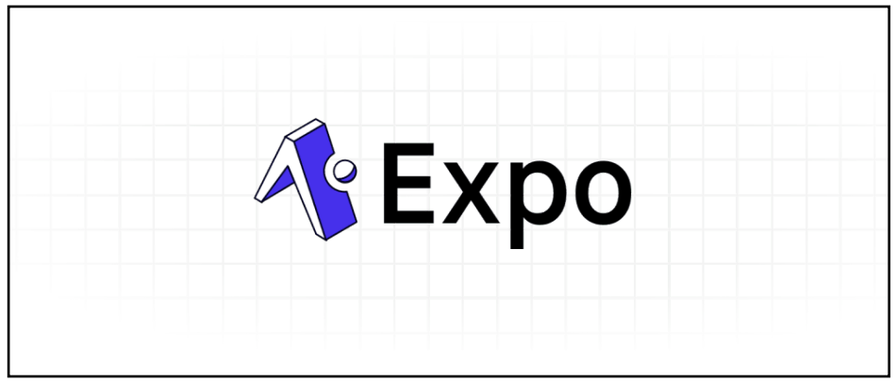
Conceitos

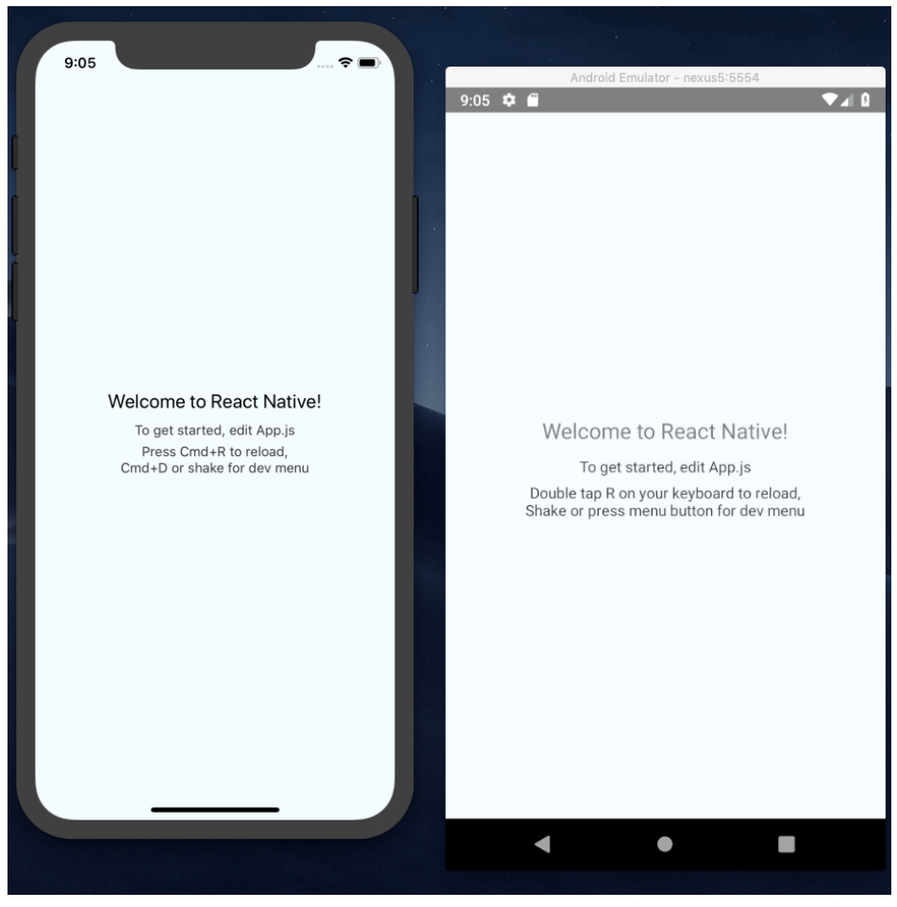

Facebook
iOS · Android
Facebook Ads Manager
iOS · Android
Facebook Analytics
iOS · Android
Instagram
iOS · Android
F8
F8
iOS · Android
Bloomberg
iOS · Android
Pinterest
iOS · Android
Skype
iOS · Android
Tesla
iOS · Android
Uber
Powering UberEATS with React Native and Uber
Walmart
iOS · Android
Wix.com
iOS · Android

