App de viagens

Nessa aula vamos criar um app para fazer viagens e para isso vamos criar um projeto. Abra o VS Code na pasta onde você deseja criar o seu projeto e em seguida abra o terminal do Vs Code e digite o comando:

```
npx create-expo-app –template
```

Na hora de colar no terminal, certifique-se que tem dois traços (--) antes da palavra “template”, para não dar erro na hora da instalação.

Em seguida escolha a opção “**Default**” e dê um nome ao seu projeto.

Depois de criado o seu projeto, **entre na pasta do seu projeto para realizar a instalação das dependências** que serão utilizadas no projeto, vá até o arquivo **package.json**:

package.json

```
{
  "name": "expo-travel-app",
  "main": "expo-router/entry",
  "version": "1.0.0",
  "scripts": {
    "start": "expo start",
    "android": "expo start --android",
    "ios": "expo start --ios",
    "web": "expo start --web",
    "test": "jest --watchAll"
  },
  "jest": {
    "preset": "jest-expo"
  },
  "dependencies": {
    "@expo/vector-icons": "^14.0.0",
    "@react-navigation/native": "^6.0.2",
    "expo": "~50.0.14",
    "expo-font": "~11.10.3",
```

```
    "expo-linking": "~6.2.2",
    "expo-router": "~3.4.8",
    "expo-splash-screen": "~0.26.4",
    "expo-status-bar": "~1.11.1",
    "expo-system-ui": "~2.9.3",
    "expo-web-browser": "~12.8.2",
    "react": "18.2.0",
    "react-dom": "18.2.0",
    "react-native": "0.73.6",
    "react-native-safe-area-context": "4.8.2",
    "react-native-screens": "~3.29.0",
    "react-native-web": "~0.19.6",
    "react-native-reanimated": "~3.6.2"
  },
  "devDependencies": {
    "@babel/core": "^7.20.0",
    "@types/react": "~18.2.45",
    "jest": "^29.2.1",
    "jest-expo": "~50.0.4",
    "react-test-renderer": "18.2.0",
    "typescript": "^5.1.3"
  },
  "private": true
}
```

Em seguida, abra o arquivo **babel.config.js**.

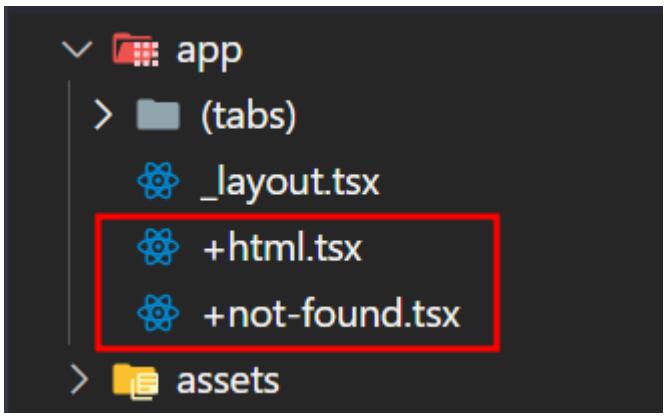
babel.config.js

```
module.exports = function (api) {
  api.cache(true);
  return {
```

```
  presets: ["babel-preset-expo"],
  plugins: ["react-native-reanimated/plugin"],
 };
};
```

Agora no seu terminal digite “**npm i**” para ser instalado as dependências que iremos utilizar.

Em seguida, dentro de **app** exclua os arquivos a seguir:



