

Exemplo Criar um Número Aleatório



Quando o usuário clicar no botão `Gerar número`, um número aleatório será exibido no label acima.

O que vamos aprender?

No curso anterior você aprendeu a criar componentes. Neste curso daremos um passo à frente e aprenderemos os conceitos:

- State Hook
- Evento `onClick`

O que é State Hook?

O **State Hook (`useState`)**, basicamente nos permite criar uma variável que ao mudar de valor, automaticamente, será alterada na tela do usuário.

Por exemplo, se a variável `nomeProduto` for alterada, precisamos que o usuário veja o novo nome do produto - o React permite que isso aconteça de forma automática - ou - `reativa`, como o nome da tecnologia implica.

O que é o evento onClick?

Ao clicar em um botão é comum que algo aconteça. Para chamarmos esta ação utilizamos o evento `onClick`.

Neste exemplo, quando o usuário clicar no botão, o código associado ao evento `onClick` será executado (a geração do número aleatório).

Esse é o conceito da programação de `eventos` - associamos um código a um evento, ou trigger. Quando este evento/trigger for disparado pela aplicação (ou pelo usuário) o código associado será executado.

Por que esses conceitos são úteis? 🤔

Por que o
evento onClick
é útil?



Ao clicar no botão
o evento onClick é
disparado



Esse evento executa
o código associado



E o State Hook
Por quê é útil?



Número aleatório:

69

Click no botão abaixo para gerar um número aleatório:

Gerar número

<

Com o State Hook
o valor da variável na tela
atualiza automaticamente







Passo 1

Criar uma variável de estado

```
const [ numeroAleatorio, setNumeroAleatorio ] = useState(0);
```



Passo 2

Exibir a variável na tela

```
<h1>{numeroAleatorio}</h1>
```

```
<h1>{numeroAleatorio}</h1>
```

Número aleatório:

41

Click no botão abaixo para gerar um número aleatório:

Gerar número



Passo 3

Criar a função
que gera um número
aleatório e atualiza
a variável

```
import React, { useState } from 'react';
import './estilo.css';

export default function PaginaInicial() {
  const [ numeroAleatorio, setNumeroAleatorio ] = useState(0);

  function gerarNumero() {
    const novoNumero = Math.floor(Math.random() * (100-1) + 1);
    setNumeroAleatorio(novoNumero);
  }

  return(
    <div className="conteudo-centralizado">
      <h3>Número aleatório:</h3>
      <h1>{numeroAleatorio}</h1>
      <div className="area-botao">
        <label>
          Click no botão abaixo para gerar um número aleatório:
        </label>

        <button onClick={gerarNumero}>
          Gerar número
        </button>
      </div>
    </div>
  );
}
```



Passo 4

Associar a função ao evento onClick

```
<button onClick={gerarNumero}>
  Gerar número
</button>
```




3. Entendendo o exemplo

[Voltar](#) [Suporte ao aluno](#) [Anotar](#) [Marcar como concluído](#)

Como vimos na aula anterior, nossa aplicação funciona da seguinte forma:

- O usuário clica no botão **Gerar número**;
- Um número aleatório substitui o número 0.

Para chegarmos nesse resultado precisamos entender o conceito de State Hook e o evento **onClick**.

State Hook

Utilizamos State Hook sempre que precisamos que uma variável seja alterada automaticamente na nossa tela (variável de estado).

Com o State Hook basicamente criamos uma variável e uma maneira do React entender que ela foi alterada (uma função atrelada a variável), como mostra a **Figura 1**.

