



Desenvolvimento para Dispositivos Móveis

Aula 08 - Estilo (Parte 1)



Material Didático do Instituto Metrôpole Digital - IMD

Termo de uso

Os materiais didáticos aqui disponibilizados estão licenciados através de Creative Commons **Atribuição-SemDerivações-SemDerivados CC BY-NC-ND**. Você possui a permissão para realizar o download e compartilhar, desde que atribua os créditos do autor. Não poderá alterá-los e nem utiliza-los para fins comerciais.

Atribuição-SemDerivações-SemDerivados

CC BY-NC-ND



<https://creativecommons.org/licenses/by-nc-nd/4.0/>

Apresentação

Nesta aula começaremos a estudar sobre estilização dos componentes *React Native*.

Objetivos

- Conhecer como criar objetos de estilização
- Conhecer como trabalhar com *StyleSheet*
- Conhecer como trabalhar com dimensões
- Conhecer como utilizar o *flex*

StyleSheet

Link do video da aula: <https://youtu.be/WkTQq4ZYTpl>

A estilização é importante para que o aplicativo tenha uma boa aparência e chame a atenção do usuário.

Folha de Estilos

Em outros momentos, vimos a estilização utilizando a *tag* `'style'` que recebe um objeto similar a um `'css'`. Continuaremos usando a *tag*, mas agora criaremos um objeto separado que contém os estilos e vamos passar esse objeto para a *tag* `style`.

Na criação do objeto, será dado um nome e criadas propriedades que terão a estilização desejada. Após isso, essa propriedade será usada na parte do código à qual é desejável aplicar tal estilização.

Criando a *StyleSheet*

Para praticar a criação e o uso do *StyleSheet*, acompanhe a aula e implemente o código abaixo:

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

export default function App() {
  return (
    <SafeAreaView style={styles.container}>
      <Text style={[styles.red, styles.grande]}>Hello From
      React</Text>
    </SafeAreaView>
  );
}
```

```
);  
}  
  
const styles = StyleSheet.create({  
  
  container: {  
    backgroundColor: '#0F0'  
  },  
  red: {  
    color: '#F00'  
  },  
  grande: {  
    fontSize: 30  
  }  
  
})
```

Inserindo estilo '*inline*'

Como visto em outras aulas, há como inserir estilo 'inline'. Veja o exemplo abaixo:

```
<Text style={{ fontSize: 50 }}>Inline Style</Text>
```

Esse método é mais adequado para estilizações pontuais.

Note: É importante lembrar de fazer a importação do 'StyleSheet'.

Dimensões

Link do video da aula: <https://youtu.be/FXd6l3EYlrE>

Dando continuidade aos estudos de estilização no *React Native*, focando agora na parte de dimensões dos componentes.

Dimensão absoluta

No código, é possível alterar as dimensões de maneira absoluta. Esse modo consiste em utilizar um valor numérico nos parâmetros de altura e largura, fazendo com que esses valores sejam fixos independentemente da resolução da tela que será usada.

Um exemplo de criação de um objeto de estilização com dimensão absoluta pode ser visto abaixo:

```
const styles = StyleSheet.create({

  pequeno: {
    width: 100,
    height: 100
  },
  medio: {
    width: 200,
    height: 200
  },
  grande: {
    width: 300,
    height: 300
  }
})
```

Dimensão percentual

Na dimensão percentual, o aplicativo irá calcular a dimensão passada nos parâmetros *width* e *height*, em forma de porcentagem, conforme a resolução da tela em que está sendo executado.

Um exemplo disso pode ser visto abaixo:

```
const styles = StyleSheet.create({

  pequeno: {
    width: '20%',
    height: '10%'
  },
  medio: {
    width: '40%',
    height: '25%'
  },
  grande: {
    width: '80%',
    height: '65%'
  }
})
```

Essa porcentagem é proporcional à *View* na qual está inserida, não a toda tela.

Prática

Para praticar isso, utilizaremos o código anterior, substituindo os componentes de `Text` por `View` 's. Assim, será possível inserir uma cor de fundo, redimensioná-las e visualizar os resultados.

Acompanhando a aula, implemente o código abaixo:

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

export default function App() {
  return (
    <SafeAreaView style={styles.container}>
      <View style={[styles.blue, styles.pequeno]}></View>
      <View style={[styles.red, styles.medio]}></View>
      <View style={[styles.blue, styles.grande]}></View>
    </SafeAreaView>
  );
}

const styles = StyleSheet.create({
  container: {
    backgroundColor: '#0F0'
  },
  red: {
    backgroundColor: '#F00'
  },
  pequeno: {
    width: 100,
    height: 100
  },
  medio: {
    width: 200,
    height: 200
  },
  grande: {
    width: '50%',
    height: '20%'
  },
  blue: {
    backgroundColor: 'powderblue'
  }
});
```

})

Flex

Link do video da aula: <https://youtu.be/l5vqKebVYrg>

Nesta aula iremos continuar a parte de dimensões, mas agora introduzindo o conceito de *Flex*.

Conhecendo o *Flex*

O *flex* é utilizado para permitir a melhor adaptação do aplicativo a cada dispositivo, ficando consistente em todas as resoluções.

Ainda que utilizar as dimensões de forma proporcional funcione, o *flex* traz muitas possibilidades a mais.

Utilizando o *Flex*

Para utilizar o *flex* nos componentes internos, o componente principal precisa ter sua dimensão bem definida. Isso pode ser feito utilizando o modo absoluto, percentual ou o *flex*.

Nos componentes internos, iremos substituir os parâmetros de *width* e *height* por *flex*. Esse parâmetro vai receber um número inteiro que representa a proporção do elemento frente aos demais.

Acompanhe a aula e faça as alterações no código. Ao final, seu objeto *styles* deve ficar como se observa abaixo:

```
const styles = StyleSheet.create({  
  
  container: {  
    backgroundColor: '#0F0',  
    flex: 1  
  },  
  red: {  
    backgroundColor: '#F00'  
  },  
  pequeno: {  
    flex: 1  
  },  
  medio: {
```

```
    flex: 2
  },
  grande: {
    flex: 3
  },
  blue: {
    backgroundColor: 'powderblue'
  }
})
```

Observe que cada componente interno utilizou uma propriedade de tamanho. A propriedade "pequena" utiliza *flex: 1*, a "média" utiliza *flex: 2* e a "grande" usa o *flex: 3*, isto é, **1+2+3 = 6**.

Isso significa que a área, ao utilizar a propriedade "pequena", obterá 1/6 do espaço, a média 2/6 e a grande 3/6.

Resumo

Nesta aula, vimos uma maneira mais usual de trabalhar com estilização de componentes do *React Native*. Dentre os conceitos e práticas, foi visto como trabalhar com *StyleSheet* e dimensões.