E agora, vamos configurar o arquivo _layout.tsx

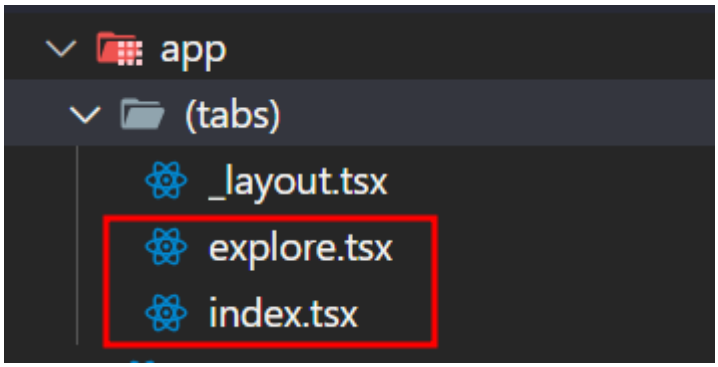
app/_layout.tsx

```
import FontAwesome from "@expo/vector-icons/FontAwesome";
import { useFonts } from "expo-font";
import { Stack } from "expo-router";
import * as SplashScreen from "expo-splash-screen";
import { useEffect } from "react";


export {
 // Catch any errors thrown by the Layout component.
 ErrorBoundary,
} from "expo-router";
```

```
export const unstable_settings = {
  // Ensure that reloading on `/modal` keeps a back button present.
  initialRouteName: "(tabs)",
};

// Prevent the splash screen from auto-hiding before asset loading is complete.
SplashScreen.preventAutoHideAsync();

export default function RootLayout() {
  const [loaded, error] = useFonts({
    SpaceMono: require("../assets/fonts/SpaceMono-Regular.ttf"),
    ...FontAwesome.font,
  });

  // Expo Router uses Error Boundaries to catch errors in the navigation tree.
  useEffect(() => {
    if (error) throw error;
  }, [error]);

  useEffect(() => {
    if (loaded) {
      SplashScreen.hideAsync();
    }
  }, [loaded]);

  if (!loaded) {
    return null;
  }
```

```
  return <RootLayoutNav />;
}

function RootLayoutNav() {
  return (
    <Stack>
      <Stack.Screen name="(tabs)" options={{ headerShown: false }} />
    </Stack>
  );
}
```

Agora vamos vá em **app/(tabs) exclua os seguintes arquivos:**



E agora vamos editar o arquivo _layout.tsx

app/(tabs)/_layout.tsx

```
import { View, Text } from "react-native";
import React from "react";
import { Tabs } from "expo-router";
import { FontAwesome, Ionicons, MaterialIcons } from "@expo/vector-icons";
import Colors from "@/constants/Colors";

export default function Layout() {
```

```
return (
 <Tabs
  screenOptions={{
   tabBarStyle: {
    backgroundColor: Colors.bgColor,
    borderTopWidth: 0,
    padding: 0,
   },
   tabBarShowLabel: false,
   tabBarActiveTintColor: Colors.black,
   tabBarInactiveTintColor: "#999",
  }}
 >
  <Tabs.Screen
   name="index"
   options={{
    tabBarIcon: ({ color }) => (
     <Ionicons name="compass" size={28} color={color} />
    ),
   }}
  />
  <Tabs.Screen
   name="category"
   options={{
    tabBarIcon: ({ color }) => (
     <MaterialIcons name="space-dashboard" size={28} color={color} />
    ),
   }}
```

```
/>
<Tabs.Screen
 name="search"
 options={{
  tabBarIcon: ({ color }) => (
   <View
    style={{
     backgroundColor: Colors.primaryColor,
     paddingHorizontal: 16,
     paddingVertical: 12,
     borderRadius: 10,
     height: 50,
    }}
   >
    <Ionicons name="search-outline" size={24} color={Colors.white} />
   </View>
  ),
 }}
/>
<Tabs.Screen
 name="bookmarks"
 options={{
  tabBarIcon: ({ color }) => (
   <Ionicons name="bookmark" size={28} color={color} />
  ),
 }}
/>
<Tabs.Screen
```

```
      name="profile"
      options={{
       tabBarIcon: ({ color }) => (
        <FontAwesome name="user" size={28} color={color} />
       ),
      }}
     />
   </Tabs>
 );
}
```

E agora na mesma pasta vamos criar as nossas tabs de navegação.

app/(tabs)/bookmarks.tsx

```
import { StyleSheet, Text, View } from "react-native";
import React from "react";

const Page = () => {
 return (
  <View style={styles.container}>
   <Text>Favoritos</Text>
  </View>
 );
};

export default Page;

const styles = StyleSheet.create({
```

```
  container: {
    flex: 1,
    justifyContent: "center",
    alignItems: "center",
  },
});
```

app/(tabs)/category.tsx

```
import { StyleSheet, Text, View } from "react-native";
import React from "react";


const Page = () => {
  return (
    <View style={styles.container}>
      <Text>Categoria</Text>
    </View>
  );
};


export default Page;


const styles = StyleSheet.create({
  container: {
    flex: 1,
    justifyContent: "center",
    alignItems: "center",
  },
});
```

