Alexandre Gomes

Professor Universitário | Pesquisador CSIM | JavaScript | Node.JS | ReactJS e React-Native | Flutter/Dart | Java | FullStack Developer | Segurança da Informação | Profissional de T.I. +27 anos

Franca, São Paulo, Brasil ·

Fatec Franca - Faculdade de Tecnologia "Dr. Thomaz Novelino"

Universidade de São Paulo

Minicurso Aplicação Mobile Android com Geolocalização (React Native)

- SITE PARA A DOCUMENTAÇÃO E PREPARAÇÃO DO AMBIENTE DE DESENVOLVIMENTO

  https://reactnative.dev/docs/environment-setup

- ARQUIVO PARA CONFIGURAÇÃO DO AMBIENTE REACT NATIVE NO S.O. WINDOWS

  Configurando o ambiente de desenvo => Arquivo disponível na pasta do meu GitHub

Após toda preparação do ambiente, podemos criar o nosso projeto.

> Abra o CMD e digite `cd\` para ficarmos na raiz do diretório c:\>

Digite os comandos a seguir:

```
npx react-native@latest init AppAndroidGeolocal
```

```
cd AppAndroidGeolocal
```

```
npx react-native run-android
```

VAMOS PARA NOSSO PROJETO: "**AppAndroidGeolocal**"

Abrir a pasta do aplicativo no Visual Studio Code (VSCode)

Você pode usar React Native sem TypeScript:

1. Remova a dependência `typescript` do arquivo `package.json`.
2. Remova o arquivo `tsconfig.json` do projeto.
3. Apagar o arquivo `App.tsx`

Vamos trabalhar com rotas, portanto vamos instalar a biblioteca **react-navigation**

```
npm install @react-navigation/native
```

```
npm install react-native-screens react-native-safe-area-context
```

```
npm install @react-navigation/stack
```

```
npm install react-native-gesture-handler
```

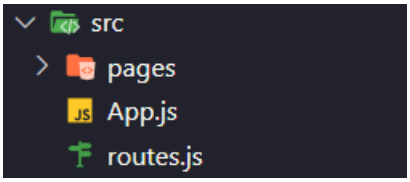
Adicione o código abaixo na classe `MainActivity` da pasta **android\app\src\main\java\com\appandroidgeolocal\MainActivity.java**

Importação na parte superior deste arquivo abaixo de sua declaração de pacote:

```
import android.os.Bundle;
```

```
public class MainActivity extends ReactActivity {
  // ...

  @Override
  protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(null);
  }

  // ...
}
```

Criar uma pasta <src> na raiz do projeto e dentro dela os arquivos <App.js> e <routes.js> e uma pasta <pages>



Dentro da pasta <pages> criar um arquivo <main.js>

No arquivo <main.js> vamos criar um componente (por enquanto uma página em branco)

```
import React, { Component } from 'react';
import { View, Text } from 'react-native';

export default class Main extends Component {
    render() {
        return (
            <View>
                <Text>Aqui é o corpo da Main</Text>
            </View>
        );
    }
}
```

No arquivo <routes.js>

```
import React from 'react';
import {NavigationContainer} from '@react-navigation/native';
import {createStackNavigator} from '@react-navigation/stack';
import {StatusBar} from 'react-native';
import Main from './pages/main';

const Stack = createStackNavigator();

export default function Routes() {
  return (
    <NavigationContainer>
      <StatusBar backgroundColor="#38A69D" barStyle="light-content" />
      <Stack.Navigator>
        <Stack.Screen
          name="Aqui é o nome que aparece no Header"
          component={Main}
          options={{headerShown: false}}
        />
      </Stack.Navigator>
    </NavigationContainer>
  );
}
```

No <App.js> da pasta <src>

```
import React from "react";
import Routes from "./routes";

const App = () => <Routes/>

export default App;
```

No arquivo <index.js> principal da raiz alterar o caminho do App

```
import App from './App';
```

para

```
import App from './src/App';
```

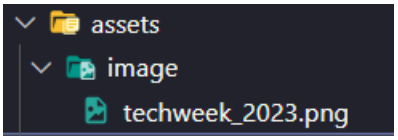
Meu arquivo <index.js> da raiz principal fica assim:

```
import {AppRegistry} from 'react-native';
import App from './src/App';
import {name as appName} from './app.json';

AppRegistry.registerComponent(appName, () => App);
```

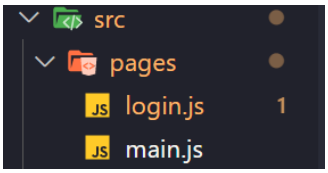
Saia da aplicação e rode o `npx react-native run-android` novamente.

