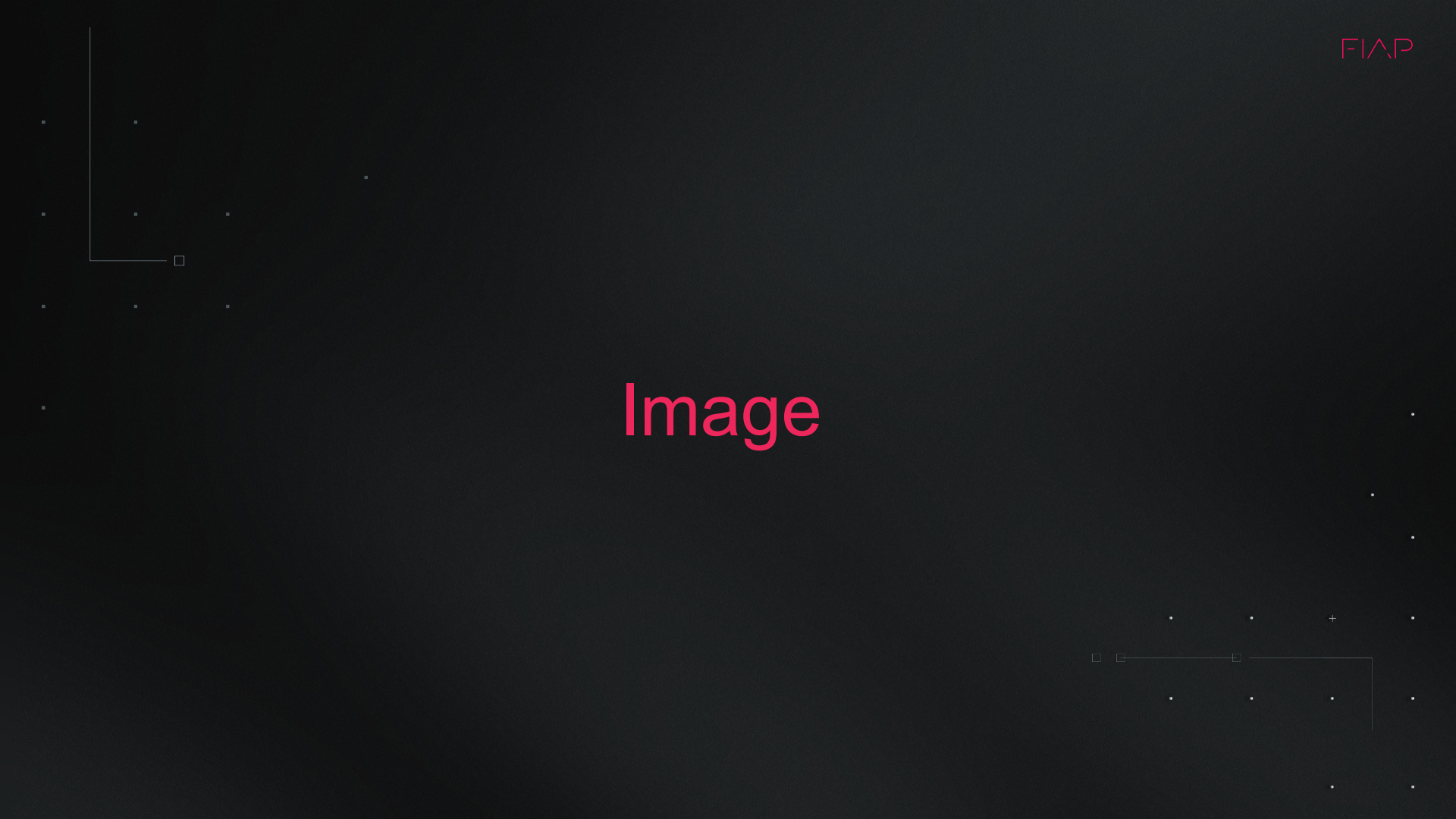
Mobile App Development

Dynamic Render, Image, ImageBackground, Components



Image

Image

Como usar image no react-native?

