

src/components/LocationAdmin.vue

```
1 <template>
2   <div class="admin-container">
3     <router-link to="/">Voltar para o Mapa</router-link>
4     <h1>Admin de Locais</h1>
5
6     <div v-if="localLoading" class="loading">Carregando...</div>
7     <div v-if="localError" class="error">{{ localError }}</div>
8     <div v-if="localSuccessMessage" class="success">{{ localSuccessMessage }}</div>
9
10    <button @click="showAddForm" :disabled="localLoading || showForm" class="btn btn-primary">Adicionar Novo Local</button>
11
12    <div v-if="showForm" class="form-section">
13      <h2>{{ isEditing ? 'Editar Local' : 'Adicionar Local' }}</h2>
14      <form @submit.prevent="handleSubmit">
15        <div class="form-group" v-if="isEditing">
16          <label for="id">ID:</label>
17          <input type="text" id="id" :value="formData.id" disabled>
18        </div>
19        <div class="form-group">
20          <label for="tipo">Tipo:</label>
21          <select id="tipo" v-model="formData.tipo" required>
22            <option v-for="tipo in tiposDisponiveis" :key="tipo" :value="tipo">
23              {{ tipo.charAt(0).toUpperCase() + tipo.slice(1) }}
24            </option>
25          </select>
26        </div>
27        <div class="form-group">
28          <label for="nome">Nome:</label>
29          <input type="text" id="nome" v-model="formData.nome" required>
30        </div>
31        <div class="form-group">
32          <label for="andar">Andar:</label>
33          <select id="andar" v-model="formData.andar" required>
34            <option v-for="andar in andaresDisponiveis" :key="andar" :value="andar">
```

```

35         {{ andar.charAt(0).toUpperCase() + andar.slice(1) }}
36     </option>
37 </select>
38 </div>
39 <div class="form-group">
40     <label for="x">Coordenada X (0-100):</label>
41     <input type="number" id="x" v-model="formData.x" required step="0.01" min="0" max="100">
42 </div>
43 <div class="form-group">
44     <label for="y">Coordenada Y (0-100):</label>
45     <input type="number" id="y" v-model="formData.y" required step="0.01" min="0" max="100">
46 </div>
47 <div class="form-actions">
48     <button type="submit" :disabled="localLoading" class="btn btn-success">{{ isEditing ? 'Salvar Alterações' :
'Criar Local' }}</button>
49     <button type="button" @click="cancelForm" :disabled="localLoading" class="btn btn-secondary">Cancelar</button>
50 </div>
51 </form>
52 </div>
53
54 <h2>Locais Existentes</h2>
55 <div class="table-container">
56     <table class="locations-table">
57         <thead>
58             <tr>
59                 <th>ID</th>
60                 <th>Tipo</th>
61                 <th>Nome</th>
62                 <th>X</th>
63                 <th>Y</th>
64                 <th>Andar</th>
65                 <th>Ações</th>
66             </tr>
67         </thead>
68         <tbody>
69             <tr v-if="locations.length === 0 && !localLoading && !localError">
70                 <td colspan="7" style="text-align: center;">Nenhum local cadastrado.</td>

```

```

71     </tr>
72     <tr v-if="locations.length === 0 && localError">
73         <td colspan="7" style="text-align: center; color: red;">Falha ao carregar locais.</td>
74     </tr>
75     <tr v-for="loc in locations" :key="loc.id">
76         <td>{{ loc.id }}</td>
77         <td>{{ loc.tipo || 'N/A' }}</td>
78         <td>{{ loc.nome }}</td>
79         <td>{{ loc.x?.toFixed ? loc.x.toFixed(2) : loc.x }}</td>
80         <td>{{ loc.y?.toFixed ? loc.y.toFixed(2) : loc.y }}</td>
81         <td>{{ loc.andar }}</td>
82         <td>
83             <button @click="showEditForm(loc)" :disabled="localLoading || showForm" class="btn btn-warning btn-
sm">Editar</button>
84             <button @click="confirmDelete(loc.id)" :disabled="localLoading || showForm" class="btn btn-danger btn-
sm">Excluir</button>
85         </td>
86     </tr>
87 </tbody>
88 </table>
89 </div>
90 </div>
91 </template>
92
93 <script setup>
94 import { ref, onMounted, watch } from 'vue';
95 import { useRouter } from 'vue-router';
96 import locationService from '@services/LocationService.js'; // Importe a instância
