

## Mobile

**Aula 04 - React Native** 

Prof. MSc. Kelson Almeida

## Introdução à Context API

React Native

- Context API: uma ferramenta para gerenciar e acessar o estado globalmente em uma aplicação React.
- Resolução de problema de prop drilling, onde os dados são passados por vários níveis de componentes.



### Criando um Contexto

React Native

 createContext: uso para criar um novo contexto.

```
javascript

import { createContext } from 'react';

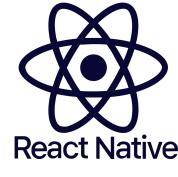
const ThemeContext = createContext('light'); // Valor padrão
```



### Provedor de Contexto

Context.Provider: permite que componentes aninhados consumam o contexto.

```
// sx
```





### Consumindo Contexto

React Native

 Como componentes podem acessar o valor do contexto usando o hook useContext.

```
import React, { useContext } from 'react';
import { ThemeContext } from './ThemeContext';

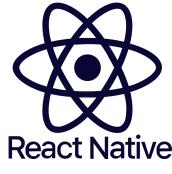
function ThemedComponent() {
  const theme = useContext(ThemeContext);
  return <View style={{ backgroundColor: theme === 'dark' ? '#333' : '#FFF' }} />
}
```



### **Atualizando Contexto**

 Como o contexto pode ser dinâmico e reativo, respondendo a mudanças de

estado.





# Vamos praticar? [com o prof]

#### Exercício Prático: Implementando um Tema Escuro/Claro com Context API

O objetivo deste exercício é criar um toggle de tema claro e escuro utilizando a Context API no React Native, o que ajudará a entender como gerenciar e utilizar o estado global em uma aplicação.

#### Passo a Passo

- Criar o Contexto de Tema: Crie um arquivo chamado ThemeContext.js para definir o contexto e seu provedor.
- Integrar o ThemeProvider no App: No arquivo App.js, envolva o componente principal com o ThemeProvider para fornecer o estado do tema aos componentes descendentes.
- Usar o Contexto para Aplicar o Tema: Em qualquer componente que deseje utilizar o tema, use o useContext para acessar o theme e toggleTheme.

