**Título**

Múltiplas Vistas: Organizando seu App com Navegação por Abas

**Objetivos**

* Entender a função e os benefícios do **BottomTabNavigator**.
* Configurar o **React Navigation** para navegação por abas.
* Integrar diferentes telas (ex: Lista, Detalhes, Favoritos, Mapa) em abas.
* Personalizar ícones e nomes das abas.
* Aprender a aninhar navegadores (ex: Stack dentro de Tabs).
* **BÔNUS:** Discussão sobre padrões de navegação e experiência do usuário (UX).

**Revisão Rápida**

* Como usamos AsyncStorage para persistir dados localmente?
* De que forma o **Context API** simplifica o gerenciamento de estado global, como uma lista de favoritos?

**Conteúdo Teórico**

* **Padrões de Navegação Mobile:**
  + **Tab Navigation:** Ideal para funções principais do aplicativo, acessíveis a qualquer momento.
  + Drawer Navigation (menu lateral): Para funcionalidades menos acessadas ou configurações.
  + Stack Navigation: Para aprofundar em detalhes ou fluxos específicos.
* **Introdução ao BottomTabNavigator:**
  + Componente do React Navigation que exibe uma barra de abas na parte inferior da tela.
  + Cada aba representa uma rota principal do aplicativo.
  + Fácil alternância entre seções.
* **Configurando BottomTabNavigator:**
  + Instalação da biblioteca específica.
  + Uso de createBottomTabNavigator().
  + Propriedades screenOptions para estilização global das abas.
  + Personalização individual de cada aba usando options em Tab.Screen.
* **Aninhamento de Navegadores:**
  + Como combinar um Stack Navigator dentro de um Tab Navigator.
  + Por que é importante: manter um histórico de navegação dentro de cada aba.
* **Ícones para Abas:**
  + Utilização de bibliotecas de ícones (ex: @expo/vector-icons).
  + Propriedades tabBarIcon e tabBarLabel.
* **Experiência do Usuário (UX):**
  + Clareza dos ícones e rótulos das abas.
  + Ordem das abas (as mais importantes primeiro).
  + Não sobrecarregar a barra de abas (geralmente 3-5 abas).

**Prática Laboratorial**

**Passo 1: Instalar o BottomTabNavigator**

* **Ação:** Primeiro, instale a biblioteca específica para o navegador de abas.
* **Comando:** No terminal, na raiz do seu projeto:
* npm install @react-navigation/bottom-tabs
* **Explicação:**
  + Este comando adiciona a biblioteca bottom-tabs ao seu projeto, permitindo que você crie a navegação por abas.

**Passo 2: Criar as Novas Telas (se necessário)**

* **Ação:** Para ter abas, precisamos de telas para cada uma delas. Se você já tem ListaPontosTuristicos e DetalhesPontoTuristico, ótimo. Vamos criar uma tela simples FavoritosScreen.js e uma MapaScreen.js para as outras abas.
* **Estrutura de Pastas (Exemplo de Telas Adicionais):**
* guia-turistico-app/
* ├── App.js
* ├── components/
* ├── context/
* ├── screens/
* │ ├── DetalhesPontoTuristico.js
* │ ├── FavoritosScreen.js # <-- Novo arquivo
* │ ├── ListaPontosTuristicos.js
* │ └── MapaScreen.js # <-- Novo arquivo
* └── services/
* **Conteúdo de screens/FavoritosScreen.js (Simples):**
* // screens/FavoritosScreen.js
* import React from 'react';
* import { View, Text, StyleSheet } from 'react-native';
* import { useFavorites } from '../context/FavoritesContext'; // <--- Importe o contexto
* const FavoritosScreen = () => {
* const { favoriteIds, isLoadingFavorites } = useFavorites(); // <--- Use o hook do contexto
* if (isLoadingFavorites) {
* return (
* <View style={styles.container}>
* <Text>Carregando favoritos...</Text>
* </View>
* );
* }
* return (
* <View style={styles.container}>
* <Text style={styles.title}>Meus Favoritos</Text>
* {favoriteIds.length === 0 ? (
* <Text>Nenhum ponto turístico favoritado ainda.</Text>
* ) : (
* <Text>IDs Favoritos: {favoriteIds.join(', ')}</Text>
* // Em aulas futuras, você listaria os cards completos aqui
* )}
* </View>
* );
* };
* const styles = StyleSheet.create({
* container: { flex: 1, justifyContent: 'center', alignItems: 'center', backgroundColor: '#f0f8ff' },
* title: { fontSize: 24, fontWeight: 'bold', marginBottom: 20 },
* });
* export default FavoritosScreen;
* **Conteúdo de screens/MapaScreen.js (Simples):**
* // screens/MapaScreen.js
* import React from 'react';
* import { View, Text, StyleSheet } from 'react-native';
* const MapaScreen = () => {
* return (
* <View style={styles.container}>
* <Text style={styles.title}>Mapa dos Pontos</Text>
* <Text>O mapa será implementado aqui em breve!</Text>
* </View>
* );
* };
* const styles = StyleSheet.create({
* container: { flex: 1, justifyContent: 'center', alignItems: 'center', backgroundColor: '#e6ffe6' },
* title: { fontSize: 24, fontWeight: 'bold', marginBottom: 20 },
* });
* export default MapaScreen;
* **Explicação:**
  + Criamos telas mínimas para que o BottomTabNavigator tenha o que renderizar.
  + A tela de Favoritos já usa nosso Contexto.