97
98 const router = useRouter();
99 const locations = ref([]);
100 const localLoading = ref(false);
101 const localError = ref(null);
102 const localSuccessMessage = ref(null);
103
104 const showForm = ref(false);
105 const isEditing = ref(false);

```

```
106 const tiposDisponiveis = ['location', 'Waypoints'];
107 const andaresDisponiveis = ['terreo', 'primeiro'];
108
109 const saveFormToStorage = () => {
110   localStorage.setItem('locationFormData', JSON.stringify({
111     ...formData.value,
112     showForm: showForm.value,
113     isEditing: isEditing.value,
114   }));
115 };
116
117 const loadFormFromStorage = () => {
118   const savedData = localStorage.getItem('locationFormData');
119   if (savedData) {
120     const parsed = JSON.parse(savedData);
121     showForm.value = parsed.showForm;
122     isEditing.value = parsed.isEditing;
123     delete parsed.showForm;
124     delete parsed.isEditing;
125     formData.value = parsed;
126   }
127 };
128
129 const validateCoordinates = (value) => {
130   if (value === null || value === undefined) return 0;
131   const num = parseFloat(value);
132   return Math.max(0, Math.min(100, num));
133 };
134
135 const formData = ref({
136   id: null,
137   tipo: 'location',
138   nome: '',
139   andar: 'terreo',
140   x: null,
141   y: null,
142 });
```

```
143
144 const resetForm = () => {
145   formData.value = {
146     id: null,
147     tipo: 'location',
148     nome: '',
149     andar: 'terreo',
150     x: null,
151     y: null,
152   };
153   localStorage.removeItem('locationFormData');
154 };
155
156 onMounted((() => {
157   loadLocations();
158   loadFormFromStorage();
159 }));
160
161 watch([formData, showForm, isEditing], () => {
162   saveFormToStorage();
163 }, { deep: true });
164
165 function clearLocalMessages() {
166   setTimeout((() => {
167     localError.value = null;
168     localSuccessMessage.value = null;
169   }, 5000));
170 }
171
172 async function loadLocations() {
173   localLoading.value = true;
174   localError.value = null;
175   locations.value = [];
176   try {
177     localLoading.value = true;
178     const result = await locationService.getAllLocations(); // Use a instância
179     locations.value = result;
```

```
180 } catch (error) {
181   localError.value = `Erro ao carregar locais: ${error.message}`;
182   locations.value = [];
183 } finally {
184   localLoading.value = false;
185   if (localError.value) clearLocalMessages();
186 }
187 }
188
189 async function handleAddLocation() {
190   localLoading.value = true;
191   localError.value = null;
192   localSuccessMessage.value = null;
193   try {
194     const dataToSend = {
195       nome: formData.value.nome,
196       andar: formData.value.andar,
197       x: formData.value.x,
198       y: formData.value.y,
199     };
200     const success = await locationService.createLocation(dataToSend); // Use a instância
201     if (success) {
202       localSuccessMessage.value = 'Local adicionado com sucesso!';
203       await loadLocations();
204       cancelForm();
205       clearLocalMessages();
206     } else {
207       localError.value = 'Falha ao adicionar local.';
208       clearLocalMessages();
209     }
210   } catch (err) {
211     console.error('Erro ao adicionar local:', err);
212     localError.value = `Erro ao criar: ${err.message}`;
213     clearLocalMessages();
214   } finally {
215     localLoading.value = false;
216   }
```

```
217 }
218
219 async function handleUpdateLocation() {
220   localLoading.value = true;
221   localError.value = null;
222   localSuccessMessage.value = null;
223   try {
224     const dataToSend = {
225       id: formData.value.id,
226       nome: formData.value.nome,
227       andar: formData.value.andar,
228       x: formData.value.x,
229       y: formData.value.y,
230     };
231     const success = await locationService.updateLocation(dataToSend.id, dataToSend); // Use a instância
232     if (success) {
233       localSuccessMessage.value = 'Local atualizado com sucesso!';
234       await loadLocations();
235       cancelForm();