app/(tabs)/index.tsx

```
import {
  Image,
  ScrollView,
  StyleSheet,
  Text,
  TextInput,
  TouchableOpacity,
  View,
} from "react-native";
import React, { useState } from "react";
import { Stack } from "expo-router";
import { Ionicons } from "@expo/vector-icons";
import Colors from "@/constants/Colors";
import { useHeaderHeight } from "@react-navigation/elements";
import CategoryButtons from "@/components/CategoryButtons";
import Listings from "@/components/Listings";
import listingData from "@/data/destinations.json";
import GroupListings from "@/components/GroupListings";
import groupData from "@/data/groups.json";


const Page = () => {
  const headerHeight = useHeaderHeight();
  const [category, setCategory] = useState("All");


  const onCatChanged = (category: string) => {
    console.log("Categpry: ", category);
```

```
  setCategory(category);
};

return (
  <>
   <Stack.Screen
    options={{
     headerTransparent: true,
     headerTitle: "",
     headerLeft: () => (
      <TouchableOpacity onPress={() => {}} style={{ marginLeft: 20 }}>
       <Image
        source={{
         uri: "https://xsgames.co/randomusers/avatar.php?g=female",
        }}
        style={{ width: 40, height: 40, borderRadius: 10 }}
       />
      </TouchableOpacity>
     ),
     headerRight: () => (
      <TouchableOpacity
       onPress={() => {}}
       style={{
        marginRight: 20,
        backgroundColor: Colors.white,
        padding: 10,
        borderRadius: 10,
        shadowColor: "#171717",
```

```
        shadowOffset: { width: 2, height: 4 },
        shadowOpacity: 0.2,
        shadowRadius: 3,
      }}
    >
      <Ionicons name="notifications" size={20} color={Colors.black} />
    </TouchableOpacity>
  ),
 }}
/>
<View style={[styles.container, { paddingTop: headerHeight }]}>
 <ScrollView showsVerticalScrollIndicator={false}>
  <Text style={styles.headingTxt}>Explore o Mundo!</Text>

  <View style={styles.searchSectionWrapper}>
   <View style={styles.searchBar}>
    <Ionicons
     name="search"
     size={18}
     style={{ marginRight: 5 }}
     color={Colors.black}
    />
    <TextInput placeholder="Pesquise aqui..." />
   </View>
   <TouchableOpacity onPress={() => {}} style={styles.filterBtn}>
    <Ionicons name="options" size={28} color={Colors.white} />
   </TouchableOpacity>
  </View>
```

```
      <CategoryButtons onCagtegoryChanged={onCatChanged} />

      <Listings listings={listingData} category={category} />

      <GroupListings listings={groupData} />
     </ScrollView>
    </View>
   </>
  );
};

export default Page;

const styles = StyleSheet.create({
 container: {
  flex: 1,
  paddingHorizontal: 20,
  backgroundColor: Colors.bgColor,
 },
 headingTxt: {
  fontSize: 28,
  fontWeight: "800",
  color: Colors.black,
  marginTop: 10,
 },
 searchSectionWrapper: {
  flexDirection: "row",
```

```
  marginVertical: 20,
 },
 searchBar: {
  flex: 1,
  flexDirection: "row",
  backgroundColor: Colors.white,
  padding: 16,
  borderRadius: 10,
 },
 filterBtn: {
  backgroundColor: Colors.primaryColor,
  padding: 12,
  borderRadius: 10,
  marginLeft: 20,
 },
});
```

app/(tabs)/profile.tsx

```
import { StyleSheet, Text, View } from "react-native";
import React from "react";


const Page = () => {
 return (
  <View style={styles.container}>
   <Text>Perfil</Text>
  </View>
 );
```

```
};

export default Page;

const styles = StyleSheet.create({
  container: {
    flex: 1,
    justifyContent: "center",
    alignItems: "center",
  },
});
```

app/(tabs)/search.tsx

```
import { StyleSheet, Text, View } from "react-native";
import React from "react";

const Page = () => {
  return (
    <View style={styles.container}>
      <Text>Pesquisas</Text>
    </View>
  );
};

export default Page;

const styles = StyleSheet.create({
```

```
  container: {
    flex: 1,
    justifyContent: "center",
    alignItems: "center",
  },
});
```

Com isso nossas abas de navegação estão criadas. Agora vamos criar uma pasta dentro de app chamada listing e nela o arquivo [id].tsx. Ele será o responsável por renderizar os detalhes de cada destino.

