

Trabalho de Avaliação – Programação Web II

Aplicação Web – React

Desenvolver uma aplicação web completa, moderna e funcional, construída em React + Vite e TailwindCSS, que combine:

- Consumo de uma API externa real (mínimo 1)
- Persistência de dados local com JSON + json-server
- Componentes estruturados
- Gestão de estado com useState e useEffect
- Organização clara da interface e lógica

O tema é livre. O aluno escolhe aquilo que quer construir, desde que respeite todos os requisitos técnicos obrigatórios.

Tema

O aluno pode escolher qualquer área de interesse, desde que:

- exista uma API externa com dados relevantes
- faça sentido guardar informação criada pelo utilizador
- seja possível implementar CRUD local
- o projeto seja estruturado e funcional

Exemplos de Temas:

- Música (API iTunes)
- Filmes/Séries (TMDB)
- Receitas (MealDB)
- Livros (Google Books)
- Desporto (SportsDB)
- Marketplace básico
- Viagens (GeoDB + OpenWeather)
- Dashboard com widgets

Requisitos Técnicos Obrigatórios

- A) React + Vite obrigatório
- B) TailwindCSS obrigatório
- C) API Externa obrigatória
- D) CRUD obrigatório com json-server
- E) Componentes obrigatórios
- F) Organização obrigatória do projeto
- G) Interface limpa obrigatória

Funcionalidades Mínimas

- Leitura de dados externos (API)
- Persistência local com JSON (CRUD)
- Adicionar elementos (POST)
- Listar elementos (GET)
- Remover elementos (DELETE)
- Atualizar elementos (PATCH)

Funcionalidades Opcionais

- Sistema de pesquisa interna
- Filtros e ordenação
- Paginação
- Vários endpoints da API
- Animações (Framer Motion)
- Dark mode
- Gráficos simples

Tarefas a Desenvolver

Com base no enunciado geral do Projeto Final React (Tema Livre) e nos conhecimentos adquiridos ao longo da 2.^a parte da disciplina, os alunos devem desenvolver uma aplicação web em **React + Vite**, com **TailwindCSS**, que integre obrigatoriamente **uma API externa real** e um sistema de persistência local com **JSON + json-server**.

A aplicação deve incluir, no mínimo, as seguintes funcionalidades:

1. Consumo de API Externa (obrigatório)

- A aplicação deve obter dados reais a partir de **pelo menos uma API externa**, usando `fetch()`.
- Deve existir uma listagem ou pesquisa que apresente dados vindos dessa API.
- É obrigatória a gestão de:
 - estado de carregamento (loading)
 - tratamento de erro (erro na API ou ausência de resultados)

2. Persistência Local com JSON (CRUD obrigatório)

- A aplicação deve permitir ao utilizador **guardar informação localmente** através de `db.json + json-server`.
- CRUD mínimo obrigatório:
 - **GET**: carregar dados guardados
 - **POST**: adicionar novos registo
 - **DELETE**: remover registo
- **PATCH/PUT** (Update) é opcional, mas valorizado.

3. Interfaces obrigatórias

A aplicação deve apresentar interfaces claramente separadas para:

- **Visualização/pesquisa de dados da API externa**
- **Gestão dos dados guardados localmente (CRUD)**
- **Dashboard/resumo simples** (ex.: total de itens guardados, categorias, favoritos, etc.)

4. Estruturação em componentes (obrigatório)

O projeto deve ter pelo menos:

- 1 componente de listagem
- 1 componente de item
- 1 componente de formulário
- 1 componente de página / view

5. Organização técnica do projeto (obrigatório)

- Código modular e organizado por pastas.
- Separação clara entre lógica de dados e interface.
- Não é aceite lógica “toda no App.jsx”.

Regras e Boas Práticas

O projeto deve estar organizado por pastas, por exemplo:

```
src/
  |- components/
  |- pages/ (opcional)
  |- services/ (opcional para fetch)
  |- App.jsx
  L index.css
```

O uso de **Vite** é obrigatório.

O uso de **TailwindCSS** é obrigatório e consistente em toda a aplicação.

Os pedidos a APIs devem ser feitos via `fetch()` ou `axios`.

O json-server deve correr no porto 3001:

```
json-server --watch db.json --port 3001
```

O ficheiro `db.json` tem de estar na raiz do projeto.

O estado React tem de estar sincronizado com a persistência JSON.

É obrigatório um `README.md` com:

- descrição do tema e objetivos
- API(s) externa(s) utilizada(s)
- instruções para correr o projeto
- breve explicação das funcionalidades

Notas Adicionais

1. Entrega

- Cada aluno deve entregar:
 - Um **relatório em PDF** com:
 1. **Capa**
 - Título do projeto
 - Nome do aluno
 - Curso e Unidade Curricular
 2. **Introdução**
 - Tema escolhido e objetivo geral da aplicação.
 3. **Desenvolvimento**
 - Descrição breve das principais funcionalidades implementadas (API externa + CRUD com json-server).
 - Print screens das páginas principais.
 - Explicação curta das decisões técnicas mais relevantes.
 4. **Conclusão**
 - O que foi aprendido ou conseguido com o projeto.
 5. **Bibliografia / Webgrafia**
 - Links da API utilizada e outras fontes consultadas.
 - Uma pasta ZIP com o projeto completo React + Vite + Tailwind + JSON.
 - O ficheiro db.json utilizado.
 - O nome do ficheiro ZIP deve seguir o formato: **react_final_nomeapelido.zip**