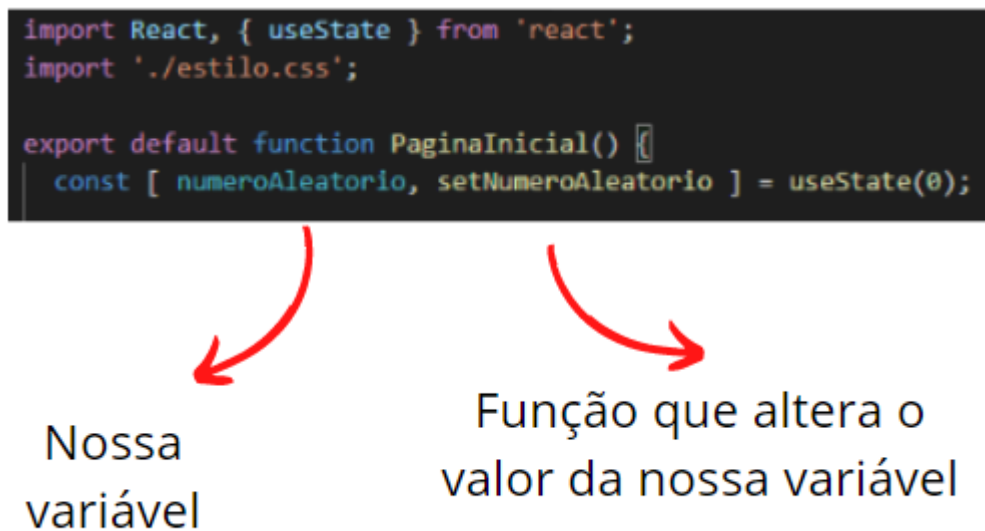


Figura 1.

Criando uma variável

No nosso exemplo criaremos uma variável de estado e a inserimos dentro do código JSX do nosso componente, como mostra a **Figura 2**.

```
return(  
  <div className="conteudo-centralizado">  
    <h3>Número aleatório:</h3>  
    <h1>{numeroAleatorio}</h1>  
    <div className='area-botao'>  
      <label>  
        | Click no botão abaixo para gerar um número aleatório:  
      </label>  
  
      <button>  
        | Gerar número  
      </button>  
    </div>  
  </div>  

```

Figura 2. Inserindo código JSX

Dessa forma o valor exibido será o da variável `numeroAleatorio` e, por ser uma variável de estado, será atualizada sempre que necessário, como vemos na **Figura 3**.