Ambiente de desenvolvimento

- Ambiente Android
  - Node e NPM
  - JDK 8 e Android Studio
  - Python 2
  - Simulador de dispositivo Android
- Ambiente Expo
  - Node e NPM
  - Pacotes expo-cli
  - Smartphone ou simulador
  - Instalar o app 'expo' na loja de seu sistema móvel

Ambiente de desenvolvimento

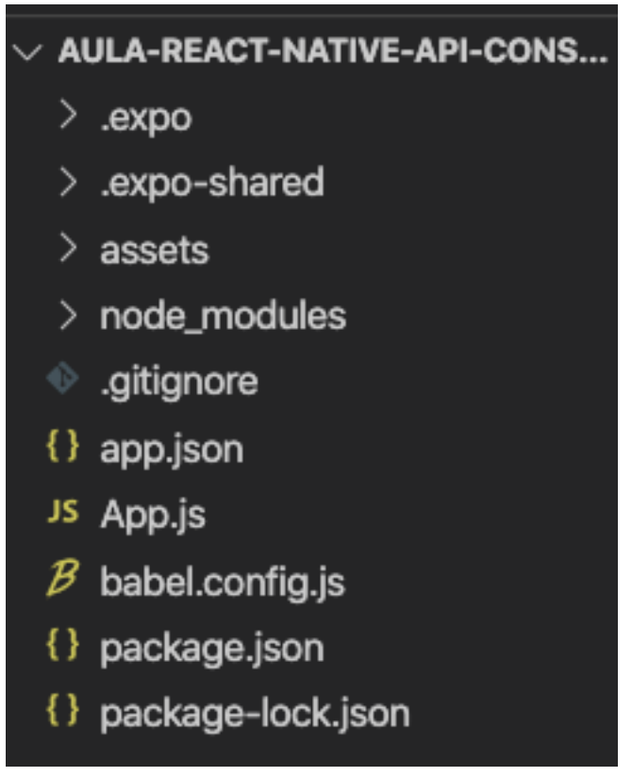




Estrutura de App no React Native

- Instalar o pacote expo-cli
  - npm install -g expo-cli
- Executar o criador de App
  - expo init meu-app —npm
- Entrar na pasta criada
  - cd meu-app
- Iniciar app criada
  - npm install
  - expo start

