

src/layouts/MapLayout.vue

```
1 <template>
2   <div class="map-container">
3     <div v-if="loading" class="loading">Carregando mapa...</div>
4     <div v-if="error" class="error">{{ error }}</div>
5
6     <div v-if="mapScale && mapDimensions" class="map-content" :style="{ width: `${mapDimensions.width}px`, height:
7       `${mapDimensions.height}px` }">
8       
9
10      <div v-for="loc in locations" :key="loc.id" class="location-pin"
11        :style="{ left: `${loc.x * mapScale.x}%`, top: `${loc.y * mapScale.y}%` }"
12        @click="selectLocation(loc)">
13        {{ loc.nome }}
14      </div>
15
16      <div v-for="segment in routeSegments" :key="segment.index" class="route-segment"
17        :style="{ left: `${segment.x * mapScale.x}%`, top: `${segment.y * mapScale.y}%`,
18          width: `${segment.length * mapScale.x}%`, transform: `rotate(${segment.angle}deg)` }">
19
20      <div v-if="userPosition" class="user-position"
21        :style="{ left: `${userPosition.x * mapScale.x}%`, top: `${userPosition.y * mapScale.y}%` }">
22        Você está aqui
23      </div>
24    </div>
25
26    <div class="location-details" v-if="selectedLocation">
27      <h2>{{ selectedLocation.nome }}</h2>
28      <p>{{ selectedLocation.descricao }}</p>
29      <button @click="clearSelection">Fechar</button>
30    </div>
31
32  </div>
33 </template>
34
```

```

35 <script setup>
36 import { ref, computed, onMounted } from 'vue';
37 import { useMapRouting } from '@/composables/useMapRouting';
38 import { useLocationStore } from '@/stores/location';
39 import { fetchInitialData } from '@/services/LocationService'; // Importe fetchInitialData diretamente
40
41 const locationStore = useLocationStore();
42 const locations = ref([]);
43 const loading = ref(true);
44 const error = ref(null);
45 const selectedLocation = ref(null);
46 const userPosition = ref({ x: 10, y: 10 }); // Posição inicial do usuário
47 const currentFloor = ref('terreo'); // Andar inicial
48 const mapScale = ref({ x: 1, y: 1 }); // Escala inicial
49 const mapDimensions = ref({ width: 800, height: 600 }); // Dimensões do mapa
50
51 onMounted(async () => {
52   try {
53     const data = await fetchInitialData(); // Chame fetchInitialData diretamente
54     if (data) {
55       locations.value = data.locais;
56       andaresDisponiveis.value = data.floors;
57     }
58   } catch (err) {
59     error.value = `Erro ao carregar dados: ${err.message}`;
60   } finally {
61     loading.value = false;
62   }
63 });
64
65 const andaresDisponiveis = ref(null);
66
67 const andarAtualInfo = computed(() => {
68   console.log('Verificando andar atual:', currentFloor.value);
69   console.log('Andares disponíveis:', andaresDisponiveis.value);
70
71   if (!andaresDisponiveis.value) {

```

```

72     return { nome: 'Carregando...', image: '' };
73 }
74
75 const andarInfo = andaresDisponiveis.value.find(andar => andar.nome.toLowerCase() === currentFloor.value);
76
77 if (!andarInfo) {
78     return { nome: 'Andar não encontrado', image: '' };
79 }
80
81 return {
82     nome: andarInfo.nome,
83     image: `/maps/${andarInfo.nome.toLowerCase()}.svg`
84 };
85 });
86
87 const { routeSegments } = useMapRouting(
88     userPosition,
89     selectedLocation,
90     currentFloor,
91     mapScale,
92     mapDimensions,
93
94 );
95
96 /***** ✨ Windsurf Command ✨ *****/
97 /**
98  * Updates the selected location to the provided location.
99  *
100  * @param {Object} location - The location object to be set as the selected location.
101  */
102
103 /***** b7aecb73-4466-4de7-b736-0ef4c0977dfc *****/
104 function selectLocation(location) {
105     selectedLocation.value = location;
106 }
107
108 function clearSelection() {

```

```
109     selectedLocation.value = null;
110 }
111 </script>
112
113 <style scoped>
114   .map-container {
115     position: relative;
116     width: 100%;
117     height: 100vh;
118   }
119
120   .map-content {
121     position: relative;
122     border: 1px solid #ccc;
123   }
124
125   .map-image {
126     width: 100%;
127     height: 100%;
128   }
129
130   .location-pin,
131   .user-position {
132     position: absolute;
133     background-color: rgba(255, 0, 0, 0.7);
134     color: white;
135     padding: 5px;
136     border-radius: 5px;
137     cursor: pointer;
138   }
139
140   .route-segment {
141     position: absolute;
142     height: 3px;
143     background-color: blue;
144     transform-origin: left;
145   }
```

```
146
147 .location-details,
148 .debug-waypoints {
149     position: absolute;
150     top: 10px;
151     right: 10px;
152     background-color: white;
153     padding: 10px;
154     border: 1px solid #ccc;
155 }
156 </style>
```