src/composables/useGeolocation.js

```
import { ref } from 'vue';

// --- Configurações ---
// Coordenadas aproximadas do ponto de referência no Campus UFC Russas (Ex: Centro)
// !!! SUBSTITUIR PELAS COORDENADAS REAIS DO SEU PONTO DE REFERÊNCIA !!!
const campusReferenceCoordinates = {
  latitude: -4.944444,  // Latitude do Campus UFC Russas
  longitude: -37.955556, // Longitude do Campus UFC Russas
};

// Dimensões em metros (aproximadas) cobertas pelo mapa a partir do ponto de referência
// !!! AJUSTAR COM BASE NO SEU MAPA E PONTO DE REFERÊNCIA !!!
// Se o ponto de referência é (0,0), estas são as distâncias máximas X e Y visíveis no mapa.
// Se o ponto de referência é o centro (50,50), ajuste a lógica de conversão.
// Vamos assumir que (0,0) no mapa % corresponde a campusReferenceCoordinates
const campusDimensions = {
  widthMeters: 500,  // Largura aproximada do campus em metros
  heightMeters: 400, // Altura aproximada do campus em metros
};
// --- Fim Configurações ---


export function useGeolocation() {
  const userPosition = ref(null); // Posição no mapa { x: number, y: number }
  const geolocationError = ref(null);
  const isGettingLocation = ref(false);

  /**
   * Converte coordenadas geográficas (lat/lng) para coordenadas percentuais no mapa.
   * Assume que campusReferenceCoordinates corresponde ao canto superior esquerdo (0%, 0%) do mapa.
   */
  const convertGeoToMapCoordinates = (lat, lng) => {
    // Fatores de conversão (aproximados)
    const metersPerLatDegree = 111320;
```

```
    const metersPerLngDegree = 111320 * Math.cos((campusReferenceCoordinates.latitude * Math.PI) / 180);

    // Calcula a diferença em metros da referência
    const latDiff = lat - campusReferenceCoordinates.latitude;
    const lngDiff = lng - campusReferenceCoordinates.longitude;

    // Distância em metros (Y cresce para baixo em geo, para cima em lat)
    const distanceY = -latDiff * metersPerLatDegree; // Invertido
    const distanceX = lngDiff * metersPerLngDegree;

    // Converte para porcentagem (0-100)
    let x = (distanceX / campusDimensions.widthMeters) * 100;
    let y = (distanceY / campusDimensions.heightMeters) * 100;

    // Limita os valores entre 0 e 100%
    x = Math.max(0, Math.min(100, x));
    y = Math.max(0, Math.min(100, y));

    console.log(`Geo (${lat.toFixed(5)}, ${lng.toFixed(5)}) -> Map (${x.toFixed(2)}%, ${y.toFixed(2)}%)`);
    return { x, y };
  };

  /**
   * Tenta obter a localização geográfica do usuário e convertê-la para o mapa.
   */
  const getUserLocationAuto = () => {
    if (!navigator.geolocation) {
      geolocationError.value = "Geolocalização não é suportada por este navegador.";
      userPosition.value = null; // Garante que não há posição antiga
      return;
    }

    isGettingLocation.value = true;
    geolocationError.value = "Obtendo sua localização...";
    userPosition.value = null; // Limpa posição anterior enquanto busca

    navigator.geolocation.getCurrentPosition(
```

```
      (position) => {
        const userLat = position.coords.latitude;
        const userLng = position.coords.longitude;
        console.log(`Localização geográfica recebida: ${userLat}, ${userLng}`);

        userPosition.value = convertGeoToMapCoordinates(userLat, userLng);
        geolocationError.value = null; // Limpa erro/mensagem
        isGettingLocation.value = false;
      },
      (err) => {
        console.error("Erro ao obter geolocalização:", err);
        switch (err.code) {
          case err.PERMISSION_DENIED:
            geolocationError.value = "Permissão de localização negada.";
            break;
          case err.POSITION_UNAVAILABLE:
            geolocationError.value = "Informação de localização indisponível.";
            break;
          case err.TIMEOUT:
            geolocationError.value = "Tempo esgotado ao buscar localização.";
            break;
          default:
            geolocationError.value = "Erro desconhecido ao obter localização.";
            break;
        }
        // Definir posição manual pode ser uma alternativa aqui
        // error.value += " Você pode definir sua localização manualmente clicando no mapa.";
        userPosition.value = null; // Garante que não usa posição inválida
        isGettingLocation.value = false;
      },
      {
        enableHighAccuracy: true, // Tenta obter a localização mais precisa
        timeout: 10000, // Tempo máximo para obter (10 segundos)
        maximumAge: 0, // Força obter uma nova localização (não usa cache)
      }
    );
  };
```

```
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 }
```