

UNIME – LAURO DE FREITAS
BACHARELADO EM SISTEMAS DE INFORMAÇÃO

JEAN CARLOS ROCHA NASCIMENTO

**DESENVOLVIMENTO COM REACT: CÁLCULO DE SOMA DE NÚMEROS
UTILIZANDO NEXT.JS**

Lauro de Freitas
2024

JEAN CARLOS ROCHA NASCIMENTO

**DESENVOLVIMENTO COM REACT: DOCUMENTAÇÃO DA CRIAÇÃO DA
PÁGINA WEB PARA SOMA DE NÚMEROS**

Trabalho apresentado à disciplina Desenvolvimento com React do Curso Bacharel em Sistemas de Informação da UNIME - Campus Lauro de Freitas, como requisito parcial para a obtenção de nota.

Orientadores:

Prof. Gian Carlo Decarli
Prof. Gian Mauricio Fritsche

Lauro de Freitas

2024

RESUMO

Este trabalho aborda o desenvolvimento de uma página web interativa em Next.js, focando na criação de um formulário que permite ao usuário inserir dois números e exibir a soma dos mesmos. O principal objetivo é implementar uma interface amigável que processe os dados inseridos de maneira eficiente e dinâmica, promovendo uma experiência responsável para os usuários.

Para atingir esse objetivo, foram adotados procedimentos que incluem a configuração inicial de um projeto Next.js e a construção de um componente que gerencia o estado dos números inseridos. A lógica de manipulação do formulário foi projetada para calcular a soma ao submeter os dados, evitando o carregamento da página e proporcionando um feedback instantâneo ao usuário. A utilização do hook useStatedo React permite um gerenciamento eficaz dos valores e resultados, garantindo que as operações matemáticas sejam realizadas corretamente.

Os resultados obtidos mostram que a implementação de uma interface simples e direta, aliada à manipulação de estado em tempo real, melhora a usabilidade da aplicação, facilitando a interação do usuário e a visualização dos resultados. Conclui-se que o uso do Next.js para o desenvolvimento de aplicações web dinâmicas é uma abordagem eficaz, alinhada às melhores práticas de desenvolvimento front-end.

Sugira-se que futuras implementações considerem a adição de funcionalidades, como validação de entradas para evitar entradas inválidas e melhorias visuais na interface, a fim de enriquecer ainda mais a experiência do usuário.

Palavras-chave: Next.js, React, Formulário, Soma de Números, Interação do Usuário.

ABSTRACT

This work addresses the development of an interactive web page in Next.js, focusing on the creation of a form that allows the user to enter two numbers and display their sum. The main objective is to implement a user-friendly interface that processes the entered data efficiently and dynamically, promoting responsible experience for users. To achieve this goal, procedures were adopted that include the initial configuration of a Next.js project and the construction of a component that manages the state of the entered numbers. The form handling logic was designed to calculate the sum when submitting the data, avoiding page loading, and providing instant feedback to the user. The use of the React useStated hook allows for effective management of values and results, ensuring that mathematical operations are performed correctly. The results obtained show that the implementation of a simple and direct interface, combined with real-time state handling, improves the usability of the application, facilitating user interaction and visualization of results. It is concluded that the use of Next.js for the development of dynamic web applications is an effective approach, aligned with the best practices of front-end development. It is suggested that future implementations consider adding features such as input validation to prevent invalid input and visual improvements to the interface to further enrich the user experience.

Keywords: Next.js, React, Form, Addition of Numbers, User Interaction.

SUMÁRIO

1- INTRODUÇÃO	6
2- OBJETIVOS	7
3- DESENVOLVIMENTO.....	8
4- CONCLUSÃO	9

1- INTRODUÇÃO

Atualmente, a criação de aplicações web interativas e responsivas é essencial para atender às demandas dos usuários, que buscam interfaces amigáveis e funcionais. O uso de frameworks e bibliotecas modernas, como Next.js e React, permite o desenvolvimento de soluções ágeis que não apenas atendem às necessidades do mercado, mas também melhoram a experiência do usuário por meio de interações dinâmicas e em tempo real.

