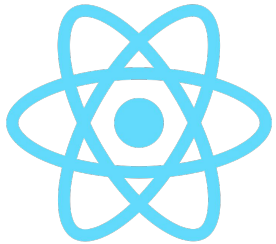
React Native Training

Day 1



React Native

O importante saber...

- React Native é uma biblioteca que utiliza o JavaScript para interagir com componentes nativos disponibilizados pela Apple e pela Google para o iOS e Android

- O React Native abstraí os componentes nativos que cada sistema nativo reconhece

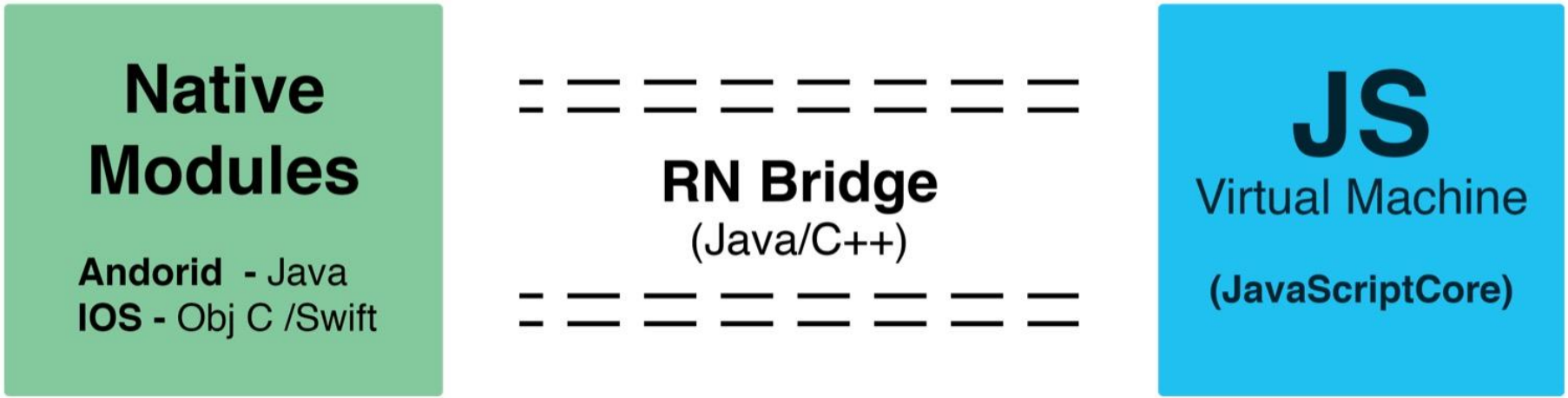
Ao usar o View do React Native, por exemplo...

O iOS está renderizando:

UIView

O Android está renderizando...

android.view



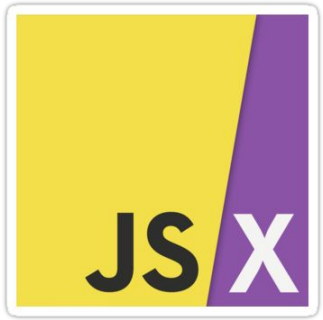
Isso não acontece magicamente

Prós

- A utilização de componentes nativos “por baixo do pano” faz com que um app feito com o React Native pareça realmente nativo
- Utilizar componentes já disponibilizados pelo sistema nativo (iOS ou Android) diminui o tamanho do app, que não os cria do zero

Contras

- A comunicação entre o lado JavaScript e o lado Nativo não é simples nem ultra rápida. Ganhamos em tempo de desenvolvimento, mas perdemos um pouco em termos de desempenho
- O JavaScript em si não é uma linguagem muito segura ou estável


JSX

Escrevendo em React

- JSX
  - Parece a sintaxe do XML só que para JavaScript
  - É transpilado para JavaScript
  - O JSX é uma extensão de sintaxe para a linguagem JavaScript.
  - Tags em minúsculas (lowercase) são tratadas como tags HTML, tags em maiúsculas (uppercase) são tratadas como componentes customizados

- Componentes são apenas funções
  - Que retornam algo (um node) que o react pode renderizar (exemplo: um , <MyButton/>)

Essencialmente, JSX é açúcar sintático

O que você escreve como...

```
<MyButton color="blue"
shadowSize={2}>

  Click Me

</MyButton>
```

Transpila para...

```
React.createElement(
    MyButton,
    {color: 'blue', shadowSize: 2},
    'Click Me'
)
```

Ao USAR um componente, você está gerando/retornando/renderizando algo na tela.

É uma maneira de declarar de forma simples...

