

# Avaliação 1 — Frameworks Modernos para Desenvolvimento de Sistemas

UNEMAT — TADS

Entrega: 05/10/2025 às 23h59 (horário de Cuiabá)

Valor: 10,0 pontos • Trabalho em grupo (até 3 alunos) • Sem prorrogação

## Objetivo

Desenvolver uma aplicação **Vue 3 + Vuetify 3** que demonstre domínio de: **vue-router** (pode ser rota automática), **componentes**, e as diretivas/ligadores **v-for**, **v-if**, **v-else**, **v-bind**, **v-on**, **v-model**, além de **persistência em localStorage**. O app deve ter **no mínimo 3 páginas/rotas**, podendo incluir um CRUD simples.

## Requisitos Funcionais e Técnicos (mínimos)

- R1. Páginas/Rotas ( 3):** por exemplo, (i) página de CRUD, (ii) página de listagem em *cards* com componentes Vuetify, (iii) página “Sobre/Informações do Sistema”.
- R2. Roteamento:** usar **vue-router**. É permitido roteamento automático por arquivos (auto-routing).
- R3. Componentização:** criar e reutilizar componentes (com **props** e, quando fizer sentido, **emit/v-model** em componente).
- R4. Diretivas e Bindings:** empregar corretamente **v-for**, **v-if/v-else**, **v-bind**, **v-on**, **v-model**.
- R5. UI com Vuetify 3:** usar *layout*, *toolbar/app bar*, *cards*, *forms*, *dialogs*, *snackbars*, etc. conforme a necessidade.
- R6. Persistência:** salvar e ler dados do app em **localStorage**. Sincronizar a interface quando houver alterações.
- R7. CRUD:** ao menos uma entidade com **criar**, **listar**, **atualizar**, **remover** (podem ser dados simples).
- R8. Projeto executável:** app deve rodar via `npm run dev` ou `bun dev/pnpm dev`, com instruções no `README.md`.

## Temas sugeridos (escolha 1)

- T1. Catálogo de Receitas Saudáveis:** CRUD de receitas (título, ingredientes, modo de preparo, tags). Listagem em *cards* filtráveis; página “Sobre” com dicas gerais e versão do sistema.
- T2. Reserva de Equipamentos (Lab/Estúdio):** CRUD de equipamentos/itens; listagem com status (disponível/reservado) e filtros; página de informações do sistema e regras de uso.

- T3. Biblioteca de Jogos/Filmes:** CRUD de itens (título, gênero, nota); listagem com busca e ordenação; página “Sobre” com estatísticas básicas (contagem por gênero).
- T4. Agenda de Serviços (Barbearia/Clínica/etc.):** CRUD de clientes/serviços; visão em *cards* dos próximos atendimentos; página “Sobre” com horário de funcionamento.
- T5. Catálogo de Cursos/Disciplinas:** CRUD de disciplinas (nome, carga horária, professor); listagem com filtros; página institucional do sistema (versão, equipe).

## Entregáveis e Submissão

- **Código-fonte no GitHub (obrigatório):** repositório público *ou* privado com acesso de leitura. Incluir `README.md` com:
  - Descrição do projeto e tema escolhido.
  - Como instalar e executar (incluindo versão do Node/Bun e comando de *dev*).
  - Estrutura de rotas e componentes (breve).
- **Envio pelo SIGAA:** cada membro do grupo deve submeter o link do GitHub pelo SIGAA dentro do prazo.
- **Identificação do grupo:** colocar no `README.md` os nomes completos e matrículas dos integrantes (até 3).

## Critérios de Avaliação (10,0 pontos)

<b>C1. Arquitetura do projeto e execução (README, rodar sem erros)</b>	<b>1,0</b>
<i>Estrutura clara; comandos de execução funcionam; dependências corretas.</i>	
<b>C2. Router e navegação (3+ rotas, params, navegação coerente)</b>	<b>1,5</b>
<i>Uso de <code>vue-router</code>; rotas nomeadas/automáticas; navegação fluida.</i>	
<b>C3. Componentização (reuso, props/emit, v-model em componentes)</b>	<b>1,5</b>
<i>Separação de responsabilidades; comunicação entre componentes.</i>	
<b>C4. Diretivas e ligações (v-for, v-if/else, v-bind, v-on, v-model) corretas</b>	<b>2,0</b>
<i>Uso significativo e sem code smell; form handling adequado.</i>	
<b>C5. UI com Vuetify (layout, responsividade, usabilidade, acessibilidade básica)</b>	<b>1,5</b>

*Uso de componentes Vuetify; feedback ao usuário (dialogs/snackbars).*

**C6. Persistência em localStorage (salvar/ler/sincronizar estado) 1,0**

*Dados persistem entre recargas; tratamento de estado inicial.*

**C7. CRUD funcional (criar/listar/editar/remover com validações mínimas) 1,5**

*Fluxos completos e estáveis, sem erros no console.*

**C8. Documentação e organização (README, nomeação, limpeza de código) 0,5**

*Clareza, padronização, comentários pontuais.*

---

**TOTAL 10,0**

### Observações de correção

- **Penalizações:** erros recorrentes no console, quebra de build, ausência de requisitos mínimos, *UI* disfuncional (até -2,0); ausência de README (-0,5); menos de 3 rotas (-0,7); não usar Vuetify (-1,0); não usar localStorage (-1,0).
- **Bônus (até +0,5 no total, sem ultrapassar 10,0):** filtros/ordenadores bem feitos; *dialogs* de confirmação; máscaras/validações de formulário; *dark mode* do Vuetify.

## Diretrizes de Qualidade

- **Acessibilidade básica:** textos alternativos quando houver imagens, contraste legível, foco de teclado.
- **UX:** feedbacks visuais ao criar/editar/excluir; prevenção de ações destrutivas (confirmações).
- **Código:** evitar `any/any-like`, evitar lógica complexa em templates; usar `computed/watch` quando apropriado.

## Política de Formação de Grupos

- Até **3 integrantes**; todos devem contribuir.
- Todos os membros devem enviar o **mesmo link** do repositório no SIGAA.
- Informar integrantes no README.md. Mudanças após a entrega **não** serão consideradas.

## Política de Prazo e Originalidade

- **Prazo final:** 05/10/2025 às 23h59 (América/Cuiabá). **Sem prorrogação.**

- **Originalidade:** trabalhos idênticos ou com forte sobreposição entre grupos/tutoriais prontos serão zerados.
- Pode-se consultar material didático e documentação, mas o **código deve ser produzido pelo grupo**.

**Checklist rápido (para o grupo):**

- ☐ 3+ rotas funcionando
- ☐ CRUD completo
- ☐ Vuetify aplicado
- ☐ v-for
- ☐ v-if/else
- ☐ v-bind
- ☐ v-on
- ☐ v-model
- ☐ Persistência em `localStorage`
- ☐ README com instruções
- ☐ Link do GitHub enviado por **todos** no SIGAA