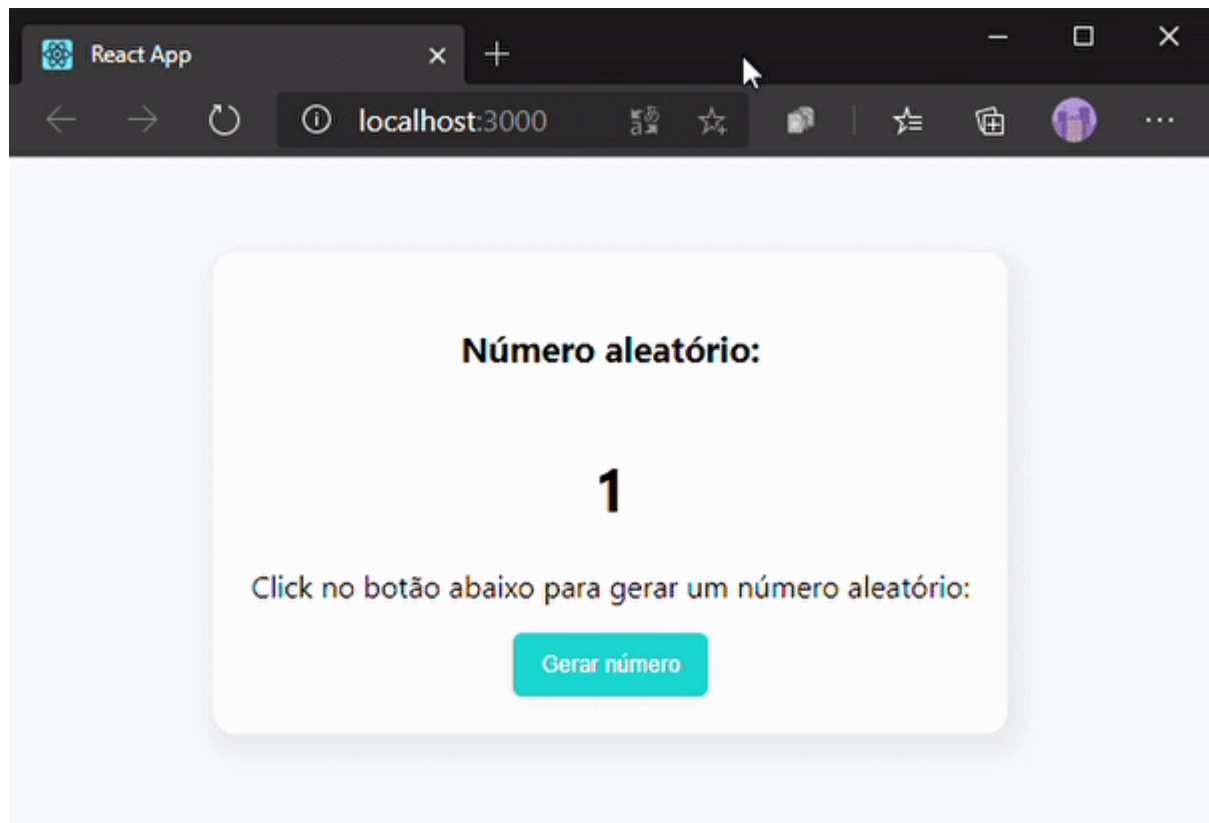


Figura 3. Exibindo o valor de numeroAleatorio

Evento `onClick`

É comum um botão executar uma ação ao ser clicado, sempre disparada pelo evento `onClick`.

Na nossa aplicação queremos que um número aleatório seja gerado e exibido na tela toda vez que clicarmos no botão, como vemos na **Animação 1**.



Animação 1. Exemplo criado

Para que isso seja possível precisamos criar essa ação, ou seja, vamos criar uma função e associá-la ao evento `onClick` do botão, como mostra a **Figura 4**.

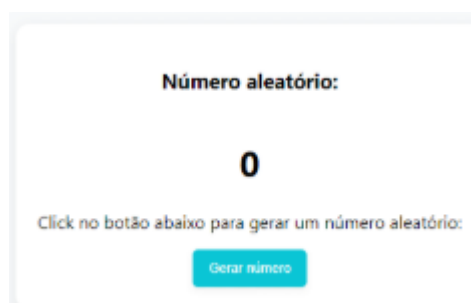


Figura 4. Associando ao evento `onClick`

Sempre que o botão for clicado gerará um número aleatório e atualizará a variável `numeroAleatorio` na página.

A estrutura base do nosso projeto é apresentada na **Figura 1**.

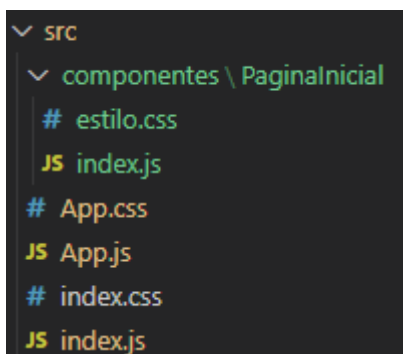


Figura 1. Estrutura base do projeto

Basicamente nosso projeto está dividido em dois componentes apresentados na **Figura 2**.