Dentro da pasta <src> criar uma pasta <assets> e dentro, uma outra pasta <image>

Copiar a imagem **techweek_2023.png** para dentro dessa pasta <image>



Dentro da pasta <pages> criar um arquivo <login.js>

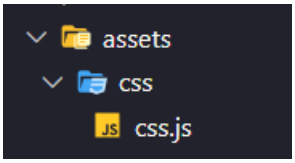


No arquivo <login.js>

```
import React, {useState} from 'react';
import {Text, View, Image, TextInput, TouchableOpacity} from 'react-native';
import {styles} from '../assets/css/css';

const Login = () => {

  const [email, setEmail] = useState('');
  const [password, setPassword] = useState('');

  const handleLogin = () => {
    // Aqui você pode implementar a lógica de login
    console.log(`Email: ${email}, Password: ${password}`);
  };

  return (
    <View style={styles.container}>
      <Image
```

```
        style={styles.image}
        source={require('../assets/image/techweek_2023.png')}
      />
      <TextInput
        style={styles.input}
        placeholder="E-mail"
        value={email}
        onChangeText={setEmail}
      />
      <TextInput
        style={styles.input}
        placeholder="Senha"
        secureTextEntry={true}
        value={password}
        onChangeText={setPassword}
      />
      <TouchableOpacity style={styles.button} onPress={handleLogin}>
        <Text style={styles.buttonText}>Entrar</Text>
      </TouchableOpacity>
    </View>
  );
};

export default Login;
```

Dentro da pasta <assets> criar uma pasta <css> e dentro, um arquivo <css.js>



No arquivo <css.js>

```
import {StyleSheet} from 'react-native';

export const styles = StyleSheet.create({
  container: {
    flex: 1,
    alignItems: 'center',
    justifyContent: 'center',
    backgroundColor: '#fff',
  },
  containerMaps: {
    flex: 1,
    backgroundColor: '#fff',
    justifyContent: 'center',
  },
  input: {
    borderWidth: 1,
    borderColor: '#ccc',
    borderRadius: 5,
```

```
    padding: 10,
    marginVertical: 10,
    width: '80%',
  },
  image:{
    width: '100%',
    resizeMode: 'contain',
  },
  imageMaps: {
    width: 110,
    resizeMode: 'contain',
    left: '70%',
    bottom: '20%',
  },
  button: {
    backgroundColor: '#3498db',
    borderRadius: 5,
    marginVertical: 5,
    alignItems: 'center',
    justifyContent: 'center',
    width: '80%',
    padding: 10,
  },
  buttonText: {
    color: '#fff',
    fontWeight: 'bold',
  },
  boldText: {
    fontSize: 23,
    fontWeight: 'bold',
    color: 'red',
  },
  text: {
    fontSize: 19,
    color: 'black',
    padding: 5,
  },
});
```

Agora vamos chamar o “Login”:

No arquivo <routes.js>

```
import React from 'react';
import {NavigationContainer} from '@react-navigation/native';
import {createStackNavigator} from '@react-navigation/stack';
import {StatusBar} from 'react-native';
import Main from './pages/main';
import Login from './pages/login'

const Stack = createStackNavigator();

export default function Routes() {
```

```
  return (
    <NavigationContainer>
      <StatusBar backgroundColor="#38A69D" barStyle="light-content" />
      <Stack.Navigator>
        <Stack.Screen
          name="login"
          component={Login}
          options={{headerShown: false}}
        />
        <Stack.Screen
          name="main"
          component={Main}
          options={{headerShown: false}}
        />
      </Stack.Navigator>
    </NavigationContainer>
  );
}
```