App/listing/[id].tsx

```
import {
  Dimensions,
  Image,
  ScrollView,
  StyleSheet,
  Text,
  TouchableOpacity,
  View,
} from "react-native";
import React from "react";
import { Stack, useLocalSearchParams, useRouter } from "expo-router";
import { ListingType } from "@/types/listingType";
import listingData from "@/data/destinations.json";
import {
  Feather,
  FontAwesome,
  FontAwesome5,
```

```
  Ionicons,
} from "@expo/vector-icons";
import Colors from "@/constants/Colors";
import Animated, {
  SlideInDown,
  interpolate,
  useAnimatedRef,
  useAnimatedStyle,
  useScrollViewOffset,
} from "react-native-reanimated";

const { width } = Dimensions.get("window");
const IMG_HEIGHT = 300;

const ListingDetails = () => {
  const { id } = useLocalSearchParams();
  const listing: ListingType = (listingData as ListingType[]).find(
    (item) => item.id === id
  );

  const router = useRouter();

  const scrollRef = useAnimatedRef<Animated.ScrollView>();
  const scrollOffset = useScrollViewOffset(scrollRef);
  const imageAnimatedStyle = useAnimatedStyle(() => {
    return {
      transform: [
        {
```

```
      translateY: interpolate(
        scrollOffset.value,
        [-IMG_HEIGHT, 0, IMG_HEIGHT],
        [-IMG_HEIGHT / 2, 0, IMG_HEIGHT * 0.75]
      ),
    },
    {
      scale: interpolate(
        scrollOffset.value,
        [-IMG_HEIGHT, 0, IMG_HEIGHT],
        [2, 1, 1]
      ),
    },
  ],
 };
});

return (
 <>
  <Stack.Screen
   options={{
    headerTransparent: true,
    headerTitle: "",
    headerLeft: () => (
     <TouchableOpacity
      onPress={() => router.back()}
      style={{
       backgroundColor: "rgba(255, 255, 255, 0.5)",
```

```
      borderRadius: 10,
      padding: 4,
    }}
  >
    <View
      style={{
        backgroundColor: Colors.white,
        padding: 6,
        borderRadius: 10,
      }}
    >
      <Feather name="arrow-left" size={20} />
    </View>
  </TouchableOpacity>
),
headerRight: () => (
  <TouchableOpacity
    onPress={() => {}}
    style={{
      backgroundColor: "rgba(255, 255, 255, 0.5)",
      borderRadius: 10,
      padding: 4,
    }}
  >
    <View
      style={{
        backgroundColor: Colors.white,
        padding: 6,
```

```
          borderRadius: 10,
        }}
      >
        <Ionicons name="bookmark-outline" size={20} />
      </View>
    </TouchableOpacity>
  ),
 }}
/>
<View style={styles.container}>
 <Animated.ScrollView
  ref={scrollRef}
  contentContainerStyle={{ paddingBottom: 150 }}
 >
  <Animated.Image
   source={{ uri: listing.image }}
   style={[styles.image, imageAnimatedStyle]}
  />
  <View style={styles.contentWrapper}>
   <Text style={styles.listingName}>{listing.name}</Text>
   <View style={styles.listingLocationWrapper}>
    <FontAwesome5
     name="map-marker-alt"
     size={18}
     color={Colors.primaryColor}
    />
    <Text style={styles.listingLocationTxt}>{listing.location}</Text>
   </View>
```

```
<View style={styles.highlightWrapper}>
 <View style={{ flexDirection: "row" }}>
  <View style={styles.highlightIcon}>
   <Ionicons name="time" size={18} color={Colors.primaryColor} />
  </View>
  <View>
   <Text style={styles.highlightTxt}>Duração</Text>
   <Text style={styles.highlightTxtVal}>
    {listing.duration} Dias
   </Text>
  </View>
 </View>
 <View style={{ flexDirection: "row" }}>
  <View style={styles.highlightIcon}>
   <FontAwesome
    name="users"
    size={18}
    color={Colors.primaryColor}
   />
  </View>
  <View>
   <Text style={styles.highlightTxt}>Pessoas</Text>
   <Text style={styles.highlightTxtVal}>{listing.duration}</Text>
  </View>
 </View>
 <View style={{ flexDirection: "row" }}>
  <View style={styles.highlightIcon}>
```

```
            <Ionicons name="star" size={18} color={Colors.primaryColor} />
          </View>
          <View>
            <Text style={styles.highlightTxt}>Avaliação</Text>
            <Text style={styles.highlightTxtVal}>{listing.rating}</Text>
          </View>
        </View>
      </View>

      <Text style={styles.listingDetails}>{listing.description}</Text>
    </View>
  </Animated.ScrollView>
</View>

<Animated.View style={styles.footer} entering={SlideInDown.delay(200)}>
  <TouchableOpacity
    onPress={() => {}}
    style={[styles.footerBtn, styles.footerBookBtn]}
  >
    <Text style={styles.footerBtnTxt}>Reserve agora</Text>
  </TouchableOpacity>
  <TouchableOpacity onPress={() => {}} style={styles.footerBtn}>
    <Text style={styles.footerBtnTxt}>R${listing.price}</Text>
  </TouchableOpacity>
</Animated.View>
</>
);
};
```

```
export default ListingDetails;


const styles = StyleSheet.create({
 container: {
  flex: 1,
  backgroundColor: Colors.white,
 },
 image: {
  width: width,
  height: IMG_HEIGHT,
 },
 contentWrapper: {
  padding: 20,
  backgroundColor: Colors.white,
 },
 listingName: {
  fontSize: 24,
  fontWeight: "500",
  color: Colors.black,
  letterSpacing: 0.5,
 },
 listingLocationWrapper: {
  flexDirection: "row",
  marginTop: 5,
  marginBottom: 10,
  alignItems: "center",
 },
```

```
listingLocationTxt: {
  fontSize: 14,
  marginLeft: 5,
  color: Colors.black,
},
highlightWrapper: {
  flexDirection: "row",
  marginVertical: 20,
  justifyContent: "space-between",
},
highlightIcon: {
  backgroundColor: "#F4F4F4",
  paddingHorizontal: 8,
  paddingVertical: 5,
  borderRadius: 8,
  marginRight: 5,
  alignItems: "center",
},
highlightTxt: {
  fontSize: 12,
  color: "#999",
},
highlightTxtVal: {
  fontSize: 14,
  fontWeight: "600",
},
listingDetails: {
  fontSize: 16,
```

```
  color: Colors.black,
  lineHeight: 25,
  letterSpacing: 0.5,
 },
 footer: {
  flexDirection: "row",
  position: "absolute",
  bottom: 0,
  padding: 20,
  paddingBottom: 30,
  width: width,
 },
 footerBtn: {
  flex: 1,
  backgroundColor: Colors.black,
  padding: 20,
  borderRadius: 10,
  alignItems: "center",
 },
 footerBookBtn: {
  flex: 2,
  backgroundColor: Colors.primaryColor,
  marginRight: 20,
 },
 footerBtnTxt: {
  color: Colors.white,
  fontSize: 16,
  fontWeight: "600",
```

```
  textTransform: "uppercase",
 },
});
```