Estrutura de App no React Native

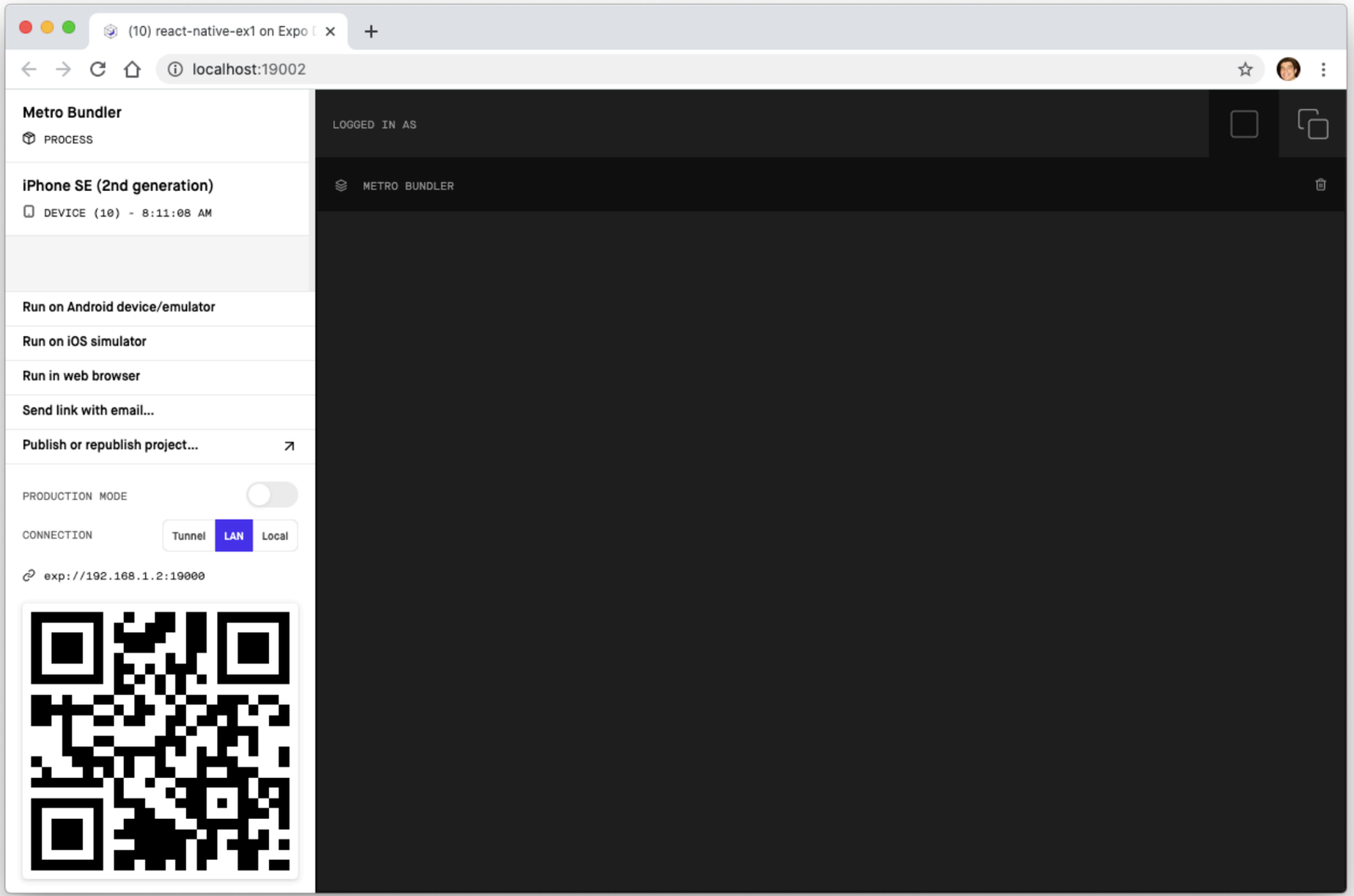


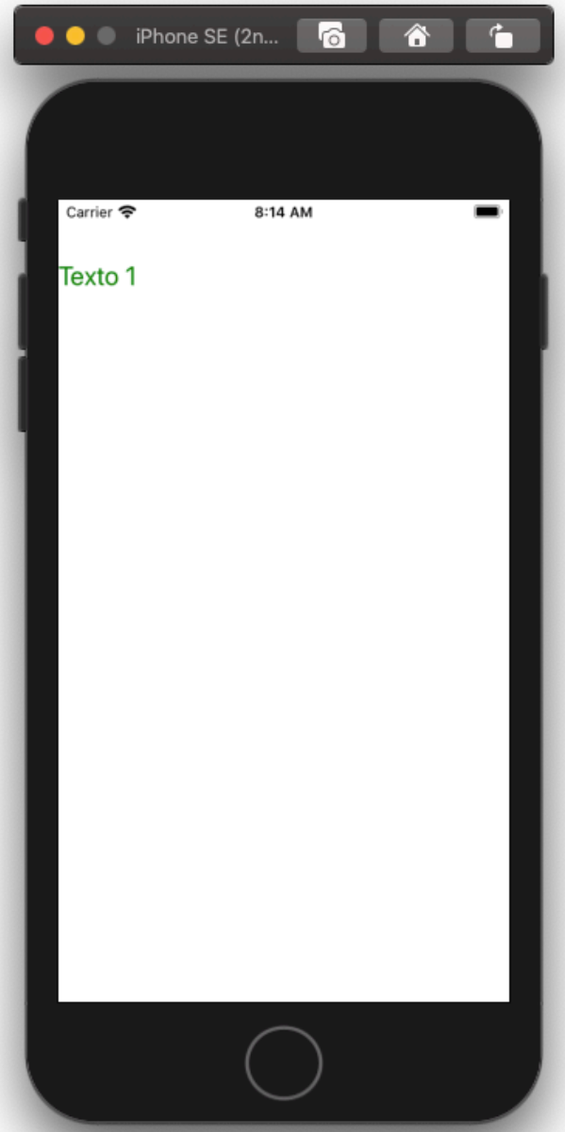
App.js

react-native-ex1 > App.js > ...

```
1   import React, {Component} from 'react'
2   import {StyleSheet, View, Text} from 'react-native'
3
4   class App extends Component {
5     render() {
6       return (
7         <View>
8           <Text style={styles.texto} >Texto 1</Text>
9         </View>
10      )
11    }
12  }
13
14  const styles = StyleSheet.create({
15    texto: {
16      fontSize: 22,
17      color: 'green',
18      marginTop: 50
19    }
20  })
21
22  export default App
```