1. App - o componente principal da nossa aplicação;
2. PaginaInicial - o corpo da nossa aplicação.

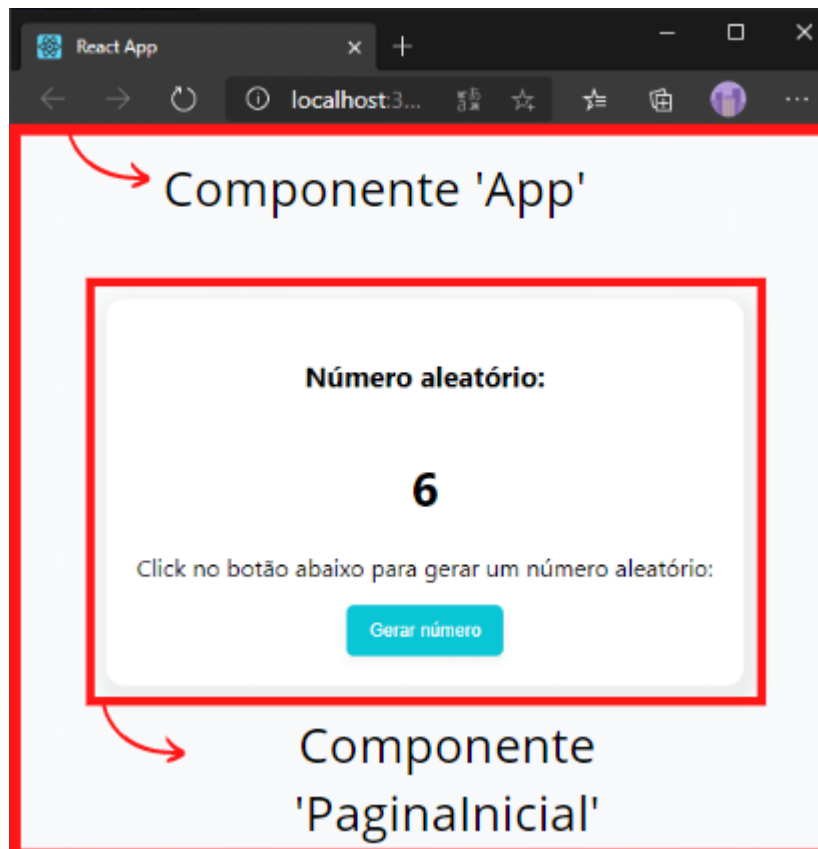
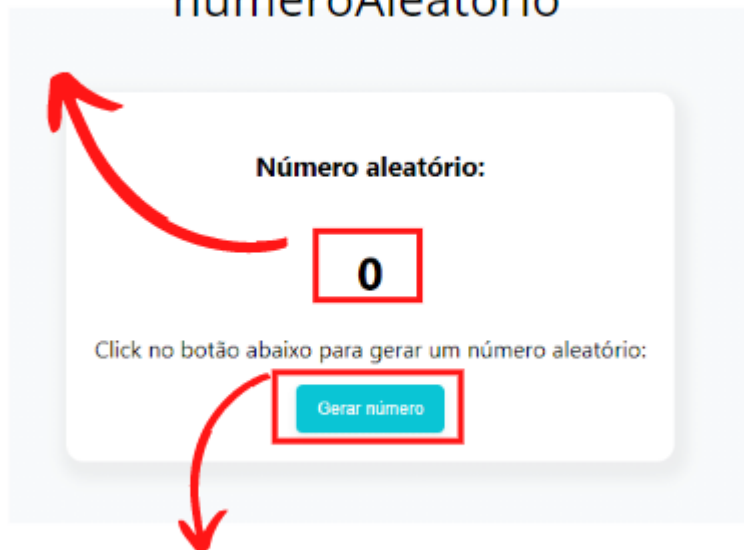


Figura 2. Componentes do projeto

No nosso componente `PaginaInicial` podemos destacar os seguintes elementos apresentados na **Figura 3**.

Tag 'h1' com a variável de estado
'numeroAleatorio'