Agora saindo da pasta app vamos para a pasta components. Nela, exclua todos os arquivos existentes e em seguida vamos crias os nosso componentes.

Components/CategoryButtons.tsx

```
import {
 ScrollView,
 StyleSheet,
 Text,
 TouchableOpacity,
 View,
} from "react-native";
import React, { useRef, useState } from "react";
import Colors from "@/constants/Colors";
import destinationCategories from "@/data/categories";
import { MaterialCommunityIcons } from "@expo/vector-icons";


type Props = {
 onCagtegoryChanged: (category: string) => void;
};


const CategoryButtons = ({ onCagtegoryChanged }: Props) => {
 const scrollRef = useRef<ScrollView>(null);
 const itemRef = useRef<TouchableOpacity[] | null[]>([]);
 const [activeIndex, setActiveIndex] = useState(0);
```

```
const handleSelectCategory = (index: number) => {
  const selected = itemRef.current[index];
  setActiveIndex(index);

  selected?.measure((x) => {
    scrollRef.current?.scrollTo({ x: x, y: 0, animated: true });
  });

  onCagtegoryChanged(destinationCategories[index].title);
};

return (
  <View>
    <Text style={styles.title}>Categorias</Text>
    <ScrollView
      ref={scrollRef}
      horizontal
      showsHorizontalScrollIndicator={false}
      contentContainerStyle={{
        gap: 20,
        paddingVertical: 10,
        marginBottom: 10,
      }}
    >
      {destinationCategories.map((item, index) => (
        <TouchableOpacity
          key={index}
          ref={(el) => (itemRef.current[index] = el)}
```

```
        onPress={() => handleSelectCategory(index)}
        style={
         activeIndex === index
          ? styles.categoryBtnActive
          : styles.categoryBtn
        }
       >
        <MaterialCommunityIcons
         name={item.iconName as any}
         size={20}
         color={activeIndex === index ? Colors.white : Colors.black}
        />
        <Text
         style={
          activeIndex === index
           ? styles.categoryBtnTxtActive
           : styles.categoryBtnTxt
         }
        >
         {item.title}
        </Text>
       </TouchableOpacity>
      ))}
    </ScrollView>
   </View>
  );
};
```

```
export default CategoryButtons;


const styles = StyleSheet.create({
 title: {
  fontSize: 22,
  fontWeight: "700",
  color: Colors.black,
 },
 categoryBtn: {
  flexDirection: "row",
  alignItems: "center",
  backgroundColor: Colors.white,
  paddingHorizontal: 16,
  paddingVertical: 10,
  borderRadius: 10,
  shadowColor: "#333333",
  shadowOffset: { width: 1, height: 2 },
  shadowOpacity: 0.1,
  shadowRadius: 3,
 },
 categoryBtnActive: {
  flexDirection: "row",
  alignItems: "center",
  backgroundColor: Colors.primaryColor,
  paddingHorizontal: 16,
  paddingVertical: 10,
  borderRadius: 10,
  shadowColor: "#333333",
```

```
    shadowOffset: { width: 1, height: 2 },
    shadowOpacity: 0.1,
    shadowRadius: 3,
  },
  categoryBtnTxt: {
    marginLeft: 5,
    color: Colors.black,
  },
  categoryBtnTxtActive: {
    marginLeft: 5,
    color: Colors.white,
  },
});
```

