

REACT-NATIVE PT.2

PROF. FERNANDO LUCAS

CONTINUANDO COM

☐ Conteúdo técnico

- ☐ Utilização de Fontes Personalizadas
- ☐ Melhorias na SignIn
- ☐ Interface Home
- ☐ Interface de detalhes do servidor
- ☐ Navegação e Rotas com React Stack Navigation
- ☐ SVG
- ☐ Gradients



NOVAS FONTES

- Se não certarmos uma fonte para nosso projeto, o React vai utilizar a fonte padrão do dispositivo.
- Vamos acessar a documentação oficial do expo.
- <https://docs.expo.dev/develop/user-interface/fonts/#use-a-google-font>
- E vamos verificar como funciona a importação de novas fonts.



Google Fonts

RELEMBRANDO GOOGLE FONTS.

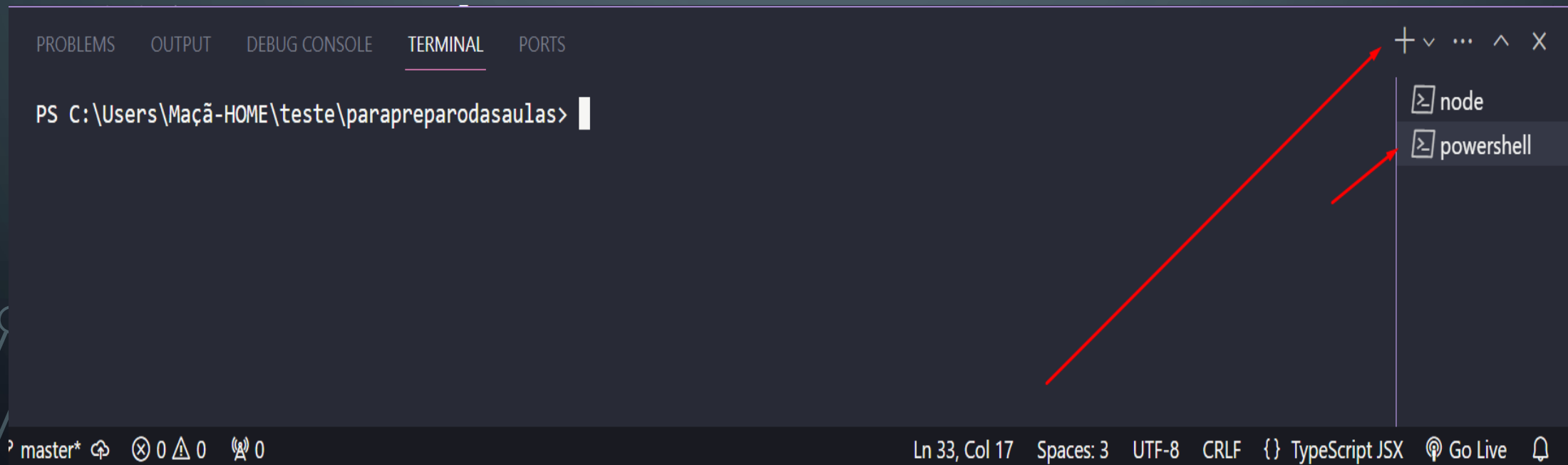
- O google fonts é a plataforma da Google que disponibiliza fonts gratuitas, ou seja fontes sem direitos autorais que podem ser utilizadas em qualquer aplicação.

INSTALANDO A NOVA FONT

Diferente do HTML onde importamos a fonte aqui no React nós vamos instalar essa fonte nos arquivos da nossa aplicação, pois o usuário quando instalar nossa aplicação ele não tem como acessar a fonte.

INSTALANDO

- Abra um novo terminal (CTRL+J)



INSTALANDO

- Digite o comando
- Esse comando vai instalar, a componentização de fontes do expo (@expo-google) e o comando @expo-google-fonts/inter é responsável por instalar a fonte inter e suas variantes diretamente do google fonts.

```
npx expo install expo-font @expo-google-fonts/inter
```

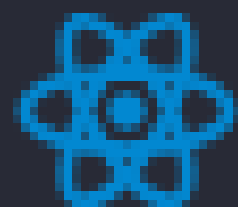
INSTALANDO

- Digite o comando
- o comando `@expo-google-fonts/rajdhani` é responsável por instalar a fonte rajdhani e suas variantes diretamente do google fonts.

```
npx expo install @expo-google-fonts/rajdhani
```


UTILIZANDO AS FONTES

- Você deve concordar que as fontes devem ser carregadas antes do aplicativo começar a ser utilizado, sendo assim precisamos utilizar o arquivo principal o primeiro que é carregado quando abrimos o aplicativo, você já sabe qual é ?



App.tsx

U

X

```
import {useFonts} from 'expo-font';
```

- Comando utilizado para importar o componente de utilização de fontes no expo.

```
import {Inter_400Regular, Inter_500Medium} from '@expo-google-fonts/inter';  
import {Rajdhani_500Medium, Rajdhani_700Bold} from '@expo-google-fonts/rajdhani';
```

- Comando utilizado para importar as fontes que foram instaladas no projeto.

```
import AppLoading from 'expo-app-loading';
```

- Este componente é responsável por fazer com a nossa tela splash continue aparecendo até que todas nossas fontes sejam carregadas.