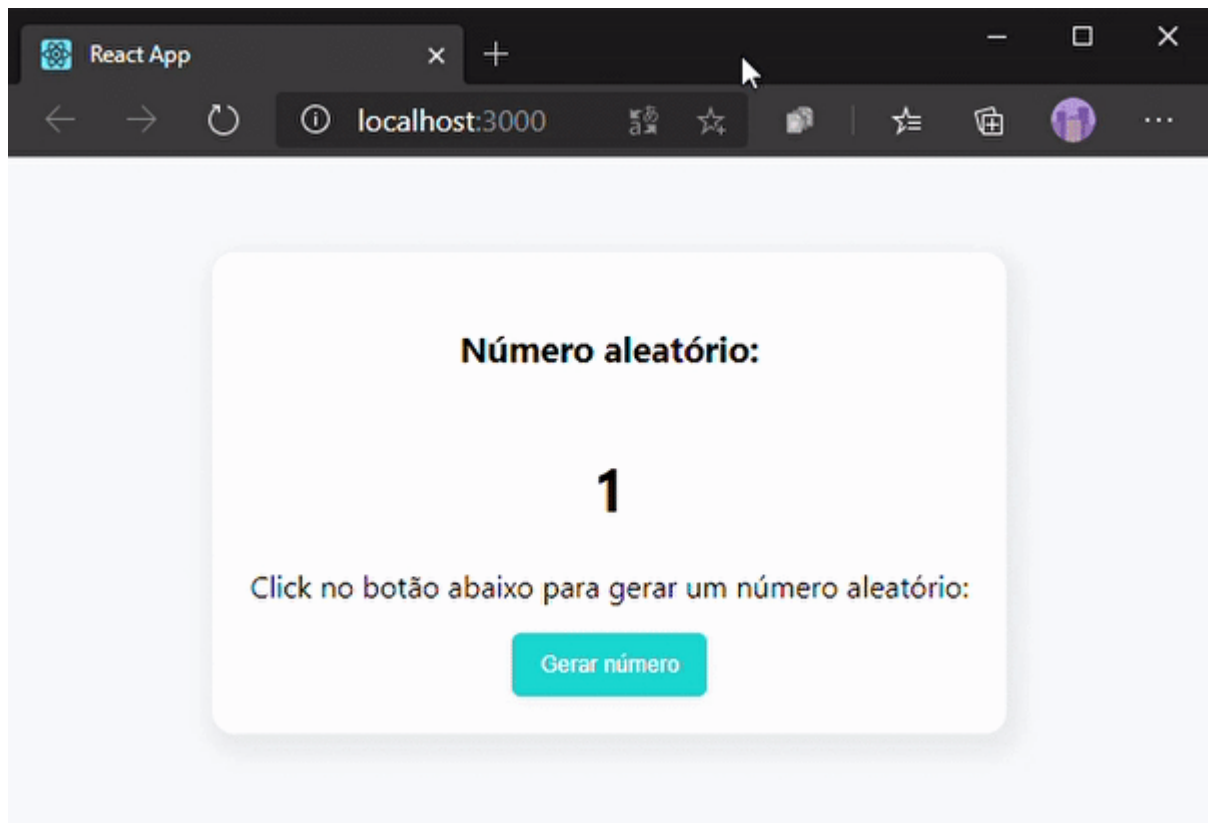
Tag 'button' que dispara a função
'gerarNumero' ao ser clicado

Figura 3. Componente PaginaInicial

6. Criando uma variável de estado

[Voltar](#) [Suporte ao aluno](#) [Anotar](#) [Marcar como concluído](#)

Com nossa estrutura base pronta começaremos a dar funcionalidade a nossa aplicação e deixar como podemos ver na **Animação 1**.



Animação 1. Página estática

Nosso primeiro passo é criar uma variável de estado e para isso abra o arquivo do componente `PaginaInicial` (componentes/Paginainicial/index.js).

Criando uma variável de estado

Segue no **Código 1** o componente com as alterações necessárias.

```

1  import React, { useState } from 'react';
2  import './estilo.css';
3
4  export default function PaginaInicial() {
5    const [ numeroAleatorio, setNumeroAleatorio ] = useState(1);
6
7    return(
8      <div className="conteudo-centralizado">
9        <h3>Número aleatório:</h3>
10       <h1>{ numeroAleatorio }</h1>
11
12       <div className='area-botao'>
13         <label>
14           Click no botão abaixo para gerar um número aleatório:
15         </label>
16
17         <button>
18           Gerar número
19         </button>
20       </div>
21     </div>
22   );
23 }

```

Código 1. Componente PaginaInicial

Entendendo o código:

Linha 1:

```
import React, { useState } from 'react';
```

Para criar e utilizar uma variável de estado precisamos importar o Hook `useState` da dependência `react`.

Importante:

1. Funções não possuem estado, por isso ao serem executadas suas variáveis deixam de ser acessíveis. Criar uma variável de estado utilizando o Hook `useState` significa preservá-la e mantê-la acessível.
2. Hooks são como gatilhos: mecanismo que permite ao React **reagir** a um momento específico. Em nosso exemplo, usamos Hooks para renderizar a tela automaticamente toda vez que a variável é modificada, e por isso a chamamos de variável de estado.