Estrutura de App no React Native

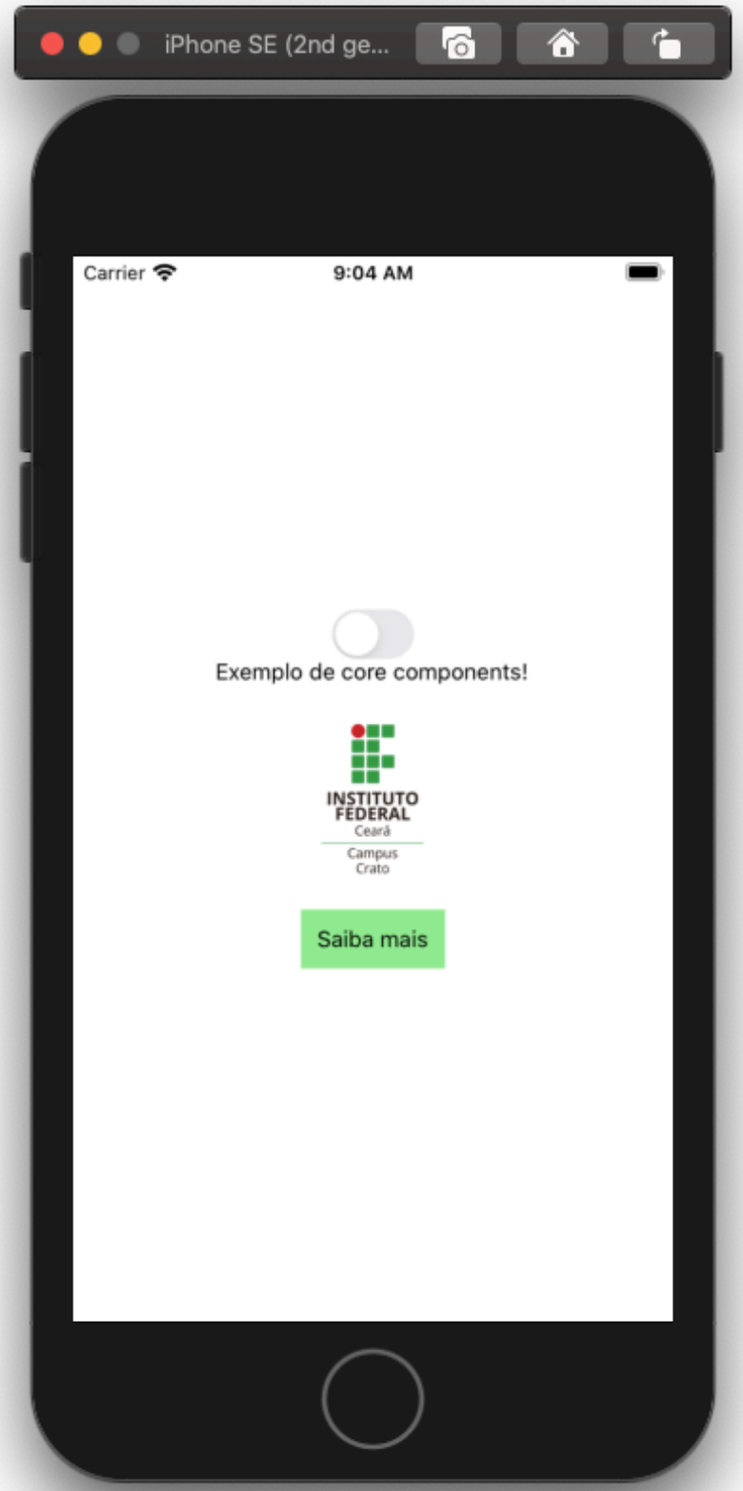




Componentes React Native

- Semelhantes aos componentes do Reactjs
- Precisam retornar um conteúdo centralizado
- Podem ter estado e propriedades
- Componentes 'core' como o View e o ListView
- Representam componentes nativos de cada S.O.
- Componentes próprios podem agregar nativos

Componentes React Native



Componentes React Native

- View
  - Container para o conteúdo da tela
  - <View> … </View>
- SafeAreaView
  - Considera apenas área livre
  - <SafeAreaView> … </SafeAreaView

Componentes React Native

- Text
  - Texto genérico
  - Reconhece click
  - <Text> … </Text>
- TextInput
  - Entrada de texto
  - Evento onChangeText
  - <TextInput> … </TextInput>

Componentes React Native

- Button
  - Botão padrão
  - Evento onPress
  - <Button> … </Button>
- TouchableOpacity
  - Container para textos que serão clicados
  - Aparência de botão
  - <TouchableOpacity> … </TouchableOpacity>

Componentes React Native

- Image
  - Imagem padrão
  - Atributo source
  - <Image />
- Switch
  - Alternância ou liga/desliga
  - Evento onValueChange
  - <Switch />