Components/GroupListings.tsx

```
import {
  FlatList,
  Image,
  ListRenderItem,
  StyleSheet,
  Text,
  View,
} from "react-native";
import React from "react";
import { GroupType } from "@/types/groupType";
import Colors from "@/constants/Colors";
import { Ionicons } from "@expo/vector-icons";
```

```
const GroupListings = ({ listings }: { listings: GroupType[] }) => {
 const renderItem: ListRenderItem<GroupType> = ({ item }) => {
  return (
   <View style={styles.item}>
    <Image source={{ uri: item.image }} style={styles.image} />
    <View>
     <Text style={styles.itemTxt}>{item.name}</Text>
     <View style={{ flexDirection: "row", alignItems: "center" }}>
      <Ionicons name="star" size={20} color={Colors.primaryColor} />
      <Text style={styles.itemRating}>{item.rating}</Text>
      <Text style={styles.itemReviews}>({item.reviews})</Text>
     </View>
    </View>
   </View>
  );
 };

 return (
  <View style={{ marginVertical: 20 }}>
   <Text style={styles.title}>Principais grupos de viagens</Text>
   <FlatList
    data={listings}
    renderItem={renderItem}
    horizontal
    showsHorizontalScrollIndicator={false}
   />
  </View>
```

```
  );
};

export default GroupListings;

const styles = StyleSheet.create({
  title: {
    fontSize: 22,
    fontWeight: "600",
    color: Colors.black,
    marginBottom: 10,
  },
  item: {
    backgroundColor: Colors.white,
    padding: 10,
    borderRadius: 10,
    marginRight: 20,
    flexDirection: "row",
    alignItems: "center",
  },
  image: {
    width: 80,
    height: 100,
    borderRadius: 10,
    marginRight: 10,
  },
  itemTxt: {
    fontSize: 14,
```

```
    fontWeight: "600",
    color: Colors.black,
    marginBottom: 8,
  },
  itemRating: {
    fontSize: 14,
    fontWeight: "600",
    color: Colors.black,
    marginLeft: 5,
  },
  itemReviews: {
    fontSize: 14,
    color: "#999",
  },
});
```