236       clearLocalMessages();
237     } else {
238       localError.value = 'Falha ao atualizar local.';
239       clearLocalMessages();
240     }
241   } catch (err) {
242     console.error('Erro ao atualizar local:', err);
243     localError.value = `Erro ao atualizar: ${err.message}`;
244     clearLocalMessages();
245   } finally {
246     localLoading.value = false;
247   }
248 }
249
250 async function handleDeleteLocation(id) {
251   localLoading.value = true;
252   localError.value = null;
253   localSuccessMessage.value = null;
```

```
254   try {
255     const success = await locationService.deleteLocation(id); // Use a instância
256     if (success) {
257       localSuccessMessage.value = 'Local excluído com sucesso!';
258       await loadLocations();
259       clearLocalMessages();
260     } else {
261       localError.value = 'Falha ao excluir local.';
262       clearLocalMessages();
263     }
264   } catch (err) {
265     console.error('Erro ao excluir local:', err);
266     localError.value = `Erro ao excluir: ${err.message}`;
267     clearLocalMessages();
268   } finally {
269     localLoading.value = false;
270   }
271 }
272
273 const handleSubmit = async () => {
274   try {
275     if (localLoading.value) return;
276     formData.value.x = validateCoordinates(formData.value.x);
277     formData.value.y = validateCoordinates(formData.value.y);
278     localLoading.value = true;
279     localError.value = '';
280     if (isEditing.value) {
281       await handleUpdateLocation();
282     } else {
283       await handleAddLocation();
284     }
285     resetForm();
286     showForm.value = false;
287     isEditing.value = false;
288     localStorage.removeItem('locationFormData');
289     await loadLocations();
290   } catch (error) {
```



```
291     console.error('Erro ao salvar local:', error);
292     localError.value = 'Erro ao salvar local. Por favor, tente novamente.';
293 } finally {
294     localLoading.value = false;
295 }
296 };
297
298 function showAddForm() {
299     resetForm();
300     isEditing.value = false;
301     showForm.value = true;
302     localError.value = null;
303     localSuccessMessage.value = null;
304 }
305
306 function showEditForm(location) {
307     isEditing.value = true;
308     formData.value = {
309         id: location.id,
310         nome: location.nome || '',
311         andar: location.andar || '',
312         x: location.x,
313         y: location.y,
314     };
315     showForm.value = true;
316     localError.value = null;
317     localSuccessMessage.value = null;
318 }
319
320 function cancelForm() {
321     resetForm();
322     showForm.value = false;
323     isEditing.value = false;
324     localStorage.removeItem('locationFormData');
325 }
326
327 function confirmDelete(id) {
```

```
328     if (window.confirm(`Tem certeza que deseja excluir o local com ID "${id}"? Esta ação não pode ser desfeita.`)) {
329         handleDeleteLocation(id);
330     }
331 }
332 </script>
333
334 <style scoped>
335 /* ... (seus estilos CSS) ... */
336 </style>
337
338 <style scoped>
339 .admin-container {
340     padding: 20px;
341     height: 100vh;
342     overflow-y: auto;
343     max-width: 1200px;
344     margin: 0 auto;
345     font-family: system-ui, -apple-system, BlinkMacSystemFont, 'Segoe UI', Roboto, Oxygen, Ubuntu, Cantarell, sans-serif;
346     background-color: #f8f9fa;
347 }
348
349 .admin-container h1 {
350     color: #2c3e50;
351     margin-bottom: 1.5rem;
352     font-size: 2rem;
353     font-weight: 600;
354 }
355
356 .admin-container h2 {
357     color: #2c3e50;
358     margin: 1.5rem 0;
359     font-size: 1.5rem;
360     font-weight: 500;
361 }
362
363 a {
364     color: #3498db;
```

```
365     text-decoration: none;
366     margin-right: 10px;
367     padding: 8px 12px;
368     border-radius: 4px;
369     background-color: #fff;
370     box-shadow: 0 1px 3px rgba(0,0,0,0.1);
371     transition: all 0.2s ease;
372 }
373
374 a:hover {
375     background-color: #3498db;
376     color: #fff;
377     text-decoration: none;
378 }
379
380 .form-section {
381     margin: 20px 0;