Testar e verificar no METRO

VAMOS SIMULAR O BOTÃO “ENTRAR” DO LOGIN PARA A “PAGE” MAIN

No arquivo <login.js>

```
import { useNavigation } from '@react-navigation/native';
```

```
const Login = () => {
  const [email, setEmail] = useState('');
  const [password, setPassword] = useState('');

  const navigation = useNavigation();

  const handleLogin = () => {
    if (email === 'a@a.com.br' && password === '123') {
      navigation.navigate('main');
    } else {
      alert('E-mail ou senha inválidos!');
    }
  };
```

Testar e verificar

Para obter a geolocalização, vamos utilizar a biblioteca “**@react-native-community/geolocation**”. É possível utilizá-la tanto no Android quanto no iOS, foi testada em vários projetos no laboratório e oferece fácil implementação.

https://www.npmjs.com/package/@react-native-community/geolocation

```
npm install @react-native-community/geolocation
```

Vamos adicionar as seguintes linhas no arquivo **AndroidManifest.xml**, que vai informar ao Android que em alguma parte de nosso código poderemos solicitar permissões referentes a localização do usuário. Dentro da tag **<manifest** insira:

Caminho>>> /* android/app/src/main/AndroidManifest.xml */

```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_BACKGROUND_LOCATION" />
```

Para solicitar as permissões, vamos instalar a biblioteca "**react-native-permissions**". Ela enviará a mensagem solicitando as permissões ao usuário e também vai nos informar do estado das permissões necessárias.

https://www.npmjs.com/package/react-native-permissions

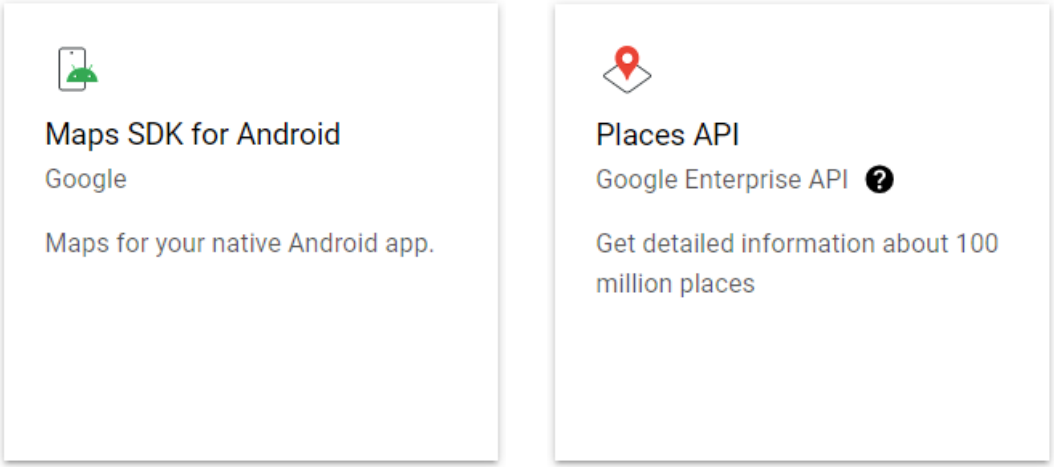
```
npm install react-native-permissions
```

Utilizando um MAPA na nossa aplicação

Após obter a geolocalização do usuário, que tal mostrarmos onde ele está localizado em um mapa? Podemos também mostrar locais próximos, ou até mesmo traçar rotas para possíveis locais desejados. Para isso, podemos utilizar diferentes **API's disponíveis, como MapBox, Google Maps, Open Street Maps e algumas outras**. Muitos destes serviços são gratuitos até um certo número de solicitações, cabendo a você selecionar o melhor para seu projeto. Para o nosso exemplo, vamos utilizar o **Google Maps API**.

Por ser um serviço restrito deveremos **criar uma conta** para nossa aplicação dentro da **'Google Maps Platform'.** Basta seguir este link <https://developers.google.com/maps/gmp-get-started?hl=pt> e então cadastrar seu app na plataforma.

Depois de cadastrado, você deverá adicionar ao seu projeto a biblioteca '**Maps SDK for Android'** e **'PLACES API'**. Seguindo todos os passos disponibilizados neste link < https://developers.google.com/maps/documentation/android-sdk/config?hl=pt-br>, você terá disponível as **API KEY's** necessárias para inserir o mapa e novas localizações em seu projeto.