- Nome
- Propriedades
  - [EXEMPLO] Um componente Button pode ter cor, tamanho, texto, ícone...
  - Inclusive, você pode declarar valores padrões para seus atributos
  - Geralmente guarda informações estáticas ou passadas por outros componentes
- Estado
  - [EXEMPLO] Um componente Relógio tem um estado que muda a cada segundo
  - Geralmente guarda informações internas ao componente
- Filhos
  - Uma BarraLateral pode receber vários filhos MenuItems
  - Uma View pode ter vários Text

Como fazer com que um componente funcione e pareça exatamente como você quer?

- Usando os componentes básicos fornecidos pelo React Native
- Customizando esses componentes básicos
- Montando componentes básicos e customizados uns dentro dos outros

Components

- Componentes **funcionais** não possuem estado e nem acesso aos métodos do ciclo de vida do componente.
- Componentes de **classe** possuem estado. Com o estado você consegue guardar dados que ao serem atualizados modificam a apresentação do componente.
- Componentes de **classe** possuem métodos de ciclo de vida e de estado

Componentes

- Componentes funcionais

```
const Hi = (props) => {

    return <Text>Olá, {props.nome}
</Text>;

};

const elemento = <Hi nome="Howl" />;
```

- Componentes de classe

```
class Hi extends Component {

    render() {

        return <Text>Oi
    {this.props.nome}!</Text>;
    }

}

const elemento = <Hi nome="Howl" />;
```

Especificando atributos

- Você pode usar aspas duplas para especificar literais de string como atributos:
  - **const** botao = <**Button** title="Saiba mais" />;

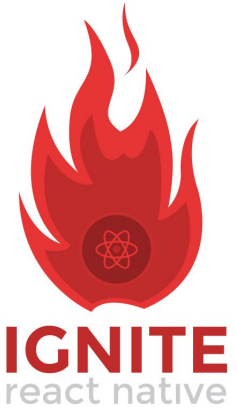
- E chaves para incorporar uma expressão em um atributo:
  - **const** valor = 'Alguma entrada';
  - **const** entrada = <**TextInput** value={valor} />;

Props e State

- PROPS
  - São passadas de um componente para o outro
  - São imutável, ou seja, somente leitura

- STATE
  - Componentes baseado em classes tem state
  - State são mutáveis e são mudados através do método setState();

Iniciar um projeto em branco



ignite new YourAppName

Which boilerplate would you like to use? Andross

Would you like Ignite Development Screens? No

What vector icon library will you use? react-native-vector-icons

What internationalization library will you use? None

What animation library will you use? None

Would you like to include redux-persist? Yes

Crie seu repositório

```
cd YourAppName
git init
git branch <new-branch>
git checkout <new-branch>
git add .
git commit -m “Initial commit”
```

Faça o upload para o GitHub

Entre na sua conta do GitHub

Crie um repositório totalmente vazio (sem geração de README, gitignore, ou licença)

Execute os seguintes comandos na pasta do seu projeto:

```
git remote add origin https://github.com/username/project.git
```

```
git push -u origin master
```

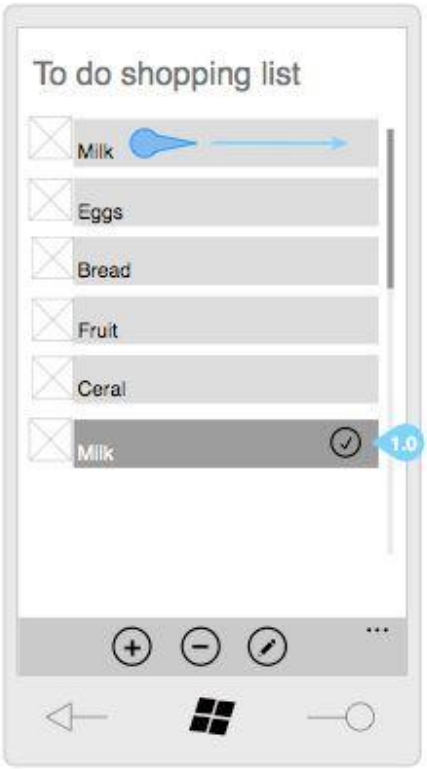
Componente

Componentes

Os componentes permitem que você divida a UI em peças independentes, reutilizáveis e pense em cada peça isoladamente.

Exemplo:
Quebrando o feed do Facebook em componentes

Quantos componentes você consegue ver?



Links importantes

React Native Layout Cheat Sheet

React Native Docs: Layout with Flexbox

React Native Guide: Functional vc Presentation Components

Desafio

Divida a tela em 3 colunas iguais

Desafio

Divida a tela em 3 partes:

Uma parte ocupará ⅔ da tela

O último terça será dividido igualmente

Desafio

Divida a parte superior (a de 2/3) em **colunas** de tamanhos iguais