- Código completo no próximo slide.

styles.ts ...\ButtonIcon U

styles.ts ...\screens U

index.tsx U

App.tsx U X

App.tsx > App

```
1  import React from 'react';
2  import {useFonts} from 'expo-font';
3  import {Inter_400Regular,Inter_500Medium} from '@expo-google-fonts/inter';
4  import {Rajdhani_500Medium, Rajdhani_700Bold} from '@expo-google-fonts/rajdhani';
5  import AppLoading from 'expo-app-loading';
6
7  import { SignIn } from './src/screens';
8
9  export default function App() {
10
11    const [fontsLoaded] =useFonts ({
12      Inter_400Regular,
13      Inter_500Medium,
14      Rajdhani_500Medium,
15      Rajdhani_700Bold
16    });
17    if (!fontsLoaded){
18      // npx expo install expo-app-loading
19      return <AppLoading/>
20    }
21
22    return(
23
24      <SignIn/>
25
26    );
```

CARREGANDO AS FONTES PARA NOSSO ARQUIVO GERAL DE TEMAS.

```
src > global > styles > theme.ts > theme > fonts
1  export const theme = {
2      colors:{
3          background: '#0D133D',
4          header: '#DDE3F0',
5          primary: '#E51C44',
6          line: '#991F36'
7      },
8
9
10     fonts:{
11         title700: 'rajdhani_700Bold',
12         title500: 'rajdhani_500Medium',
13         text400: 'Inter_400Regular',
14         text500: 'Inter_500Medium',
15     }
16 }
```

APLICANDO NOSSAS FONTES ATRAVÉS DO ARQUIVO DE TEMAS.

```
title:{  
  color:theme.colors.header,  
  fontSize:40,  
  marginBottom:16,  
  textAlign:'center',  
  fontFamily:theme.fonts.title700,  
  lineHeight:40  
},  
subtitle:{  
  color:theme.colors.header,  
  fontSize:15,  
  textAlign:'center',  
  ⚡marginBottom:64,  
  lineHeight:25  
}  
});
```

NOSSA STATUS-BAR TAMBÉM DEVE IR PARA O APP.TSX

index.tsx U App.tsx U theme.ts U

src > screens > index.tsx > SignIn

```
9      const [text, setText] = useState ("Você não digitou nada ainda");
10
11      return(
12
13        <View style = {styles.container}>
14
15          <StatusBar barStyle='light-content' backgroundColor={"transparent"} translucent />
16
```

App.tsx U X

```
import { StatusBar } from 'react-native'
```

```
return(
  <>
    <StatusBar barStyle='light-content' backgroundColor={"transparent"} translucent />
    <SignIn/>
  </>
)
```

ALTERANDO O ARQUIVO

TS theme.ts styles M X

```
export const theme = {
  colors: {
    primary: '#E51C44',

    secondary100: '#0A1033',
    secondary90: '#0D133D',
    secondary80: '#0E1647',
    secondary70: '#1B2565',
    secondary60: '#1B2565',
    secondary50: '#243189',
    secondary40: '#1D2766',
    secondary30: '#495BCC',

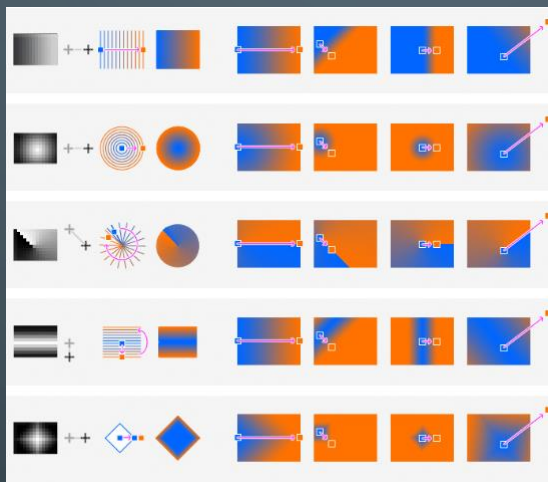
    overlay: 'rgba(0,0,0,0.7)',
    highlight: '#ABB1CC',
    heading: '#DDE3F0',
    line: '#991F36',
    on: '#32BD50',
  },

  fonts: {
    title700: 'Rajdhani_700Bold',
    title500: 'Rajdhani_500Medium',
    text400: 'Inter_400Regular',
    text500: 'Inter_500Medium',
  }
};
```

- Essas serão as cores do nosso gradiente.
- Quando você alterar é normal que aplicação perca algumas cores, você precisa aponta-las novamente no código.

CONSTRUINDO O GRADIENTE

- Também conhecido como gradiente, o degradê é um efeito em que uma cor se transforma gradualmente em outra sobre um fundo sólido, uma imagem ou plano de fundo.



PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

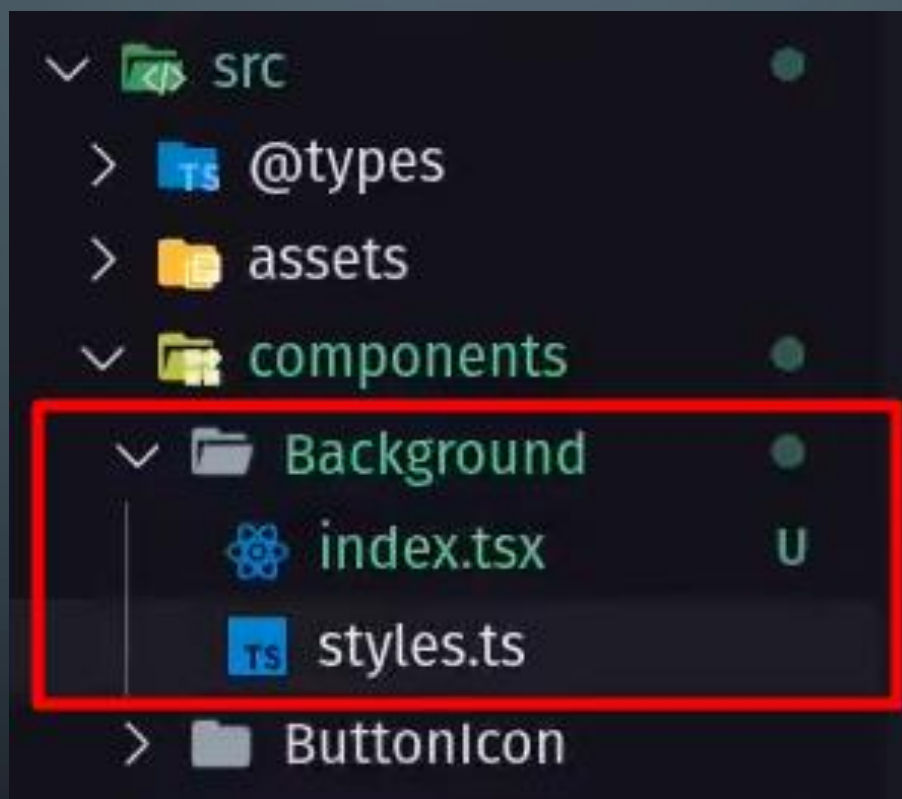
```
PS C:\Users\Maçã-HOME\teste\parapreparodasaulas> expo install expo-linear-gradient
```

+ v ... ^ x

node

powershell

CRIANDO O COMPONENTE GRADIENTE



EDITANDO ARQUIVOS DE ESTILOS

```
import { StyleSheet } from "react-native";
import { theme } from "../../global/styles/theme";

export const styles = StyleSheet.create ({
  container: {
    flex: 1
  }
});
```

- Respeitando o mesmo padrão dos outros arquivos início.
- Vamos apenas utilizar o flex1 para que o gradiente consiga cobrir a tela toda.

AGORA VAMOS EDITAR O ARQUIVO INDEX DO GRADIENTE.

- Inicialmente vamos trazer o arquivo de estilos e nosso arquivo theme, que possui todas as cores da nossa aplicação.

```
import React from "react";
import { LinearGradient } from "expo-linear-gradient";

import { styles } from "../styles";
import { theme } from "../../global/styles/theme";
```

- Criando a função que vai guardar nosso gradiente.

```
export function Background () {
  return (
    <LinearGradient>
    </LinearGradient>
  )
}
```

CONTINUE EDITANDO O ARQUIVO INDEX DO GRADIENTE.

- Vamos entregar o estilo container, Entregaremos também as cores do gradiente.

```
export function Backgorund () {  
  return (  
    <LinearGradient  
      style={ styles.container}  
      colors={['theme.colors.secondary80',theme.colors.secondary100]}  
    >  
  
    </LinearGradient>  
  )  
}
```

- Vamos refinar mais ainda o nosso código para deixar mais organizado.

```
export function Backgorund () {  
  const {secondary80,secondary100} = theme.colors  
  return (  
    <LinearGradient  
      style={ styles.container}  
      colors={['secondary80,secondary100]}  
    >  
  
    </LinearGradient>  
  )  
}
```

AGORA VAMOS CRIAR UM PROPRIEDADE

- Vamos criar uma propriedade para que possamos utilizar esse gradiente em todas as telas do nosso app.

```
export function Background({ children }: Props) {  
  const { secondary80, secondary100 } = theme.colors;  
  
  return (  
    <LinearGradient  
      style={styles.container}  
      colors={[secondary80, secondary100]}  
    >  
      {children}  
    </LinearGradient>  
  )  
}
```


AGORA NO ARQUIVO APP.TSX VAMOS INVOCAR NOSSO GRADIENTE.

- Primeiro como sempre vamos importar.

```
import { SignIn } from './src/screens/SignIn';  
import { Background } from './src/components/Background';
```

- Agora só colocar dentro da execução da nossa aplicação.

E já podemos verificar nosso degrade no app.

Lembre-se de retirar

o background da tela de

Sig-in.

```
return(  
  <Background>  
    <StatusBar  
      barStyle="light-content"  
      backgroundColor="transparent"  
      translucent  
    />  
    <SignIn />  
  </Background>  
);
```