**Passo 3: Configurar o BottomTabNavigator no App.js**

* **Ação:** Substitua o Stack Navigator principal por um Tab Navigator e aninhe um Stack Navigator dentro de uma das abas (Lista/Detalhes).
* **Mudança em App.js:**
* // App.js
* import 'react-native-gesture-handler';
* import React from 'react';
* import { NavigationContainer } from '@react-navigation/native';
* import { createStackNavigator } from '@react-navigation/stack';
* import { createBottomTabNavigator } from '@react-navigation/bottom-tabs'; // <--- Importe createBottomTabNavigator
* import { Ionicons } from '@expo/vector-icons'; // <--- Importe Ionicons para os ícones das abas
* import { FavoritesProvider } from './context/FavoritesContext';
* // Importe suas telas
* import ListaPontosTuristicos from './screens/ListaPontosTuristicos';
* import DetalhesPontoTuristico from './screens/DetalhesPontoTuristico';
* import FavoritosScreen from './screens/FavoritosScreen'; // <--- Nova tela
* import MapaScreen from './screens/MapaScreen'; // <--- Nova tela
* const Stack = createStackNavigator(); // <--- Stack Navigator para a aba "Explorar"
* const Tab = createBottomTabNavigator(); // <--- Tab Navigator principal
* // <--- Componente para a Aba de Exploração (Lista + Detalhes)
* function ExplorarStack() {
* return (
* <Stack.Navigator>
* <Stack.Screen
* name="ListaPontos"
* component={ListaPontosTuristicos}
* options={{ title: 'Explorar Pontos' }} // Título do cabeçalho
* />
* <Stack.Screen
* name="DetalhesPonto"
* component={DetalhesPontoTuristico}
* options={{ title: 'Detalhes do Ponto' }}
* />
* </Stack.Navigator>
* );
* }
* export default function App() {
* return (
* <FavoritesProvider>
* <NavigationContainer>
* <Tab.Navigator
* screenOptions={({ route }) => ({
* // <--- Configuração dos ícones das abas
* tabBarIcon: ({ focused, color, size }) => {
* let iconName;
* if (route.name === 'Explorar') {
* iconName = focused ? 'compass' : 'compass-outline';
* } else if (route.name === 'Favoritos') {
* iconName = focused ? 'heart' : 'heart-outline';
* } else if (route.name === 'Mapa') {
* iconName = focused ? 'map' : 'map-outline';
* }
* return <Ionicons name={iconName} size={size} color={color} />;
* },
* tabBarActiveTintColor: 'blue', // Cor do ícone/texto da aba ativa
* tabBarInactiveTintColor: 'gray', // Cor do ícone/texto da aba inativa
* headerShown: false, // <--- Esconde o cabeçalho duplicado que viria do Tab Navigator
* })}
* >
* <Tab.Screen
* name="Explorar"
* component={ExplorarStack} // <--- Usa o Stack Navigator como componente da aba
* options={{ tabBarLabel: 'Explorar' }} // Rótulo na aba
* />
* <Tab.Screen
* name="Favoritos"
* component={FavoritosScreen}
* options={{ tabBarLabel: 'Favoritos' }}
* />
* <Tab.Screen
* name="Mapa"
* component={MapaScreen}
* options={{ tabBarLabel: 'Mapa' }}
* />
* </Tab.Navigator>
* </NavigationContainer>
* </FavoritesProvider>
* );
* }
* **Entendimento:**
  + createBottomTabNavigator(): Cria a base para nossa navegação por abas.
  + ExplorarStack(): É uma função que encapsula nosso Stack Navigator existente. Isso é o **aninhamento de navegadores**: a aba "Explorar" agora contém sua própria pilha de telas (ListaPontos e DetalhesPonto).
  + Tab.Navigator: Componente principal para definir as abas.
  + screenOptions: Propriedade crucial para configurar o estilo e os ícones de todas as abas globalmente.
    - tabBarIcon: Uma função que retorna o ícone com base no estado focused da aba.
    - headerShown: false: Esconde o cabeçalho padrão do Tab Navigator, pois queremos que os cabeçalhos sejam gerenciados pelo Stack Navigator aninhado.
  + Tab.Screen: Define cada aba, com seu name, component e options (para rótulo, etc.).

**Desafio da Aula**

* No seu aplicativo "Guia Turístico":
  + Instale o @react-navigation/bottom-tabs.
  + Crie as telas FavoritosScreen.js e MapaScreen.js (podem ser simples inicialmente).
  + No App.js:
    - Mantenha seu FavoritesProvider envolvendo tudo.
    - Crie um Stack Navigator separado (ex: ExplorarStack) para as telas ListaPontosTuristicos e DetalhesPontoTuristico.
    - Crie um BottomTabNavigator principal.
    - Defina três abas: "Explorar" (usando o ExplorarStack como componente), "Favoritos" (usando FavoritosScreen), e "Mapa" (usando MapaScreen).
    - Use screenOptions no Tab.Navigator para definir ícones para cada aba (usando Ionicons do @expo/vector-icons) e personalizar as cores dos ícones/texto ativos/inativos.
    - Certifique-se de que a prop headerShown do Tab.Navigator esteja configurada corretamente para evitar cabeçalhos duplicados.
* Faça **commits no Git** com as alterações.

**Próximos Passos**

* Na próxima aula, vamos adicionar a funcionalidade de mapa usando react-native-maps e exibir os pontos turísticos diretamente no mapa.