

# **Programação Mobile**

## **React Native - Aula 02**

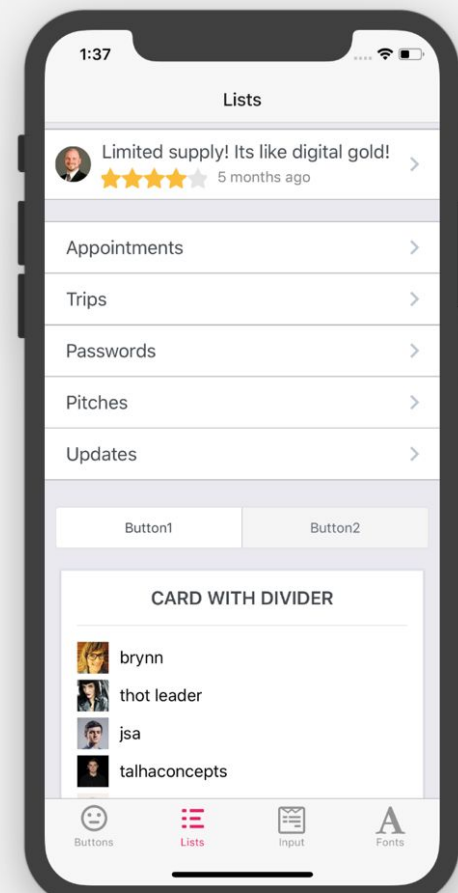
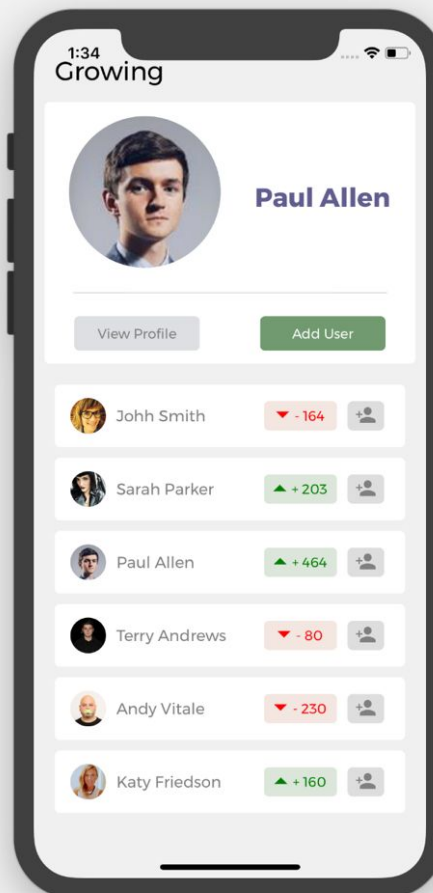
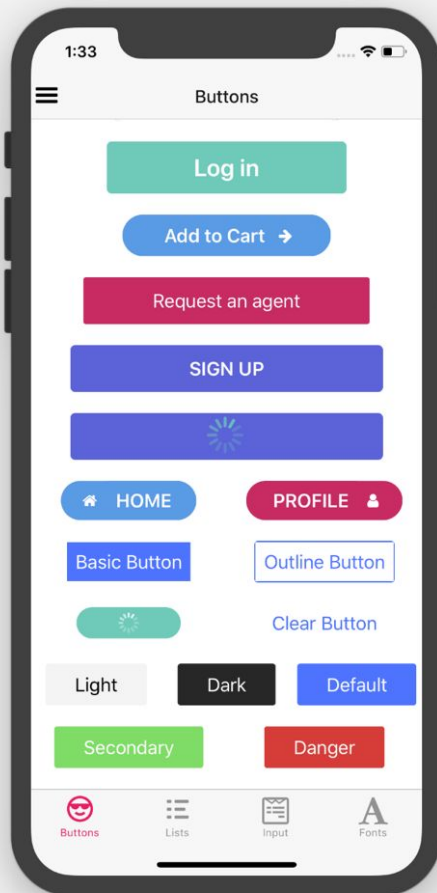
Professor: João Felipe Bragança



# Objetivos da Aula

- Entender os Componentes
- Criar Componentes Funcionais
- Utilizar Propriedades

# Componentes



# Componentes

- Os componentes permitem que você divida a interface do usuário em partes independentes e reutilizáveis e pense em cada parte isoladamente.
- Podemos utilizar componentes prontos, como `<Text>`, `<Image>` e `<Button>`.
- Podemos criar nossos próprios componentes.
- Existem componentes de função, que são mais modernos. Iremos sempre utilizar componentes funcionais.
- Podemos utilizar componentes de Classe. São mais verbosos e se tornaram obsoletos com o surgimento dos Hooks. Veremos em aulas posteriores.

