

src/composables/useMapInteraction.js

```
1 import { ref } from 'vue';
2
3 export function useMapInteraction(initialScale = 1, minScale = 0.5, maxScale = 3, zoomStep = 1.2) {
4   const mapScale = ref(initialScale);
5   const popupInfo = ref({ visible: false, x: 0, y: 0, text: "" });
6   const settingUserLocationMode = ref(false); // Novo estado para modo de definição manual
7
8   const zoomIn = () => {
9     mapScale.value = Math.min(mapScale.value * zoomStep, maxScale);
10  };
11
12  const zoomOut = () => {
13    mapScale.value = Math.max(mapScale.value / zoomStep, minScale);
14  };
15
16  const showPopup = (local) => {
17    if (local && typeof local.x === 'number' && typeof local.y === 'number') {
18      popupInfo.value = { visible: true, x: local.x, y: local.y, text: local.nome };
19    } else {
20      console.warn("Tentativa de mostrar popup para local inválido:", local);
21      popupInfo.value = { visible: false, x: 0, y: 0, text: "" };
22    }
23  };
24
25
26  const hidePopup = () => {
27    popupInfo.value.visible = false;
28  };
29
30  /**
31   * Processa um clique no mapa. Pode fechar menu, definir localização manual, etc.
32   * @param {MouseEvent} event - 0 evento de clique.
33   * @param {Object} mapRect - 0 BoundingClientRect do elemento .map-image.
34   * @param {Function} callbackSetLocation - Função para chamar ao definir localização manual.
```

```

35 * @param {Function} callbackCloseMenu - Função para chamar se precisar fechar o menu.
36 */
37 const handleMapClick = (event, mapRect, callbackSetLocation, callbackCloseMenu) => {
38   // 1. Tenta fechar o menu se clicar fora dele
39   // (A lógica de checar se o menu está aberto e se o clique foi fora dele
40   // deve residir no componente App.vue ou onde o estado 'isMenuOpen' vive)
41   if (callbackCloseMenu) {
42     callbackCloseMenu(); // App.vue decide se fecha ou não
43   }
44
45
46   // 2. Se estiver no modo de definir localização manual
47   if (settingUserLocationMode.value) {
48     if (!mapRect) {
49       console.error("Map rect não disponível para calcular clique.");
50       settingUserLocationMode.value = false; // Sai do modo
51       return;
52     }
53
54     // Calcula coordenadas do clique relativas ao elemento da imagem do mapa
55     const clickXRelative = event.clientX - mapRect.left;
56     const clickYRelative = event.clientY - mapRect.top;
57
58     // Converte coordenadas do clique para coordenadas relativas à imagem *não escalada*
59     // Precisamos considerar a origem da transformação (centro) e a escala
60     const originX = mapRect.width / 2;
61     const originY = mapRect.height / 2;
62     const scale = mapScale.value;
63
64     // Posição na imagem como se não houvesse escala (em pixels, relativa ao canto sup esq)
65     const imageX = (clickXRelative - originX) / scale + originX;
66     const imageY = (clickYRelative - originY) / scale + originY;
67
68     // Converte para porcentagem (0-100) relativa à dimensão da imagem
69     const xPercent = (imageX / mapRect.width) * 100;
70     const yPercent = (imageY / mapRect.height) * 100;

```

```

72
73     // Chama a função de callback para efetivamente definir a localização
74     if (callbackSetLocation) {
75         // Passa as coordenadas calculadas em %
76         callbackSetLocation(xPercent, yPercent);
77     }
78
79     settingUserLocationMode.value = false; // Sai do modo após definir
80 }
81 // 3. Lógica adicional de clique no mapa (ex: deselegionar local?) pode ser adicionada aqui
82 };
83
84
85 /** Ativa o modo de definir localização manual */
86 const enableSetUserLocationMode = () => {
87     settingUserLocationMode.value = true;
88 };
89
90
91 return {
92     mapScale,           // ref(number)
93     popupInfo,         // ref({ visible, x, y, text })
94     settingUserLocationMode, // ref(boolean) - Indica se está esperando clique para definir local
95
96     zoomIn,            // function
97     zoomOut,           // function
98     showPopup,         // function(local)
99     hidePopup,         // function()
100     handleMapClick,    // function(event, mapRect, callbackSetLocation, callbackCloseMenu)
101     enableSetUserLocationMode, // function()
102 };
103 }

```