Este trabalho propõe o desenvolvimento de uma página web utilizando Next.js, onde o usuário pode inserir dois números em um formulário simples. Ao submeter o formulário, a aplicação calcula e exibe a soma desses números. A escolha do Next.js se justifica por suas características inovadoras, que incluem renderização do lado do servidor e geração de páginas estáticas, proporcionando um desempenho superior e uma experiência mais fluida para o usuário.

A proposta visa não apenas implementar uma funcionalidade básica de soma, mas também ilustrar como a lógica de manipulação de estado e a interação do usuário podem ser realizadas de forma eficiente. O uso de hooks do React, especialmente o useState, é fundamental para gerenciar o estado dos valores inseridos e para fornecer feedback instantâneo, essencial em qualquer aplicação que vise uma boa experiência do usuário.

Neste contexto, o trabalho é organizado da seguinte forma: inicialmente, será apresentada a configuração do projeto Next.js e a criação do componente responsável pelo formulário. Em seguida, serão discutidas as lógicas de manipulação de eventos e atualização de estado, culminando na apresentação dos resultados obtidos e na análise da experiência do usuário ao interagir com a aplicação. Por fim, serão apresentadas considerações finais e sugestões para futuras implementações que visem enriquecer ainda mais a funcionalidade e a usabilidade da aplicação.

Assim, o presente estudo não apenas contribui para o entendimento prático da utilização do Next.js, mas também enfatiza a importância de um design centrado no usuário, que promove interações mais efetivas e intuitivas.

2- OBJETIVO

O presente trabalho tem como objetivo principal desenvolver uma página web interativa utilizando o Next.js, que permita ao usuário realizar operações simples de soma. Para atingir esse objetivo, foram definidos os seguintes objetivos específicos:

1. **Compreensão da Configuração do Projeto:** Comece um novo projeto Next.js, familiarizando-se com a estrutura e as ferramentas oferecidas pelo framework. Isso inclui a instalação do pacote create-next-app a configuração do ambiente de desenvolvimento.
2. **Criação do Componente de Formulário:** Desenvolva um componente na página principal (pages/index.js) que contenha um formulário para a inserção de dois números. Este componente deve ser intuitivo e responsivo, facilitando a interação do usuário.
3. **Implementação da Lógica de Soma:** Integrar a lógica necessária para calcular a soma dos dois números inseridos no formulário. Isso envolve a manipulação de eventos para capturar as entradas do usuário e a atualização do estado da aplicação.
4. **Atualização e Exibição dos Resultados:** Após o envio do formulário, a aplicação deve exibir a soma dos números inseridos de forma clara e destacada, promovendo uma experiência visual agradável e informativa.
5. **Melhoria da Experiência do Usuário:** Garantir que a interação do usuário com o formulário seja fluida, utilizando feedbacks visuais que indiquem erros ou confirmações durante a inserção dos dados. Isso inclui a utilização de mensagens de erro dinâmicas e a manutenção do estado dos campos de entrada.
6. **Avaliação da Usabilidade:** Analisar a experiência do usuário ao interagir com a aplicação, identificando possíveis áreas de melhoria. Isso envolve considerar aspectos como a especificação das instruções, a facilidade de uso do formulário e a eficácia do feedback fornecido.
7. **Proposta de Futuras Implementações:** Sugerir melhorias e expansões para a aplicação, como a adição de funcionalidades mais complexas (por exemplo, outras operações matemáticas) e melhorias na interface,

envolvem enriquecer ainda mais a interação do usuário e a funcionalidade da aplicação.

Dessa forma, espera-se que a realização deste trabalho contribua não apenas para o entendimento da utilização do Next.js na construção de aplicações web, mas também para a prática de boas abordagens em design de interfaces e interação com o usuário.

3- DESENVOLVIMENTO

O desenvolvimento da aplicação foi realizado em etapas, desde a configuração inicial do projeto até a implementação das funcionalidades desejadas. Este processo envolve diversas decisões técnicas e o uso de boas práticas de desenvolvimento. A seguir, detalhamos as etapas principais que foram apresentadas ao desenvolvimento da página web interativa.

