

Objetivos

- Personalizar a aparência das abas inferiores adicionando ícones/imagens personalizados a elas.
- Projetar o formulário de entrada para a tela de retirada/retorno do aplicativo.
- Escrever o código para preencher automaticamente a caixa de entrada quando o ID do livro e os IDs do aluno forem digitalizados.

Para instalar essas fontes, vamos usar os comandos

- `expo install @expo-google-fonts/rajdhani`
- `expo install expo-font@~8.4.0`
- `npm install react-native vector-icons`

Vamos importa as bibliotecas e estruturar as funções necessárias.

```
import { Rajdhani_600SemiBold } from "@expo-google-fonts/rajdhani";
import * as Fonts from "expo-font";

export default class App extends Component {
  constructor() {
    super();
    this.state = {
      fontsLoaded: false
    };
  }

  async loadFonts() {
    await Fonts.loadFonts({
      Rajdhani_600SemiBold: Rajdhani_600SemiBold
    });
    this.setState({ fontsLoaded: true });
  }

  componentDidMount() {
    this.loadFonts();
  }
}
```

Para garantir que as fontes sejam carregadas antes de renderizar qualquer coisa na tela, vamos escrever uma condição para verificar isso

```
    this.setState({fontsLoaded:true});
  }

  componentDidMount(){
    this.loadFonts();
  }
  render() {
    const { fontLoaded } = this.state;
    if (fontLoaded) {
      return <BottomTabNavigator />;
    }
    return null;
  }
}
```

(property) App.state: {
 fontsLoaded: boolean;
}

usaremos uma função predefinida chamada `screenOptions`. Esta função usa o objeto `route` (rota) como parâmetro padrão. Agora vamos definir o campo `tabBarIcon` dentro de `screenOption`. O `tabBarIcon` deve retornar um componente (dependendo da aba que está selecionada).

- `focused` (focado): Retorna `true` (verdadeiro) se a aba foi tocada. Caso contrário, retorna `false` (falso).
- `horizontal`: Retorna `true` (verdadeiro) se o dispositivo estiver no modo paisagem. Retorna `false` (falso) se o dispositivo estiver no modo retrato.
- `tintColor` (cor da tinta): Retorna o `tintColor` (cor da tinta) que está ativo.

```
import Ionicons from "react-native-vector-icons/Ionicons";

import { Component } from "react/cjs/react.production.min";

const Tab = createBottomTabNavigator();

export default class BottomTabNavigator extends Component{

  render(){
    return (
      <NavigationContainer>
        <Tab.Navigator
          screenOptions={({ route }) => ({
            tabBarIcon: ({ focused, color, size }) => {
              let iconName;

              if (route.name === "Transação") {
                iconName = "book";
              } else if (route.name === "Pesquisa") {
                iconName = "search";
              }

              return (
                <Ionicons
                  name={iconName}
                  size={size}
                  color={color}
                  // size={size}
                />
              )
            }
          })
        />
      </NavigationContainer>
    )
  }
}
```

```

        <Icons
          name={iconName}
          size={size}
          color={color}
          // size={size}
        />
      );
    }
  })}
  tabBarOptions={{
    activeTintColor: "#FFFFFF",
    inactiveTintColor: "black",
    style: {
      height: 130,
      borderTopWidth: 0,
      backgroundColor: "#5653d4"
    },
    labelStyle: {
      fontSize: 20,
      fontFamily: "Rajdhani_600SemiBold"
    },
    labelPosition: "beside-icon",
    tabStyle: {
      marginTop: 25,
      marginLeft: 10,
      marginRight: 10,
      borderRadius: 30,
      borderWidth: 2,
      alignItems: "center",
      justifyContent: "center",
      backgroundColor: "#5653d4"
    }
  }}
}

```

>