Devemos instalar a dependência "**react-native-google-places-autocomplete**" que é um componente personalizável de preenchimento automático do Google Places para aplicativos React-Native iOS e Android.

https://www.npmjs.com/package/react-native-google-places-autocomplete

```
npm install react-native-google-places-autocomplete
```

No arquivo **android/app/src/main/AndroidManifest.xml**

```
      </activity>
      <meta-data
        android:name="com.google.android.geo.API_KEY"
        android:value="PASSAR_A_KEY_AQUI"/>
    </application>
```

Para renderizar o mapa, vamos utilizar a biblioteca '**react-native-maps'**. Seguindo esse link < https://github.com/react-native-maps/react-native-maps/blob/master/docs/installation.md> você verá como instalar.

```
npm install react-native-maps
```

VAMOS TRABALHAR NA NOSSA PAGE "**MAIN**" modificando para **function**. Ela será a página que fará a permissão para usar o GPS, terá um botão de localização atual e visualização no mapa.

No arquivo <main.js>

```
import React from 'react';
import { View } from 'react-native';
import { styles } from '../assets/css/css'

export default function Main(){
    return (
        <View >
            <View style={styles.map}>
```

```
            </View>

            <View style={styles.search}>

            </View>
        </View>
    )
}
```

No arquivo <css.js> acrescentar

```
  map: {
    height: '70%',
  },
  search: {
    height: '30%',
    backgroundColor: '#ccc',
    padding: 10,
  },
```

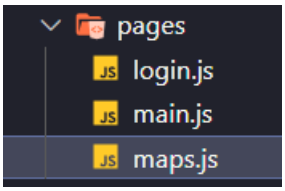
Testar e verificar

Voltando no arquivo <main.js> criar as permissões para o GPS.

```
import React, {useState} from 'react';
import {Text, View, Image, PermissionsAndroid, TouchableOpacity, Platform} from 'react-
native';
import Geolocation from '@react-native-community/geolocation';
import {useNavigation} from '@react-navigation/native';
import {styles} from '../assets/css/css';

export default function Main() {
  const [currentLatitude, setCurrentLatitude] = useState('');
  const [currentLongitude, setCurrentLongitude] = useState('');

  const callLocation = () => {
    if (Platform.OS === 'ios') {
      getLocation();
    } else {
      const requestLocationPermission = async () => {
        const granted = await PermissionsAndroid.request(
          PermissionsAndroid.PERMISSIONS.ACCESS_FINE_LOCATION,
          {
            title: 'Permissão de Acesso à Localização',
            message: 'Este aplicativo precisa acessar sua localização.',
            buttonNeutral: 'Pergunte-me depois',
            buttonNegative: 'Cancelar',
            buttonPositive: 'OK',
          },
        );
        if (granted === PermissionsAndroid.RESULTS.GRANTED) {
          getLocation();
        } else {
          alert('Permissão de Localização negada');
```

```
        }
      };
      requestLocationPermission();
    }
  };

  const getLocation = () => {
    Geolocation.getCurrentPosition(
      position => {
        const currentLatitude = JSON.stringify(position.coords.latitude);
        const currentLongitude = JSON.stringify(position.coords.longitude);
        setCurrentLatitude(currentLatitude);
        setCurrentLongitude(currentLongitude);
      },
      error => alert(error.message),
      {enableHighAccuracy: true, timeout: 20000, maximumAge: 1000},
    );
  };

  const navigation = useNavigation();

  const viweMaps = () => {
    if (currentLatitude === '' || currentLongitude === '') {
      alert('Não foi possível obter a localização atual');
      return;
    } else {
      navigation.navigate('maps', {
        lat: currentLatitude,
        long: currentLongitude,
      });
    }
  };

  return (
    <View style={styles.container}>
      <Image
        style={styles.image}
        source={require('../assets/image/maps.png')}
      />
      <Text style={styles.boldText}>Sua localização atual</Text>
      <Text style={styles.text}>Latitude: {currentLatitude}</Text>
      <Text style={styles.text}>Longitude: {currentLongitude}</Text>
      <TouchableOpacity style={styles.button} onPress={callLocation}>
        <Text style={styles.buttonText}>Obter Localização Atual</Text>
      </TouchableOpacity>
      <TouchableOpacity style={styles.button} onPress={viweMaps}>
        <Text style={styles.buttonText}>Ver Localização no Google Maps</Text>
      </TouchableOpacity>
    </View>
  );
}
```

