src/components/AppSidebar.vue

```vue
<script setup>
import { ref, computed, watch } from 'vue';
import LocationItem from './LocationItem.vue';

const props = defineProps({
    isOpen: Boolean,
    allLocais: Array,
    floors: Array,
    currentFloorId: String,
    selectedLocalId: [Number, String, null],
    loading: Boolean,
    error: String
});

const emit = defineEmits(['update:currentFloorId', 'select-local', 'close']);

const searchTerm = ref('');

// Filtra locais baseado na busca E opcionalmente no andar atual
// Decisão: Mostrar sempre locais do andar atual + resultados da busca em outros andares
const filteredLocais = computed(() => {
    const term = searchTerm.value.toLowerCase().trim();
    if (!term) {
        // Sem busca, mostra apenas os do andar atual
        return props.allLocais.filter(local => local.andar === props.currentFloorId);
    } else {
        // Com busca, mostra os que batem com a busca (em qualquer andar)
        return props.allLocais.filter(local =>
            local.nome.toLowerCase().includes(term)
        );
    }
});


```

```
const internalCurrentFloor = ref(props.currentFloorId);

// Observa mudança externa no currentFloorId
watch(() => props.currentFloorId, (newVal) => {
    internalCurrentFloor.value = newVal;
});


const changeFloor = () => {
    //searchTerm.value = ''; // Limpa busca ao trocar de andar? Opcional.
    emit('update:currentFloorId', internalCurrentFloor.value);
};

const handleSelectLocal = (local) => {
    emit('select-local', local);
    // Opcional: Fechar sidebar ao selecionar?
    // emit('close');
};

</script>

<template>
    <div class="sidebar" :class="{ 'is-open': isOpen }">
        <h1>Local</h1>

        <div class="floor-selector">
            <select v-model="internalCurrentFloor" @change="changeFloor">
                <option v-for="floor in floors" :key="floor.id" :value="floor.id">
                    {{ floor.name }}
                </option>
            </select>
        </div>

        <input v-model="searchTerm" class="search-box" placeholder="Buscar local..." />

        <div v-if="error" class="error" style="position: static; transform: none; margin-bottom: 10px;">
            {{ error }}
```

```
        </div>

        <div v-if="loading && !allLocais.length">Carregando locais...</div>

        <div v-if="!loading && !filteredLocais.length && searchTerm">Nenhum local encontrado para "{{ searchTerm }}".
        </div>
        <div v-if="!loading && !filteredLocais.length && !searchTerm">Nenhum local neste andar.</div>


        <LocationItem v-for="local in filteredLocais" :key="local.id" :local="local"
            :isSelected="selectedLocalId === local.id" @select="handleSelectLocal" />

    </div>
</template>

<style scoped>
/* Estilos específicos da sidebar já estão em main.css */
/* Se precisar de algo MUITO específico, adicione aqui
.sidebar {
    /* Herdado de main.css
}

.sidebar.is-open {
    /* Herdado de main.css
}*/

/* Ajuste para erro dentro da sidebar */
.error {
    position: static;
    /* Sobrescreve o absoluto do global */
    transform: none;
    margin-bottom: 15px;
    width: auto;
    /* Garante que não fique gigante */
    left: auto;
    top: auto;
    text-align: left;
```

```
    padding: 10px;
    /* Menor padding */
}
</style>
```