Components/Listings.tx

```
import {
  FlatList,
  Image,
  ListRenderItem,
  StyleSheet,
  Text,
  TouchableOpacity,
  View,
} from "react-native";
import React, { useEffect, useState } from "react";
```

```
import { ListingType } from "@/types/listingType";
import Colors from "@/constants/Colors";
import { FontAwesome5, Ionicons } from "@expo/vector-icons";
import { Link } from "expo-router";

type Props = {
  listings: any[];
  category: string;
};

const Listings = ({ listings, category }: Props) => {
  const [loading, setLoading] = useState(false);

  useEffect(() => {
    console.log("Update Listing");
    setLoading(true);

    setTimeout(() => {
      setLoading(false);
    }, 200);
  }, [category]);

  const renderItems: ListRenderItem<ListingType> = ({ item }) => {
    return (
      <Link href={`/listing/${item.id}`} asChild>
        <TouchableOpacity>
          <View style={styles.item}>
            <Image source={{ uri: item.image }} style={styles.image} />
```

```
        <View style={styles.bookmark}>
         <Ionicons
          name="bookmark-outline"
          size={20}
          color={Colors.white}
         />
        </View>
        <Text style={styles.itemTxt} numberOfLines={1} ellipsizeMode="tail">
         {item.name}
        </Text>
        <View
         style={{ flexDirection: "row", justifyContent: "space-between" }}
        >
         <View style={{ flexDirection: "row", alignItems: "center" }}>
          <FontAwesome5
           name="map-marker-alt"
           size={18}
           color={Colors.primaryColor}
          />
          <Text style={styles.itemLocationTxt}>{item.location}</Text>
         </View>
         <Text style={styles.itemPriceTxt}>R${item.price}</Text>
        </View>
       </View>
     </TouchableOpacity>
   </Link>
  );
};
```

```
  return (
    <View>
      <FlatList
        data={loading ? [] : listings}
        renderItem={renderItems}
        horizontal
        showsHorizontalScrollIndicator={false}
      />
    </View>
  );
};

export default Listings;

const styles = StyleSheet.create({
  item: {
    backgroundColor: Colors.white,
    padding: 10,
    borderRadius: 10,
    marginRight: 20,
    width: 220,
  },
  image: {
    width: 200,
    height: 200,
    borderRadius: 10,
    marginBottom: 30,
```

```
  },
  bookmark: {
    position: "absolute",
    top: 185,
    right: 30,
    backgroundColor: Colors.primaryColor,
    padding: 10,
    borderRadius: 30,
    borderWidth: 2,
    borderColor: Colors.white,
  },
  itemTxt: {
    fontSize: 16,
    fontWeight: "600",
    color: Colors.black,
    marginBottom: 10,
  },
  itemLocationTxt: {
    fontSize: 12,
    marginLeft: 5,
  },
  itemPriceTxt: {
    fontSize: 12,
    fontWeight: "600",
    color: Colors.primaryColor,
  },
});
```

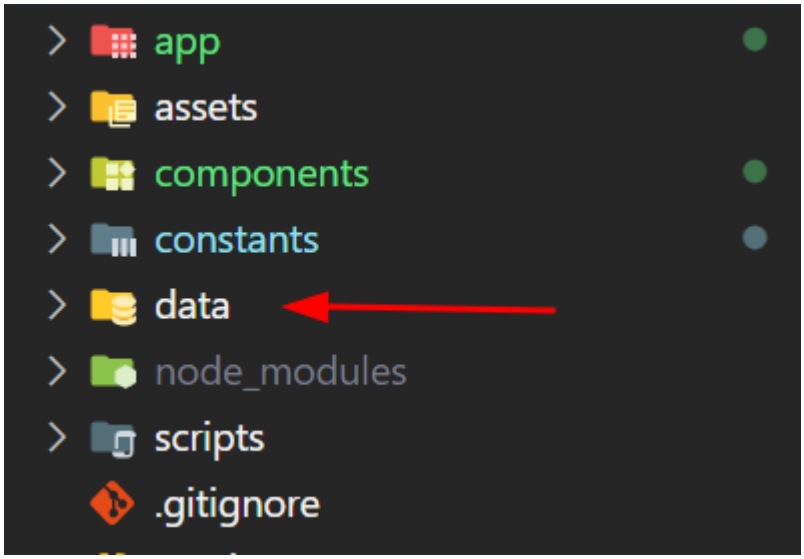
Finalizando os components, vamos alterar o arquivo Colors.ts

Constants/Colors.ts

```
export default {
 primaryColor: '#ff7f36',
 bgColor: '#F4F4F4',
 black: '#27283a',
 white: '#FFFFFF',
}
```

A pasta Hooks não será utilizada, então podemos apagar.

Agora na raiz do projeto crie a pasta **"data"**. Nela iremos armazenar as informações que serão passadas para as nossas views, simulando informações que viriam do backend.



Nela crie o arquivo categories.ts

data/categories.ts

```
const destinationCategories = [
 {
   title: "All",
```

```
  iconName: "hiking",
},
{
  title: "Praias",
  iconName: "beach",
},
{
  title: "Montanhas",
  iconName: "terrain",
},
{
  title: "Cidades",
  iconName: "city",
},
{
  title: "Florestas",
  iconName: "tree",
},
{
  title: "Lagos",
  iconName: "swim",
},
{
  title: "Locais Históricos",
  iconName: "castle",
},
{
  title: "Parques Nacionais",
```

```
  iconName: "pine-tree",
 },
 {
  title: "Ilhas",
  iconName: "island",
 },
 {
  title: "Desertos",
  iconName: "weather-sunny",
 },
];

export default destinationCategories;
```

Em seguida, crie o arquivo groups.json.