Dentro da pasta <pages> criar um arquivo <maps.js>



Agora vamos chamar o “maps”:

No arquivo <routes.js>

```
import React from 'react';
import {NavigationContainer} from '@react-navigation/native';
import {createStackNavigator} from '@react-navigation/stack';
import {StatusBar} from 'react-native';
import Main from './pages/main';
import Login from './pages/login';
import Maps from './pages/maps';

const Stack = createStackNavigator();

export default function Routes() {
  return (
    <NavigationContainer>
      <StatusBar backgroundColor="#38A69D" barStyle="light-content" />
      <Stack.Navigator>
        <Stack.Screen
          name="login"
          component={Login}
          options={{headerShown: false}}
        />
        <Stack.Screen
          name="main"
          component={Main}
          options={{headerShown: false}}
        />
        <Stack.Screen
          name="maps"
          component={Maps}
          options={{headerShown: false}}
        />
      </Stack.Navigator>
    </NavigationContainer>
  );
}
```

No arquivo <maps.js> vamos testar a rota para o botão do MAIN “**Ver Localização no Google Maps**”

```
import React, { Component } from 'react';
import { View, Text } from 'react-native';

export default class Maps extends Component {
    render() {
        return (
            <View>
                <Text>Aqui é o corpo do Maps</Text>
            </View>
        );
    }
}
```

Testar o botão do MAIN “**Ver Localização no Google Maps**”

Ainda no arquivo <maps.js> vamos modificar a classe para os Hooks

```
import React, {useEffect, useState} from 'react';
import MapView, {Marker} from 'react-native-maps';
import {styles} from '../assets/css/css';
import {View, Text, Image} from 'react-native';
import {GooglePlacesAutocomplete} from 'react-native-google-places-autocomplete';

const Maps = ({route}) => {
  const {lat, long} = route.params;
  const [latitude, setLatitude] = useState(parseFloat(lat));
  const [longitude, setLongitude] = useState(parseFloat(long));
  const [destinoLat, setDestinoLat] = useState(null);
  const [destinoLon, setDestinoLon] = useState(null);

  useEffect(() => {
    if (destinoLat !== null && destinoLon !== null) {
      setLatitude(destinoLat);
      setLongitude(destinoLon);
    }
  }, [destinoLat, destinoLon]);

  console.log('lat', latitude);
  console.log('long', longitude);

  return (
    <View style={styles.containerMaps}>
      <MapView
        style={styles.map}
        showsUserLocation={true}
        showsMyLocationButton={false}
        toolbarEnabled={false}
        region={{
          latitude,
          longitude,
```

```
          latitudeDelta: 0.007,
          longitudeDelta: 0.007,
        }}>
        <Marker
          coordinate={{
            latitude,
            longitude,
          }}
          title="Meu local"
          description="Localização atual"
        />
      </MapView>
      <View style={styles.search}>
        <Text style={styles.text}>Digite sua nova localização:</Text>
        <GooglePlacesAutocomplete
          placeholder="Para onde vamos?"
          onPress={(data, details = null) => {
            setDestinoLat(details.geometry.location.lat);
            setDestinoLon(details.geometry.location.lng);
          }}
          query={{
            key: 'PASSR_A_KEY_PLACE_API_AQUI',
            language: 'pt-br',
          }}
          enablePoweredByContainer={false}
          fetchDetails={true}
          styles={{listView: {height: 100}}}
        />
        <Image
          style={styles.imageMaps}
          source={require('../assets/image/google-maps-vetor.png')}
        />
      </View>
    </View>
  );
};

export default Maps;
```

"Use este passo a passo com muito café."

Prof. Esp. Alexandre Gomes da Silva

linkedin.com/in/alexandre-gomes-218985118

Currículo Lattes http://lattes.cnpq.br/6386688512462449

https://github.com/XandyGomes

alexandre.silva251@fatec.sp.gov.br

(16) 99201-1010