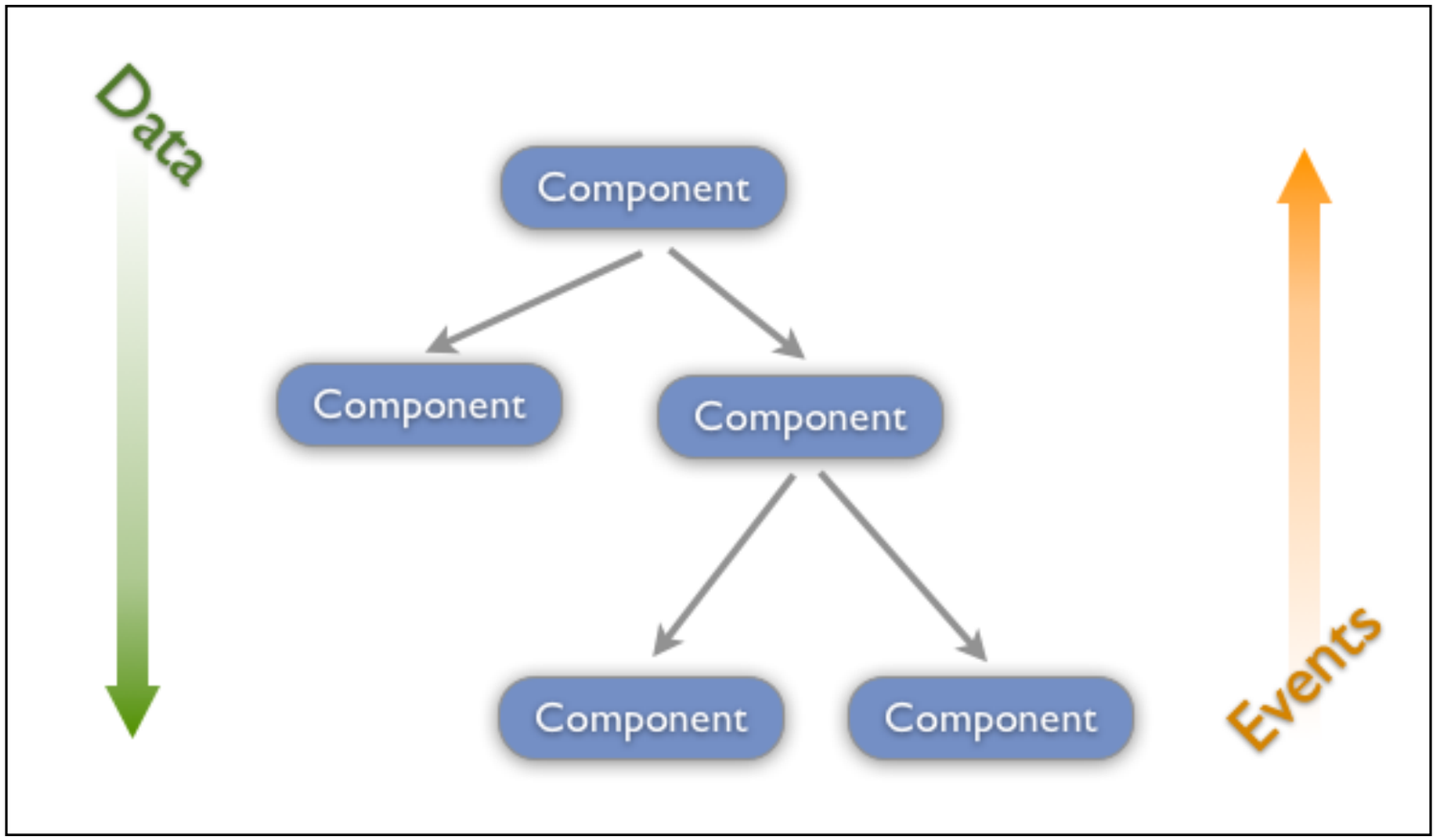
Componentes React Native

- Componentes próprios retornam algum JSX
- Podem agregar outros componentes próprios
- Podem agregar componentes 'core'
- Semelhantes aos componentes próprios no Reactjs
  - <Busca> … </Busca>
  - <Cartao> … </Cartao>

Props

- Forma usada para passar parâmetros para um componente
- Passados por meio de atributos HTML
- As props podem ser integradas ao estado do componente
- As props são apenas leitura

Props


Data
Component
Component
Component
Component
Component
Events

Props

App.js

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

import Componente1 from './Componente1';


export default class App extends Component {

  render(){
    return (
      <View>
        <Componente1 myState="Testando props" />
      </View>
    );
  }
```

Componente1.js

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

export default class Componente1 extends Component {

  render(){
    return (
      <View>
        <Text>{this.props.myState}</Text>
      </View>
    );
  }
```

State

- Armazenar valores que pertecem ao componente
- Quando o State muda, o componente é renderizado novamente
- Inicializado no construtor
- Usar a função this.setState({ }) para alterar o state

