

src/composables/useGeolocation.js

```
1 import { ref } from 'vue';
2
3 // --- Configurações ---
4 // Coordenadas aproximadas do ponto de referência no Campus UFC Russas (Ex: Centro)
5 // !!! SUBSTITUIR PELAS COORDENADAS REAIS DO SEU PONTO DE REFERÊNCIA !!!
6 const campusReferenceCoordinates = {
7   latitude: -4.9438, // Exemplo - Latitude do ponto de referência (0,0 no mapa %)
8   longitude: -37.9553, // Exemplo - Longitude do ponto de referência (0,0 no mapa %)
9 };
10
11 // Dimensões em metros (aproximadas) cobertas pelo mapa a partir do ponto de referência
12 // !!! AJUSTAR COM BASE NO SEU MAPA E PONTO DE REFERÊNCIA !!!
13 // Se o ponto de referência é (0,0), estas são as distâncias máximas X e Y visíveis no mapa.
14 // Se o ponto de referência é o centro (50,50), ajuste a lógica de conversão.
15 // Vamos assumir que (0,0) no mapa % corresponde a campusReferenceCoordinates
16 const campusDimensions = {
17   widthMeters: 1500, // Largura total coberta pelo mapa em metros
18   heightMeters: 1200, // Altura total coberta pelo mapa em metros
19 };
20 // --- Fim Configurações ---
21
22
23 export function useGeolocation() {
24   const userPosition = ref(null); // Posição no mapa { x: number, y: number }
25   const geolocationError = ref(null);
26   const isGettingLocation = ref(false);
27
28   /**
29    * Converte coordenadas geográficas (lat/lng) para coordenadas percentuais no mapa.
30    * Assume que campusReferenceCoordinates corresponde ao canto superior esquerdo (0%, 0%) do mapa.
31    */
32   const convertGeoToMapCoordinates = (lat, lng) => {
33     // Fatores de conversão (aproximados)
34     const metersPerLatDegree = 111320;
```

```

35  const metersPerLngDegree = 111320 * Math.cos((campusReferenceCoordinates.latitude * Math.PI) / 180);
36
37  // Calcula a diferença em metros da referência
38  const latDiff = lat - campusReferenceCoordinates.latitude;
39  const lngDiff = lng - campusReferenceCoordinates.longitude;
40
41  // Distância em metros (Y cresce para baixo em geo, para cima em lat)
42  const distanceY = -latDiff * metersPerLatDegree; // Invertido
43  const distanceX = lngDiff * metersPerLngDegree;
44
45  // Converte para porcentagem (0-100)
46  let x = (distanceX / campusDimensions.widthMeters) * 100;
47  let y = (distanceY / campusDimensions.heightMeters) * 100;
48
49  // Limita os valores entre 0 e 100%
50  x = Math.max(0, Math.min(100, x));
51  y = Math.max(0, Math.min(100, y));
52
53  console.log(`Geo (${lat.toFixed(5)}, ${lng.toFixed(5)}) -> Map (${x.toFixed(2)}%, ${y.toFixed(2)}%)`);
54  return { x, y };
55 };
56
57 /**
58  * Tenta obter a localização geográfica do usuário e convertê-la para o mapa.
59  */
60 const getUserLocationAuto = () => {
61   if (!navigator.geolocation) {
62     geolocationError.value = "Geolocalização não é suportada por este navegador.";
63     userPosition.value = null; // Garante que não há posição antiga
64     return;
65   }
66
67   isGettingLocation.value = true;
68   geolocationError.value = "Obtendo sua localização...";
69   userPosition.value = null; // Limpa posição anterior enquanto busca
70
71   navigator.geolocation.getCurrentPosition(

```

```

72 (position) => {
73     const userLat = position.coords.latitude;
74     const userLng = position.coords.longitude;
75     console.log(`Localização geográfica recebida: ${userLat}, ${userLng}`);
76
77     userPosition.value = convertGeoToMapCoordinates(userLat, userLng);
78     geolocationError.value = null; // Limpa erro/mensagem
79     isGettingLocation.value = false;
80 },
81 (err) => {
82     console.error("Erro ao obter geolocalização:", err);
83     switch (err.code) {
84         case err.PERMISSION_DENIED:
85             geolocationError.value = "Permissão de localização negada.";
86             break;
87         case err.POSITION_UNAVAILABLE:
88             geolocationError.value = "Informação de localização indisponível.";
89             break;
90         case err.TIMEOUT:
91             geolocationError.value = "Tempo esgotado ao buscar localização.";
92             break;
93         default:
94             geolocationError.value = "Erro desconhecido ao obter localização.";
95             break;
96     }
97     // Definir posição manual pode ser uma alternativa aqui
98     // error.value += " Você pode definir sua localização manualmente clicando no mapa.";
99     userPosition.value = null; // Garante que não usa posição inválida
100    isGettingLocation.value = false;
101 },
102 {
103     enableHighAccuracy: true, // Tenta obter a localização mais precisa
104     timeout: 10000, // Tempo máximo para obter (10 segundos)
105     maximumAge: 0, // Força obter uma nova localização (não usa cache)
106 }
107 );
108 };

```

```
109
110 /**
111  * Define a posição do usuário manualmente com base em um clique no mapa.
112  * @param {number} clickXPercent - Coordenada X do clique em porcentagem (0-100).
113  * @param {number} clickYPercent - Coordenada Y do clique em porcentagem (0-100).
114  */
115 const setUserLocationManually = (clickXPercent, clickYPercent) => {
116   const finalX = Math.max(0, Math.min(100, clickXPercent));
117   const finalY = Math.max(0, Math.min(100, clickYPercent));
118   userPosition.value = { x: finalX, y: finalY };
119   geolocationError.value = null; // Limpa qualquer erro anterior
120   console.log(`Localização definida manualmente: X: ${finalX.toFixed(2)}%, Y: ${finalY.toFixed(2)}%`);
121 }
122
123
124 return {
125   userPosition,          // ref({x, y} | null)
126   geolocationError,      // ref(string | null)
127   isGettingLocation,     // ref(boolean)
128   getUserLocationAuto,   // function
129   setUserLocationManually // function (xPerc, yPerc)
130 };
131 }
```