

Atividade Avaliativa (Opcional na Avaliação 1)

— 3,0 pontos

Criação de Interface Vue.js (Introdução): `v-on`, `v-bind`, `v-model`, `v-if/else`, `v-for`, `localStorage`, `computed`

Tema do CRUD (proibido *to-do list*)

Catálogo de Empréstimo de Equipamentos de Laboratório. O estudante deve construir uma aplicação *single-page* com Vue.js (pode ser via CDN ou projeto mínimo Vue 3) para cadastrar, listar, atualizar e remover **equipamentos** (ex.: multímetro, notebook, projetor) e controlar seu **status de empréstimo** (“disponível” ou “emprestado”). Os dados devem **persistir em localStorage**.

Modelo de Dados (mínimo)

Cada equipamento deve possuir, no mínimo:

- `id` (string ou número, único);
- `nome` (string, obrigatório);
- `categoria` (string, obrigatório; ex.: “Informática”, “Medição”);
- `patrimonio` (string, obrigatório; ex.: “UNEMAT-1234”);
- `status` (enum: “disponível” | “emprestado”, obrigatório);

Requisitos Funcionais e de Interface

- R1 Cadastro (Create):** Formulário com `v-model` para todos os campos obrigatórios e botão “Adicionar”. Impedir inclusão com campos vazios (*validação simples*).
- R2 Listagem (Read):** Exibir os equipamentos em lista/tabela usando `v-for`. Quando a lista estiver vazia, exibir mensagem amigável usando `v-if/v-else`.
- R3 Edição (Update):** Permitir selecionar um item para edição (`v-on` em um botão “Editar”). Carregar os dados no formulário com `v-model`, permitir salvar as alterações.
- R4 Remoção (Delete):** Botão “Remover” por item; confirmar a remoção com `v-on` (ex.: `window.confirm`).
- R5 Status Visual (v-bind):** Destacar o status com ligação de atributos/estilos (por exemplo, `v-bind:class` ou `:class` aplicando cor diferente para “disponível”/“emprestado”).
- R6 Persistência:** Toda alteração (criar/editar/remover) deve atualizar o `localStorage`. Ao recarregar a página, a lista deve ser restaurada do `localStorage`.
- R7 Propriedades Computadas (computed):** Implementar, no mínimo, **duas** computadas:
- R7a)** Contadores derivados: total de equipamentos, total disponíveis, total emprestados;

R7b) Lista filtrada por `categoria` e/ou por `status` (opcionalmente ligada a um campo `v-model`).

R8 Acessibilidade mínima: rótulos associados aos campos, `aria-label` quando necessário, e foco coerente ao entrar em modo de edição.

Critérios de Avaliação (3,0 pontos)

A correção será **estritamente** baseada nos itens abaixo. Entregas parciais recebem pontuação proporcional.

C1 CRUD Funcional 1,40 pts

- Create com validação mínima (`v-model` + bloqueio de envio inválido) 0,35
- Read com `v-for` e mensagem vazia via `v-if/v-else` 0,25
- Update (carrega dados no formulário, salva alterações) 0,45
- Delete (confirmação e remoção do estado) 0,35

C2 Uso correto das diretivas básicas 0,80 pts

- `v-model` aplicado a todos os campos do formulário 0,20
- `v-on` para ações (adicionar, editar, salvar, remover, filtrar) 0,20
- `v-bind` (`:class`, `:disabled`, `:title` ou similares) para estados/atributos 0,15
- `v-if/v-else` (estado vazio, alternância entre “Adicionar”/“Salvar”) 0,15
- `v-for` com `:key` único e estável 0,10

C3 Persistência no localStorage 0,40 pts

- Salva alterações (criar/editar/remover) e restaura na carga inicial 0,40

C4 Propriedades computed úteis 0,40 pts

- Contadores derivados corretos (total/disponíveis/emprestados) 0,20
- Lista filtrada (por categoria e/ou status) derivada (`computed`, não método) 0,20

Total: 3,0 pontos

Critérios Objetivos de Aceite (testes rápidos do professor)

A aplicação:

T1 Ao abrir pela primeira vez, exibe mensagem de lista vazia (`v-if/v-else`).

T2 Permite cadastrar um equipamento válido; o item aparece imediatamente na lista (`v-for`).

T3 Ao apertar “Editar” em um item, o formulário é preenchido; “Salvar” atualiza **exatamente** aquele item.

T4 “Remover” solicita confirmação e exclui o item correto.

- T5** Recarregar a página mantém os dados (persistência via `localStorage`).
- T6** Os contadores computados refletem os números após qualquer operação.
- T7** A filtragem (quando aplicada) altera a lista sem perder os dados originais (derivada, não destrutiva).
- T8** Há pelo menos um `v-bind` visível (ex.: `cor/classe` do status, `:disabled` em botão quando formulário inválido).

Restrições e Observações

- **Não é permitido** implementar como lista de tarefas (*to-do list*).
- Pode usar apenas Vue 3 via CDN e JavaScript puro (sem backend). **Não** usar bibliotecas de estado/roteamento.
- IDs devem ser estáveis e usados como `:key` no `v-for`.
- O código deve estar organizado (componentização opcional; comentários claros).

Entrega

- Entregar um arquivo `.zip` contendo: `index.html`, `main.js` (e `styles.css`, se houver).
- Incluir no topo do `index.html` um bloco de comentário com: Nome completo, Matrícula, Turma.
- Opcional: publicar no GitHub Pages e incluir o link no `README.md`.

Penalidades

- Ausência de qualquer um dos campos obrigatórios do modelo de dados: $-0,10$ por campo, até $-0,30$.
- Erros que impeçam testar o CRUD: até $-0,50$.
- Não utilização de `computed` quando solicitado: $-0,40$ (C4).
- Não persistir no `localStorage`: $-0,40$ (C3).