State

```
24    render(){
25      return (
26        <View>
27          <Button onPress={this.mudarConteudo} title="mudar conteudo" >Mudar conteúdo</Button>
28          <Button onPress={this.mudarCor} title="mudar cor" >Mudar cor</Button>
29          <View style={{backgroundColor: this.state.cor}}>
30            <Text>{this.state.mensagem}</Text>
31          </View>
32        </View>
33      )
34    }
35  }
36
37  export default App
```

State

JS App.js

JS App.js > ...

```
1   import React, { Component } from 'react'
2   import { Button, Text, View } from 'react-native'
3
4   class App extends Component {
5
6     constructor(props){
7       super(props)
8       this.state = {
9         mensagem: "Lorem ipsum dolor sit amet",
10        cor: "green"
11      }
12      this.mudarConteudo = this.mudarConteudo.bind(this)
13      this.mudarCor = this.mudarCor.bind(this)
14    }
15
16    mudarConteudo() {
17      this.setState({mensagem: "Consectetur adipisicing elit"})
18    }
19
20    mudarCor() {
21      this.setState({cor: 'orange'})
22    }
```

Events

- Programar ações de resposta
- Usar o camelCase para nomes de eventos
- É preciso vincular os métodos para a instância atual usando o bind
- Eventos com nomes diferentes do HTML
  - onPress
  - onChangeText
- O objeto que disparou o evento é passado como parâmetro
  - event.nativeEvent.text

Estilo de componentes

- Aplicar o mesmo formato de CSS
- Usando camelCase para as propriedades
- Usar {{ algumAtributo: valor }}
- CSS no mesmo arquivo pode ser definido com constante

Aplicação de busca

App.js | Busca.js | Cartao.js

App.js > ...

```
1   import React, {Component} from 'react'
2   import { StyleSheet, View} from 'react-native'
3   import Busca from './components/Busca'
4
5   export default class App extends Component {
6
7     render() {
8       return (
9         <View style={styles.container}>
10          <Busca></Busca>
11        </View>
12      )
13    }
14  }
```

Aplicação de busca

JS App.js | JS Busca.js | JS Cartao.js

components > JS Busca.js > styles > botao1 > borderRadius

```
1   import React, {Component} from 'react'
2   import {View, Text, TextInput, StyleSheet, TouchableOpacity} from 'react-native'
3   import Cartao from './Cartao'
4
5   class Busca extends Component {
6
7       constructor(props) {
8           super(props)
9           this.state = {
10              textoEntrada: '',
11              dados: [],
12          }
13          this.buscar = this.buscar.bind(this)
14          this.atualizarTexto = this.atualizarTexto.bind(this)
15      }
16
17      atualizarTexto(event) {
18          this.setState({textoEntrada: event.nativeEvent.text})
19      }
```

Aplicação de busca

```
21      buscar() {
22        let url = 'https://aula-node-api-server.herokuapp.com/api/' + this.state.textoEntrada
23        console.log(url)
24        fetch(url)
25        .then( (resultado) => {
26          return resultado.json()
27        })
28        .then( (resultado) => {
29          this.setState({dados: resultado})
30        })
31        .catch( (error) => {
32          this.setState({textoResultado: 'Valor não encontrado'})
33        })
34      }
35
36      render() {
37          return (
38              <View style={styles.container}>
39                  <Text style={styles.titulo1} >Busca rápida de Professores</Text>
40                  <Text style={styles.titulo1} >Informe o termo:</Text>
41                  <TextInput placeholder='Termo' style={styles.entrada1} onChange={this.atualizarTexto} />
42                  <TouchableOpacity style={styles.botao1} onPress={this.buscar} >
43                      <Text style={styles.textoBotao1} >Buscar</Text>
44                  </TouchableOpacity>
45                  <View>
46                      {
47                          this.state.dados.map( (item, i) => {
48                              return ( <Cartao nome={item.nome} area={item.area} key={i} /> )
49                          })
50                      }
51                  </View>
52
53              </View>
54          )
55      }
```

Aplicação de busca