

src/components/AppSidebar.vue

```
1 <script setup>
2 import { ref, computed, watch } from 'vue';
3 import LocationItem from './LocationItem.vue';
4
5 const props = defineProps({
6   isOpen: Boolean,
7   allLocais: Array,
8   floors: Array,
9   currentFloorId: String,
10  selectedLocalId: [Number, String, null],
11  loading: Boolean,
12  error: String
13 });
14
15 const emit = defineEmits(['update:currentFloorId', 'select-local', 'close']);
16
17 const searchTerm = ref('');
18
19 // Filtra locais baseado na busca E opcionalmente no andar atual
20 // Decisão: Mostrar sempre locais do andar atual + resultados da busca em outros andares
21 const filteredLocais = computed(() => {
22   const term = searchTerm.value.toLowerCase().trim();
23   if (!term) {
24     // Sem busca, mostra apenas os do andar atual
25     return props.allLocais.filter(local => local.andar === props.currentFloorId);
26   } else {
27     // Com busca, mostra os que batem com a busca (em qualquer andar)
28     return props.allLocais.filter(local =>
29       local.nome.toLowerCase().includes(term)
30     );
31   }
32 });
33
34
```

```
35 | const internalCurrentFloor = ref(props.currentFloorId);
36 |
37 | // Observa mudança externa no currentFloorId
38 | watch(() => props.currentFloorId, (newVal) => {
39 |   internalCurrentFloor.value = newVal;
40 | });
41 |
42 |
43 | const changeFloor = () => {
44 |   //searchTerm.value = ''; // Limpa busca ao trocar de andar? Opcional.
45 |   emit('update:currentFloorId', internalCurrentFloor.value);
46 | };
47 |
48 | const handleSelectLocal = (local) => {
49 |   emit('select-local', local);
50 |   // Opcional: Fechar sidebar ao selecionar?
51 |   // emit('close');
52 | };
53 |
54 | </script>
55 |
56 | <template>
57 |   <div class="sidebar" :class="{ 'is-open': isOpen }">
58 |     <h1>Locais</h1>
59 |
60 |     <div class="floor-selector">
61 |       <select v-model="internalCurrentFloor" @change="changeFloor">
62 |         <option v-for="floor in floors" :key="floor.id" :value="floor.id">
63 |           {{ floor.nome }}
64 |         </option>
65 |       </select>
66 |     </div>
67 |
68 |     <div class="search-container">
69 |       <input v-model="searchTerm" class="search-box" placeholder="Buscar local..." />
70 |     </div>
71 |
```

```

72     <div v-if="error" class="error">
73         {{ error }}
74     </div>
75
76     <div v-if="loading && !allLocais.length" class="status-message">Carregando locais...</div>
77
78     <div v-if="!loading && !filteredLocais.length && searchTerm" class="status-message">
79         Nenhum local encontrado para "{{ searchTerm }}".
80     </div>
81     <div v-if="!loading && !filteredLocais.length && !searchTerm" class="status-message">
82         Nenhum local neste andar.
83     </div>
84
85     <div class="locations-list">
86         <div
87             v-for="local in filteredLocais"
88             :key="local.id"
89             class="location-item"
90             :class="{ 'selected': selectedLocalId === local.id }"
91             @click="handleSelectLocal(local)"
92         >
93             {{ local.nome }}
94         </div>
95     </div>
96
97 </div>
98 </template>
99
100 <style scoped>
101     .sidebar {
102         width: 300px;
103         height: 100vh;
104         background: white;
105         position: fixed;
106         left: 0;
107         top: 0;
108         transform: translateX(-100%);

```

```
109     transition: transform 0.3s ease;
110     z-index: 1000;
111     box-shadow: 2px 0 5px rgba(0, 0, 0, 0.1);
112     padding: 20px;
113 }
114
115 .sidebar.is-open {
116     transform: translateX(0);
117 }
118
119 .floor-selector select {
120     width: 100%;
121     padding: 8px;
122     margin-bottom: 15px;
123     border: 1px solid #ddd;
124     border-radius: 4px;
125     background: white;
126 }
127
128 .search-container {
129     margin-bottom: 15px;
130 }
131
132 .search-box {
133     width: 100%;
134     padding: 8px;
135     border: 1px solid #ddd;
136     border-radius: 4px;
137 }
138
139 .locations-list {
140     max-height: calc(100vh - 200px);
141     overflow-y: auto;
142 }
143
144 .location-item {
145     padding: 10px;
```

```
146     margin: 5px 0;
147     background: #f8f9fa;
148     border: 1px solid #ddd;
149     border-radius: 4px;
150     cursor: pointer;
151     transition: all 0.2s ease;
152 }
153
154 .location-item:hover {
155     background: #e9ecef;
156 }
157
158 .location-item.selected {
159     background: #007bff;
160     color: white;
161     border-color: #0056b3;
162 }
163
164 .error {
165     color: #dc3545;
166     padding: 10px;
167     margin-bottom: 15px;
168     background: #f8d7da;
169     border-radius: 4px;
170 }
171
172 .status-message {
173     padding: 10px;
174     color: #6c757d;
175     text-align: center;
176 }
177 </style>
```