Aqui utilizamos o Hook `useState` para criar uma variável de estado.

Linha 5:

```
const [ numeroAleatorio, setNumeroAleatorio ] = useState(1);
```

Na linha 5 consumimos o Hook `useState` que foi importado. A sintaxe é:

```
const [NOME_DA_VARIAVEL_DE_ESTADO, FUNCAO_PARA_USO_INTERNO] =  
  
  useState(VALOR_INICIAL_DA_VARIAVEL);
```

Vejamos:

- **NOME_DA_VARIAVEL_DE_ESTADO** = é a variável que queremos manter o estado e acompanhar a sua mudança de valor.
- **FUNCAO_PARA_USO_INTERNO** = aqui apenas indicamos um nome de uma função. O React utilizará esse nome de função internamente para acompanhar a mudança de estado da variável. Não precisamos implementar essa função, apenas definir seu nome.
- **useState(VALOR_INICIAL_DA_VARIAVEL)** = no nosso exemplo iniciamos a variável com o valor 1 (mas poderia ser qualquer valor).

valor alterado exibido no navegador

Já temos nossa variável de estado criada e exibida na tela 😊

A variável de estado fica no componente PaginaInicial.

'useState'

Entendendo a função



Chamamos a função useState

```
const [ numeroAleatorio, setNumeroAleatorio ] = useState(1);
```

Passamos o valor 1 como
parâmetro

Essa função nos devolve um
array com dois valores

```
useState(1);
```

Retorna um array

```
[ variavel, setVariavel ]
```

function **useState(1)**



return [variavel, setVariavel];



Variável de estado com o
valor inicial 1

Função que vai alterar a
variável

const [numeroAleatorio, setNumeroAleatorio] = useState(1);



Apelidamos o
primeiro index de
'numeroAleatorio'

Apelidamos o
segundo index de
'setNumeroAleatorio'



[variavel, setVariavel]

numeroAleatorio



Variável de estado criada.