1. Configuração do Projeto

Para iniciar o projeto, foi utilizada uma ferramenta create-next-app que simplifica a criação de novas aplicações Next.js. O comando `npx create-next-app@latest nome-do-projeto` foi executado no terminal, o que gerou automaticamente a estrutura básica do projeto, incluindo os diretórios essenciais, como `page` e `public`. Após a instalação, o servidor de desenvolvimento foi iniciado com `npm run dev`, permitindo a visualização em tempo real das alterações realizadas.

2. Estrutura do Componente do Formulário

A criação do componente foi realizada no arquivo `pages/index.js`, que representa a página inicial da aplicação. Este componente contém um formulário que aceita dois números como entrada. A estrutura do formulário foi projetada para ser intuitiva, com rótulos claros e campos de entrada adequados. A escolha do tipo de entrada **number** para os campos garante que apenas valores numéricos possam ser inseridos, evitando erros de validação.


```

<div>
  <h1>Calculadora de Soma</h1>
  <form onSubmit={handleSubmit}>
    <div>
      <label>
        Número 1:
      <input
        type="number"
        value={num1}
        onChange={(e) => setNum1(e.target.value)} // Atualiza o estado do número 1
      />
    </label>
  </div>
  <div>
    <label>
      Número 2:
    <input
      type="number"
      value={num2}
      onChange={(e) => setNum2(e.target.value)} // Atualiza o estado do número 2
    />
  </label>
</div>
  <button type="submit">Calcular</button>
</form>
  {soma !== null && <h2>Soma: {soma}</h2>} {/* Exibe a soma se não for nula */}
</div>

```

3. Manipulação de Eventos e Lógica de Soma

A lógica para calcular a soma dos números inseridos foi inovadora na função `handleSubmit`, que é chamada quando o formulário é apresentado. O uso `e.preventDefault()` impede que uma página seja recarregada, o que pode resultar na perda dos dados inseridos pelo usuário. Os valores dos campos são convertidos para números usando `parseFloat`, e a soma é calculada e armazenada no estado através da função `setSoma`.

```

const handleSubmit = (e) => {
  e.preventDefault(); // Evita o comportamento padrão do formulário
  const resultado = parseFloat(num1) + parseFloat(num2); // Converte as entradas
  setSoma(resultado); // Atualiza o estado da soma
};

```

4. Exposição dos Resultados

Após o projeto, a soma é exibida na interface apenas se não for nula. Essa condição é verificada por meio de uma renderização condicional, que melhora a experiência do usuário, fornecendo um feedback instantâneo sobre o resultado da operação.

```
{soma !== null && <h2>Soma: {soma}</h2>} {/* Exibe a soma se não for nula */}
```

5. Avaliação da Usabilidade

Para garantir uma boa experiência do usuário, foram realizadas avaliações da usabilidade da aplicação. O feedback dos usuários foi considerado ao longo do desenvolvimento, e os ajustes foram feitos na interface, como o esclarecimento das instruções e a adição de mensagens de erro. A aplicação foi testada em diferentes dispositivos para garantir sua responsividade e funcionalidade.

6. Futuras Implementações

Com base nas avaliações realizadas, sugerimos futuras implementações que poderiam enriquecer a aplicação. Entre essas sugestões estão a adição de mais operações matemáticas (como subtração, multiplicação e divisão), bem como melhorias visuais na interface, como animações e transições mais suaves. Essas melhorias visam proporcionar uma interação ainda mais envolvente e intuitiva para os usuários.

4- CONCLUSÃO

Através das etapas descritas, foi possível criar uma aplicação funcional e interativa que permite a soma de dois números, utilizando os princípios fundamentais do Next.js e práticas de desenvolvimento de interface. A experiência adquirida durante o processo não apenas aprimorou o conhecimento sobre o Next.js, mas também reforçou a importância de considerar a usabilidade e a interação do usuário em aplicações web.