data/ groups**.json**

```
[
 {
  "id": "1",
  "name": "Agência de viagens de férias",
  "image": "https://images.unsplash.com/photo-1516496636080-
14fb876e029d?q=80&w=3388&auto=format&fit=crop&ixlib=rb-
4.0.3&ixid=M3wxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVufDB8fHx8fA%3D%3D",
  "rating": 4.7,
  "reviews": 1450
 },
 {
  "id": "2",
```

```
  "name": "Planejadores de turismo felizes",
  "image": "https://images.unsplash.com/photo-1618591362251-1c9f70b59192?q=80&w=3177&auto=format&fit=crop&ixlib=rb-4.0.3&ixid=M3wxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVufDB8fHx8fA%3D%3D",
  "rating": 4.8,
  "reviews": 650
 },
 {
  "id": "3",
  "name": "Passeios e viagens reais",
  "image": "https://images.unsplash.com/photo-1600468636011-c75ae69b7fcb?q=80&w=2625&auto=format&fit=crop&ixlib=rb-4.0.3&ixid=M3wxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVufDB8fHx8fA%3D%3D",
  "rating": 4.9,
  "reviews": 550
 }
]
```

Posteriormente, na mesma pasta cole o arquivo **destinations.json disponibilizado na aula.**

Com isso temos a nossa base de dados pronta.

Agora, na raiz do projeto crie a pasta “**types”**.

types/groupType.ts

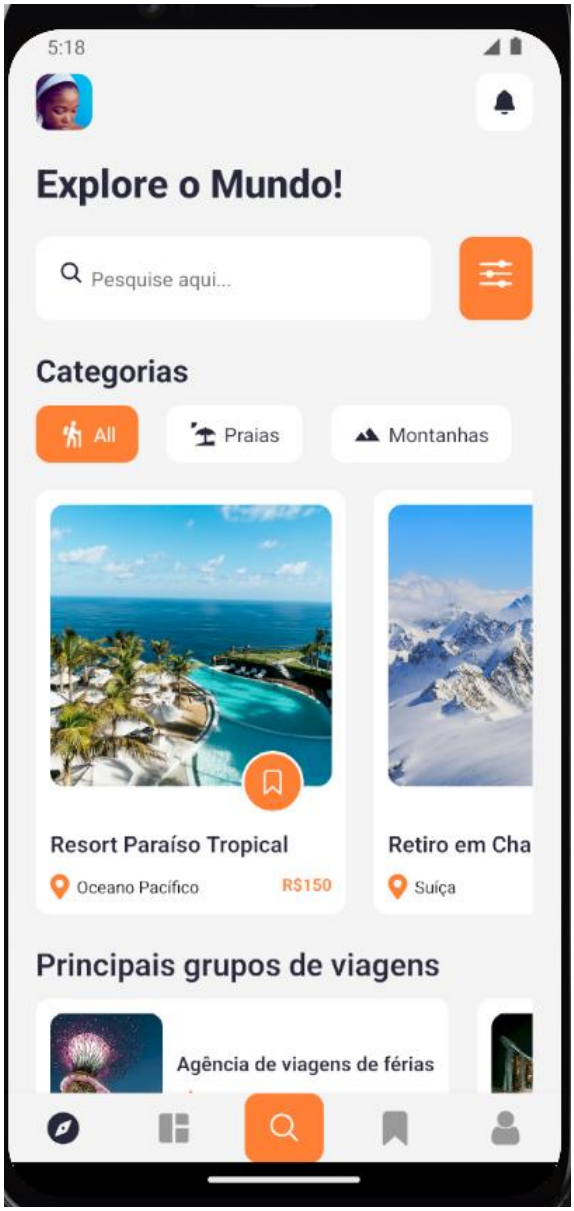
```
export interface GroupType {
 id: string;
 name: string;
 image: string;
 rating: number;
```

```
  reviews: number;
}
```

types/listingType.ts

```
export interface ListingType {
  id: string;
  name: string;
  image: string;
  description: string;
  rating: number;
  price: string;
  duration: string;
  location: string;
  category: string;
}
```

Parabéns!!! Com isso o nosso projeto está pronto. Agora no terminal digite “npm start” para iniciar a aplicação e ver se está funcionando.


5:18
Explore o Mundo!
Pesquise aqui...
Categorias
All
Praias
Montanhas
Resort Paraíso Tropical
Oceano Pacífico
R$150
Retiro em Cha
Suíça
Principais grupos de viagens
Agência de viagens de férias


5:18
Resort Paraíso Tropical
Oceano Pacífico
Duração
3 Dias
Pessoas
3
Avaliação
4.8
Fuja para o melhor retiro paradisíaco tropical com praias imaculadas, vegetação exuberante e acomodações luxuosas.
RESERVE AGORA
R$150