# Componente Funcional



```
1  import { StatusBar } from 'expo-status-bar';
2  import { StyleSheet, Text, View } from 'react-native';
3
4  export default function App() {
5    return (
6      <View style={styles.container}>
7        <Text>Open up App.js to start working on your app!</Text>
8        <StatusBar style="auto" />
9      </View>
10   );
11 }
12
13 const styles = StyleSheet.create({
14   container: {
15     flex: 1,
16     backgroundColor: '#fff',
17     alignItems: 'center',
18     justifyContent: 'center',
19   },
20 });
```

# Componente Funcional

- O App.js é um ótimo exemplo de um componente funcional
- Um Componente Funcional é sempre uma função que começa com a letra maiúscula e retorna um elemento visual, JSX
- Para que a aplicação possa utilizar um componente, é necessário exportá-lo
- Podemos utilizar apenas um `export default` por componente.
- Podemos utilizar também o `export`, quantas vezes forem necessários.
- A forma de exportar muda a forma como deve-se importar

```
export default function App(){}  
import nomeDaMinhaEscolha from './App'  
  
export function App(){}  
import { App } from './App'|
```

# Componente News.js



```
1  import { View, Text } from "react-native";
2
3  export default function News() {
4    return (
5      <View>
6        <Text>Notícia</Text>
7      </View>
8    );
9  }
```

- Foi criado um componente funcional chamado “News”
- Apenas exibe um texto fixo com o conteúdo “Notícias”
- Foi exportado com default, então pode ser importado com qualquer nome
- É um componente estático, sempre vai exibir a mesma coisa
- Porém, pode ser chamado quantas vezes for preciso.
- Fica claro o reuso de código, podemos utilizar várias vezes escrevendo apenas uma vez.



# Utilizando News no App.js



```
1  import { StatusBar } from "expo-status-bar";
2  import { StyleSheet, Text, View } from "react-native";
3  import News from "../src/components/News";
4
5  export function App() {
6    return (
7      <View style={styles.container}>
8        <News />
9        <News />
10       <News />
11      </View>
12    );
13  }
```

- News foi importado com o mesmo nome do componente, News
- Foi utilizado 3 vezes, porém exibirá sempre a mesma mensagem
- Para explorar todo potencial de um componente, podemos enviar dados através de propriedades.
- Podemos passar textos, objetos, states e funções.
- O componente pai App pode passar qual a notícia que o componente filho News deverá exibir
- Basta nomear uma propriedade na chamada do componente News, em App, e receber tais propriedades dentro do componente News



# Passagem de Props



```
JS App.js M x
aula-02 > JS App.js > ...
1 import { StyleSheet, Text, View } from "react-native";
2 import News from "../src/components/News";
3
4 export function App() {
5   return (
6     <View style={styles.container}>
7       <News noticia="Primeira Notícia" />
8       <News noticia="Segunda Notícia" />
9       <News noticia="Terceira Notícia" />
10    </View>
11  );
12 }
13
14 const styles = StyleSheet.create({
15   container: {
16     flex: 1,
17     backgroundColor: "#fff",
18     alignItems: "center",
19     justifyContent: "center",
20   },
21 });
22
```

```
JS News.js U x
aula-02 > src > components > JS News.js > ...
3 //Dentro dos () da função, podemos receber
4 //as propriedades que vem com componente pai
5 export default function News(props) {
6   return (
7     <View>
8       /* Podemos utilizar as props como se fossem variáveis */
9       /* Basta colocar props.nomeDaPropriedade */
10      <Text>{props.noticia}</Text>
11    </View>
12  );
13 }
14
```

# Componente Funcional

- O App.js é um ótimo exemplo de um componente funcional
- Um Componente Funcional é sempre uma função que começa com a letra maiúscula e retorna um elemento visual, JSX
- Para que a aplicação possa utilizar um componente, é necessário exportá-lo
- Podemos utilizar apenas um `export default` por componente.
- Podemos utilizar também o `export`, quantas vezes forem necessários.
- A forma de exportar muda a forma como deve-se importar

```
export default function App(){}  
import nomeDaMinhaEscolha from './App'  
  
export function App(){}  
import { App } from './App'|
```

# Saiba Mais

- Componentes Funcionais: <https://reactnative.dev/docs/intro-react>
- Conhecendo o legado Componente de Classe:  
<https://pt-br.legacy.reactjs.org/docs/react-component.html>
- Conheça mais sobre os Componentes do React Native:  
<https://reactnative.dev/docs/components-and-apis>



**Obrigado(a)!**