

Justificativa: Uso do Método `fetch` no Navegador para Interação com o Backend

Durante o desenvolvimento do projeto com foco no backend, tomei a decisão de utilizar o método `fetch` no navegador para interagir com a API do backend em vez de recorrer a ferramentas dedicadas como o Postman ou Insomnia. Essa escolha foi baseada em algumas considerações técnicas e de implementação que julguei relevantes para o contexto do desenvolvimento:

1. **Integração Simples e Testes Iterativos:**
 - Optei pelo `fetch` no navegador devido à facilidade de integração direta com o código front-end e à conveniência de realizar testes iterativos diretamente no ambiente em que o aplicativo seria executado.
2. **Feedback em Tempo Real e Agilidade no Desenvolvimento:**
 - Utilizar o `fetch` permitiu obter feedback em tempo real durante a implementação das funcionalidades do frontend que dependiam das chamadas à API do backend. Isso facilitou a identificação e correção de problemas de forma ágil e eficiente.
3. **Adequação ao Escopo e Complexidade do Projeto:**
 - Considerando que o foco principal do projeto era o backend, a utilização do `fetch` no frontend proporcionou uma maneira direta de realizar testes simples e específicos das funcionalidades da API, adequando-se ao escopo e à natureza do projeto.
4. **Familiaridade e Facilidade de Implementação:**
 - Como parte da equipe de desenvolvimento full-stack, minha familiaridade com JavaScript e a API `fetch` contribuiu para a escolha dessa abordagem, tornando o processo de interação com o backend mais familiar e de implementação direta.

É válido ressaltar que, embora reconheça as vantagens e recursos robustos oferecidos por ferramentas dedicadas como o Postman ou Insomnia para testes e validações detalhadas de APIs, para este contexto específico de desenvolvimento centrado no backend, a escolha do `fetch` no navegador foi considerada mais apropriada para agilidade, integração e compatibilidade com o escopo do projeto.

Essa decisão foi tomada levando em conta a natureza mais simplificada e iterativa dos testes, alinhada ao contexto do desenvolvimento backend.

EXEMPLO:

```
fetch("http://127.0.0.1:8000/cadastrar_produto",{  
  
  method: "POST",  
  
  headers: {
```

```
    'Content-Type': 'application/json'

  },
  body: JSON.stringify({
    nome: "Arroz",
    descricao: "Arroz do mato",
    preco: 8.99,
    imagem_url: "https://propao.agilecdn.com.br/2314_1.jpg"
  })
})

.then(e=>e.text())

.then(e=>console.log(e))
```