É a instanciação de uma imagem como um componente. A fonte da imagem pode ser uma url ou um arquivo que esteja no seu projeto

Image

Principais Props

- source
- style
- alt (acessibilidade)

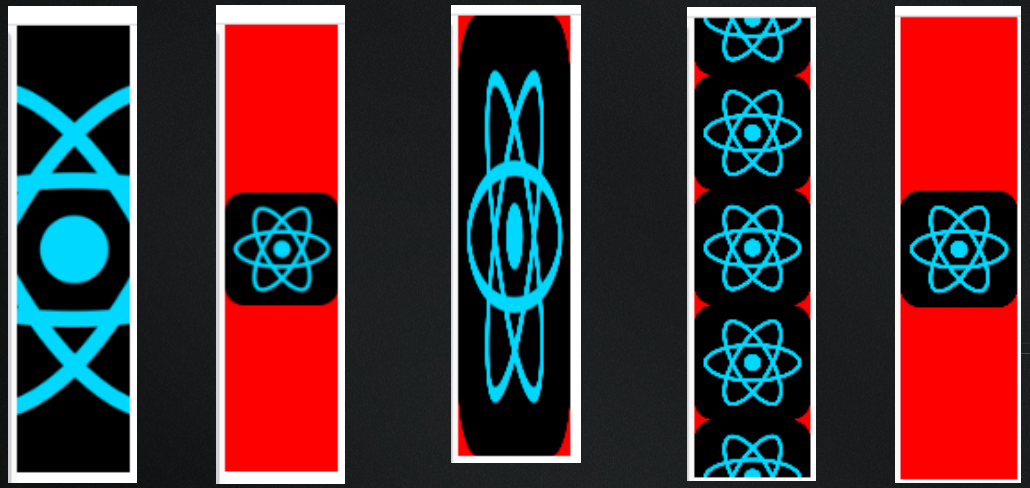
Image

resizeMode:

- cover: Uma dimensão do tamanho correto, outra maior. Mantém o ratio
- contain: Uma dimensão do tamanho correto, outra menor. Mantém o ratio
- stretch: Redimensiona a imagem para o tamanho declarado, pode mudar o ratio
- opacity

Image

resizeMode:
cover, contain, stretch, repeat, center



Image

Exemplos de source:

```
<Image
    style={styles.tinyLogo}
    source={require('@expo/snack-static/react-native-logo.png')} />
<Image
    style={styles.tinyLogo}
    source={{ uri: 'https://reactnative.dev/img/tiny_logo.png' }} />
```

ImageBackground

ImageBackground

É um componente que funciona como um container, igual à View. Mas evita o uso do backgroundImage style para ter uma imagem no fundo.
Possui props e style tanto de Image e quanto de View

ImageBackground

Principais Props

- As props do Image
- style
- imageStyle
- imageRef

Relembrando a dica

Como saber as props/styles de um componente react-native?

Todos os componentes básicos do react-native e das principais bibliotecas são bem documentados. Basta procurar no site onde está a documentação e podemos achar essas informações além de exemplos de uso

Relembrando a dica

ImageBackground

A common feature request from developers familiar with the web is `background-image`. To handle this use case, you can use the `<ImageBackground>` component, which has the same props as `<Image>`, and add whatever children to it you would like to layer on top of it.

You might not want to use `<ImageBackground>` in some cases, since the implementation is basic. Refer to `<ImageBackground>`'s source code for more insight, and create your own custom component when needed.

Note that you must specify some width and height style attributes.

Example

ImageBackground

Expo

```
<ImageBackground source={image} resizeMode="cover" style=
```

Componentização

Por que criar componentes personalizados em react?

1 - Evita duplicidade de código
2 - Organização e legibilidade
3 - Permite testes unitários mais simples e eficientes
4 - Facilita o fluxo de dados

```
class Classe extends React.PureComponent {
  state = {
    titleText: "Ninguém apertou o botão",
  };

  onPressButton = (name) => {
    this.setState({titleText: `${name} apertou o botão`});
  };

  render() {
    return (
      <View>
        <Text>
          {this.state.titleText}
        </Text>
        <Button
          title="Stop capturing"
          onPress={() => onPressButton(this.props.name)}
          color="#FF0000"
        />
      </View>
    );
  }
}
```

```jsx
const functionalComponent = (props) => {
    const [titleText, setTitleText] = React.useState("Ninguém apertou o botão");

    const onPressButton = (name) => {
        setTitleText(`${name} apertou o botão`);
    };

    return (
      <View>
        <Text>
          {titleText}
        </Text>
        <Button
          title="Stop capturing"
          onPress={() => onPressButton(props.name)}
          color="#FF0000"
        />
      </View>
    )
}
```

Componentização

E como sabemos se devemos criar um componente específico?

1 - Reutilização: Se aquele pedaço de código é utilizado em muitos lugares, provavelmente deveria ser um componente
2 - Responsabilidade única: Cada componente deve, idealmente ter uma responsabilidade
3 - Tamanho: Se o seu componente pai está muito grande, talvez deva ser dividido em componentes menores

Instanciação Dinâmica

Como criar componentes de maneira dinâmica?

