



Projeto 2: Borboleta

Este projeto é a **continuação direta do Projeto 1**, com o objetivo de migrar o site desenvolvido anteriormente para uma arquitetura baseada em **componentização com React**, utilizando o **framework Next.js**.

A proposta é introduzir conceitos fundamentais de aplicações reativas, como a separação da interface em componentes reutilizáveis, uso de estado interno para controle da interface, carregamento de dados via API e criação de rotas simples e dinâmicas.

Assim como no Projeto 1, o caráter **extensionista** deve ser mantido: o site continuará voltado a um público real, com impacto direto na comunidade.

Descrição

- Migrar o site do Projeto 1 para **componentização em React**.
- Utilizar **estado (useState, useEffect)** para interatividade.
- Realizar **chamada de API** (client-side) para preencher parte da interface.
- Implementar **rotas em Next.js**:
 - Rota simples (sem parâmetro) — ex.: /sobre

- Rota dinâmica (com parâmetro) — ex.: `/projetos/[id]`
- Manter o **caráter extensionista** (público-alvo real, impacto na comunidade) e o **README em formato de tutorial**.

Entregáveis

- **Repositório GitHub** com projeto Next.js.
- **README.md** (tutorial) cobrindo: ideação/protótipos, explicação extensionista, explicação sobre a vantagem da migração para componentes ReactJS, trechos de código comentados, como rodar, e aprendizados.
- **Deploy opcional** (ex.: Vercel) — não obrigatório.
- **Demonstração em aula/laboratório** com todos os integrantes apresentando sua parte.

Tech: escolhas didáticas (simples)

- CSS Modules e Global - `styles/globals.css` (sem Tailwind ou CSS-in-JS).
- `fetch` no cliente com `useEffect`.
- Sem Context API (apenas `useState` / `useEffect`).

Avaliação

A nota será composta por:

30% check-ins + 20% comprometimento com extensão + 50% funcionamento e entrega final

Os check-ins serão feitos em 3 momentos diferentes: Parte 1, Parte 2 e Parte 3.

Em todos os check-ins será necessário:

- Explicar o código desenvolvido até o momento da apresentação.
- Apresentar o andamento do Tutorial.
- Atualizar o GitHub com o código desenvolvido.
- Enviar no Moodle: link do GitHub, código desenvolvido (zipado), tutorial (PDF)

Todos os check-ins serão acompanhados de uma apresentação na aula de laboratório, todos os integrantes do grupo devem explicar a parte em que trabalho e estarem prontos para questionamentos sobre as estratégias de implementação. A não apresentação será contada como não entrega, desta forma o grupo ficará com nota zero na entrega que não apresentar. TODOS DEVEM ESTAR PRESENTES.

Uma nota não será atribuída para cada check-in, a nota final do projeto será fornecida após a entrega final do mesmo.

Para cada check-in, espera-se:

Parte 1: Estrutura inicial do projeto em NextJS com estrutura básica e CSS inicial

Parte 2: Projeto totalmente migrado para ReactJS com NextJS, com todas as funcionalidades prontos, mas ainda sem consumo de API. Ainda pode ter parte a finalizar, mas o principal precisa estar migrado, sem código legado.

Parte 3: Projeto funcionando com API. Assim como na parte 2 ainda pode estar faltando detalhes para finalizar, mas nada que seja essencial para a validação desta parte.