382     padding: 25px;
383     background-color: #fff;
384     border-radius: 8px;
385     box-shadow: 0 2px 4px rgba(0,0,0,0.1);
386 }
387
388 .form-group {
389     margin-bottom: 1.2rem;
390 }
391
392 label {
393     display: block;
394     margin-bottom: 0.5rem;
395     color: #2c3e50;
396     font-weight: 500;
397     font-size: 0.95rem;
398 }
399
400 input[type="text"],
401 input[type="number"],
```

```
402 | select {
403 |     width: 100%;
404 |     padding: 10px 12px;
405 |     border: 2px solid #e2e8f0;
406 |     border-radius: 6px;
407 |     font-size: 1rem;
408 |     background-color: #fff;
409 |     transition: all 0.2s ease;
410 | }
411 |
412 | input[type="text"]:hover,
413 | input[type="number"]:hover {
414 |     border-color: #cbd5e0;
415 | }
416 |
417 | input[type="text"]:focus,
418 | input[type="number"]:focus {
419 |     outline: none;
420 |     border-color: #3498db;
421 |     box-shadow: 0 0 0 3px rgba(52,152,219,0.2);
422 | }
423 |
424 | .form-actions {
425 |     margin-top: 1.5rem;
426 |     display: flex;
427 |     gap: 12px;
428 | }
429 |
430 | /* Estilos para a tabela */
431 | .table-container {
432 |     margin-top: 20px;
433 |     max-height: calc(100vh - 500px);
434 |     overflow-y: auto;
435 |     background-color: #fff;
436 |     border-radius: 8px;
437 |     box-shadow: 0 2px 4px rgba(0,0,0,0.1);
438 |     scrollbar-width: thin;
```

```
439     scrollbar-color: #94a3b8 #f1f5f9;
440 }
441
442 .table-container::-webkit-scrollbar {
443     width: 8px;
444 }
445
446 .table-container::-webkit-scrollbar-track {
447     background: #f1f5f9;
448     border-radius: 8px;
449 }
450
451 .table-container::-webkit-scrollbar-thumb {
452     background-color: #94a3b8;
453     border-radius: 8px;
454     border: 2px solid #f1f5f9;
455 }
456
457 .locations-table {
458     width: 100%;
459     border-collapse: separate;
460     border-spacing: 0;
461 }
462
463 .locations-table thead {
464     position: sticky;
465     top: 0;
466     z-index: 1;
467 }
468
469 .locations-table th {
470     background-color: #f8fafc;
471     color: #1e293b;
472     font-weight: 600;
473     padding: 16px;
474     text-align: left;
475     border-bottom: 2px solid #e2e8f0;
```

```
476     white-space: nowrap;
477 }
478
479 .locations-table td {
480     padding: 14px 16px;
481     border-bottom: 1px solid #e2e8f0;
482     color: #475569;
483 }
484
485 .locations-table tr:last-child td {
486     border-bottom: none;
487 }
488
489 .locations-table tbody tr {
490     transition: all 0.2s ease;
491 }
492
493 .locations-table tbody tr:hover {
494     background-color: #f1f5f9;
495 }
496
497 .locations-table td button {
498     margin-right: 8px;
499     padding: 6px 12px;
500     border-radius: 4px;
501     font-size: 0.875rem;
502     transition: all 0.2s ease;
503 }
504
505 /* Estilos para os botões */
506 .btn {
507     padding: 8px 16px;
508     border: none;
509     border-radius: 6px;
510     font-weight: 500;
511     cursor: pointer;
512     transition: all 0.2s ease;
```

```
513 | }
514 |
515 | .btn:disabled {
516 |     opacity: 0.6;
517 |     cursor: not-allowed;
518 | }
519 |
520 | .btn-primary {
521 |     background-color: #3498db;
522 |     color: white;
523 | }
524 |
525 | .btn-primary:hover:not(:disabled) {
526 |     background-color: #2980b9;
527 | }
528 |
529 | .btn-success {
530 |     background-color: #2ecc71;
531 |     color: white;
532 | }
533 |
534 | .btn-success:hover:not(:disabled) {
535 |     background-color: #27ae60;
536 | }
537 |
538 | .btn-warning {
539 |     background-color: #f1c40f;
540 |     color: #2c3e50;
541 | }
542 |
543 | .btn-warning:hover:not(:disabled) {
544 |     background-color: #f39c12;
545 | }
546 |
547 | .btn-danger {
548 |     background-color: #e74c3c;
549 |     color: white;
```

```
550 | }
551 |
552 | .btn-danger:hover:not(:disabled) {
553 |     background-color: #c0392b;
554 | }
555 |
556 | .btn-secondary {
557 |     background-color: #95a5a6;
558 |     color: white;
559 | }
560 |
561 | .btn-secondary:hover:not(:disabled) {
562 |     background-color: #7f8c8d;
563 | }
564 | </style>
```