É muito comum termos um desafio em que não temos um número específico de componentes. Principalmente quando lidamos com dados que não sabemos no início da aplicação, ou quando temos uma configuração que queremos poder mudar facilmente

Instanciação Dinâmica

Como criar componentes de maneira dinâmica?

Ex: Lista de favoritos, gerando um card para cada produto favorito

Ex2: Um jogo de xadrez, que cada espaço é um botão (não queremos declarar 64x uma casa)

Instanciação Dinâmica

Solução: O React aceita um array de componentes no JSX

Podemos gerar um array que será mapeado para componentes, ou então criar um array vazio que é populado com componentes conforme o necessário

Relembrando Arrays

Funções importantes:

JavaScript Demo: Array.forEach()

```
1 const array1 = ['a', 'b', 'c'];
2
3 array1.forEach(element => console.log(element));
4
5 // Expected output: "a"
6 // Expected output: "b"
7 // Expected output: "c"
8
```

Relembrando Arrays

Funções importantes:

JavaScript Demo: Array.map()

```
1 const array1 = [1, 4, 9, 16];
2
3 // Pass a function to map
4 const map1 = array1.map(x => x * 2);
5
6 console.log(map1);
7 // Expected output: Array [2, 8, 18, 32]
8
```

Relembrando Arrays

Funções importantes:

JavaScript Demo: Array.filter()

```
1 const words = ['spray', 'limit', 'elite', 'exuberant', 'destruction', 'present'];
2
3 const result = words.filter(word => word.length > 6);
4
5 console.log(result);
6 // Expected output: Array ["exuberant", "destruction", "present"]
7
```

Instanciação Dinâmica

Queremos transformar um array de valores, em um array de componentes

```
<View>
    {
        [
            <Button title={1} onPress={() => pressMe(1)} />,
            <Button title={2} onPress={() => pressMe(2)} />,
            <Button title={3} onPress={() => pressMe(3)} />,
            <Button title={4} onPress={() => pressMe(4)} />,
        ]
    }
</View>
```

Instanciação Dinâmica

Qual método de Array usaremos?

```
<View>
    {
        [1, 2, 3, 4].map(value =>
            <Button title={value} onPress={() => pressMe(value)}/>
        )
    }
</View>
```

React Hooks

React hooks

O que é um hook?

Os hooks são métodos especiais utilizados pelo componentes React do tipo functional component.

Principais:

- useEffect
- useState

React hooks

useState

É o hook que utilizamos para adicionar um estado interno de um componente. Ou seja, qualquer informação que precisamos salvar **dentro** do componente utilizamos esse hook.

Exemplos de uso:

- Inputs dos usuários
- Controlar estado de componentes filhos
- Mudanças internas do componente

React hooks

```
const [variable, setVariable] = React.useState('valor inicial opcional')
```

```
const changeVariable = (newValue) => {
    console.log('Minha variável era: ', variable);
    console.log('Vou mudar para: ', newValue);
    setVariable(newValue);
    console.log('Mudou para: ', variable)
}
```

React hooks

useEffect

É o hook que utilizamos para reagir a mudanças internas ou externas do componente.
Podemos pensar nesse hook como um callback, ou um efeito colateral adicionado quando há uma mudança no componente.

React hooks

useEffect

É o hook que utilizamos para reagir a mudanças internas ou externas do componente.
Podemos pensar nesse hook como um callback, ou um efeito colateral adicionado quando há uma mudança no componente.

React hooks

useEffect

```
React.useEffect(() => {
    doThis();
    doThat();
    return () => {
        doThisCleanup();
    }
},[var1, var2])
```

setup

cleanup

dependencies

React hooks

useEffect

Setup: Esse código é executado quando o componente é **montado**, e em cada **nova renderização** do componente

React hooks

useEffect

Cleanup: Esse código é executado **antes** de cada nova renderização, e antes do componente ser desmontado

React hooks

useEffect

Dependencies: Determina quais condições para esse hook useEffect ser executado. Aceita nulo, array vazio e array com variáveis

Nulo: Executa toda renderização
Array vazio: Executa ao montar e desmontar
Array populado: Executa sempre que **alguma** das variáveis do array for modificada

React hooks

useEffect

Cleanup: Esse código é executado **antes** de cada nova renderização, e antes do componente ser desmontado

Links

https://beta.reactjs.org/reference/react/useEffect

https://reactnative.dev/docs/components-and-apis

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array

Dúvidas, anseios, desabafos?


FIAP