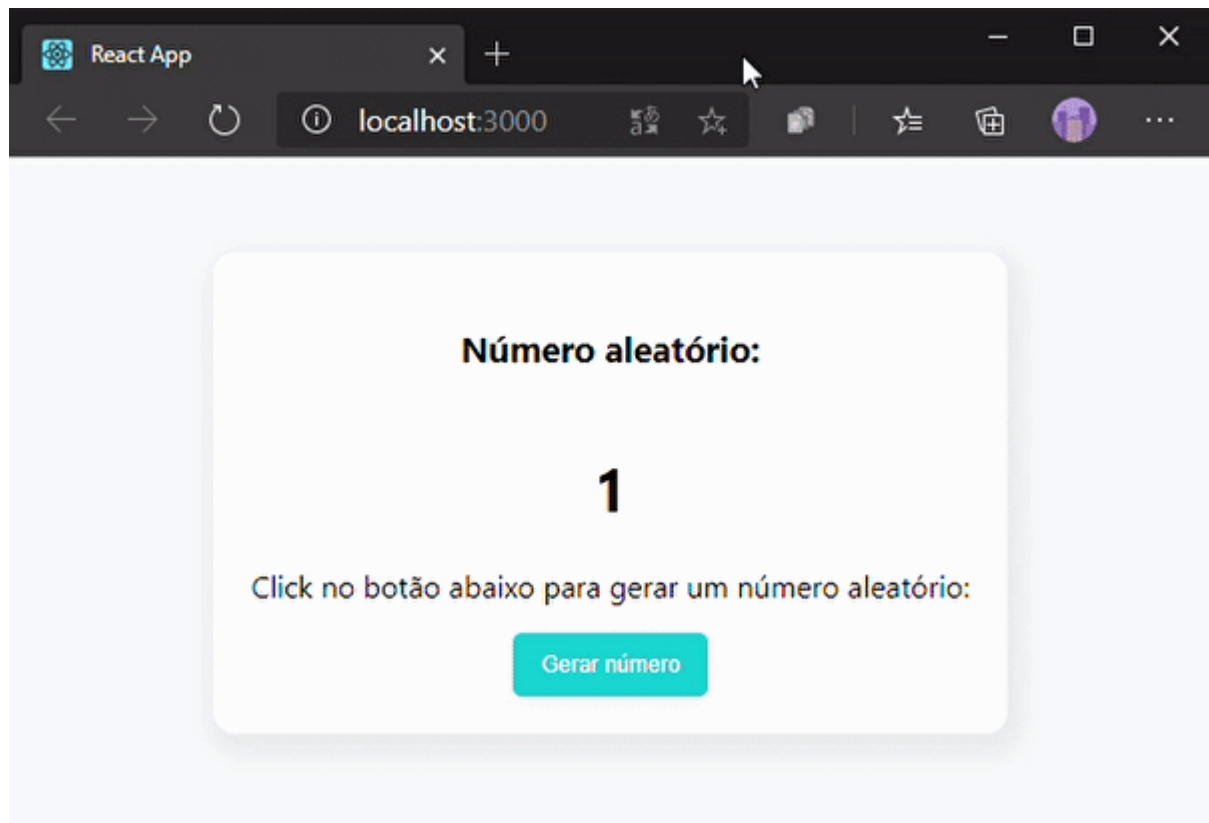


setNumeroAleatorio()



Função para alterar o valor
de 'numeroAleatorio'.

Com a variável de estado criada, nosso próximo passo é gerar um número aleatório e exibir na tela, como vemos na **Animação 1**.



Animação 1. Exemplo criado

Criação do Código da Pagina Inicial com a função que cria Gera Numero Aleatória

```

1  import React, { useState } from 'react';
2  import './estilo.css';
3
4  export default function PaginaInicial() {
5      const [ numeroAleatorio, setNumeroAleatorio ] = useState(1);
6
7      function gerarNumero() {
8          const novoNumero = Math.floor(Math.random() * (100-1) + 1);
9          setNumeroAleatorio(novoNumero);
10     }
11
12     return(
13         <div className="conteudo-centralizado">
14             <h3>Número aleatório:</h3>
15             <h1>{ numeroAleatorio }</h1>
16
17             <div className='area-botao'>
18                 <label>
19                     Click no botão abaixo para gerar um número aleatório:
20                 </label>
21
22                 <button onClick={ gerarNumero }>
23                     Gerar número
24                 </button>
25             </div>
26         </div>
27     );
28 }

```

Código 1. Evento onClick

Nas linhas de 7 a 10 criamos a função que vai gerar o número aleatório.

Na linha 8 basicamente criamos uma variável que vai receber o número aleatório. Utilizamos `Math.random() * (100-1) + 1` para gerar um número aleatório entre 1 e 100. Em seguida `Math.floor` para retornar um número inteiro.

`Math` é um objeto nativo da linguagem JavaScript. Ele possui propriedades e métodos para funções matemáticas.

No nosso exemplo utilizamos os seguintes métodos:

- `random()` - para gerar um número aleatório;
- `floor()` - para arredondar o número e retornar um inteiro.

Na linha 9 atribuímos o valor gerado a variável de estado `numeroAleatorio` através da função `setNumeroAleatorio()`.

Aqui vemos o conceito de Hook explicado na aula anterior. Observe que chamamos a função `setNumeroAleatorio` passando um novo valor para atualizar a variável de estado.

Nesse momento o React vai renderizar a tela, atualizando o valor de `numeroAleatorio` exibido na página.

Linha 21:

```
<button onClick={ gerarNumero }>
```

Para chamarmos a função `gerarNumero`, quando o botão for clicado, utilizamos o evento `onClick`. No React utilizamos camelCase para nomear eventos.

A função `gerarNumero` que vamos passar para o evento 'onClick' precisa ser passada entre chaves. Lembrando que no JSX colocamos código JavaScript entre chaves.

Ao salvar o arquivo podemos ver no navegador nossa aplicação funcionando, como mostra a **Animação 2**.

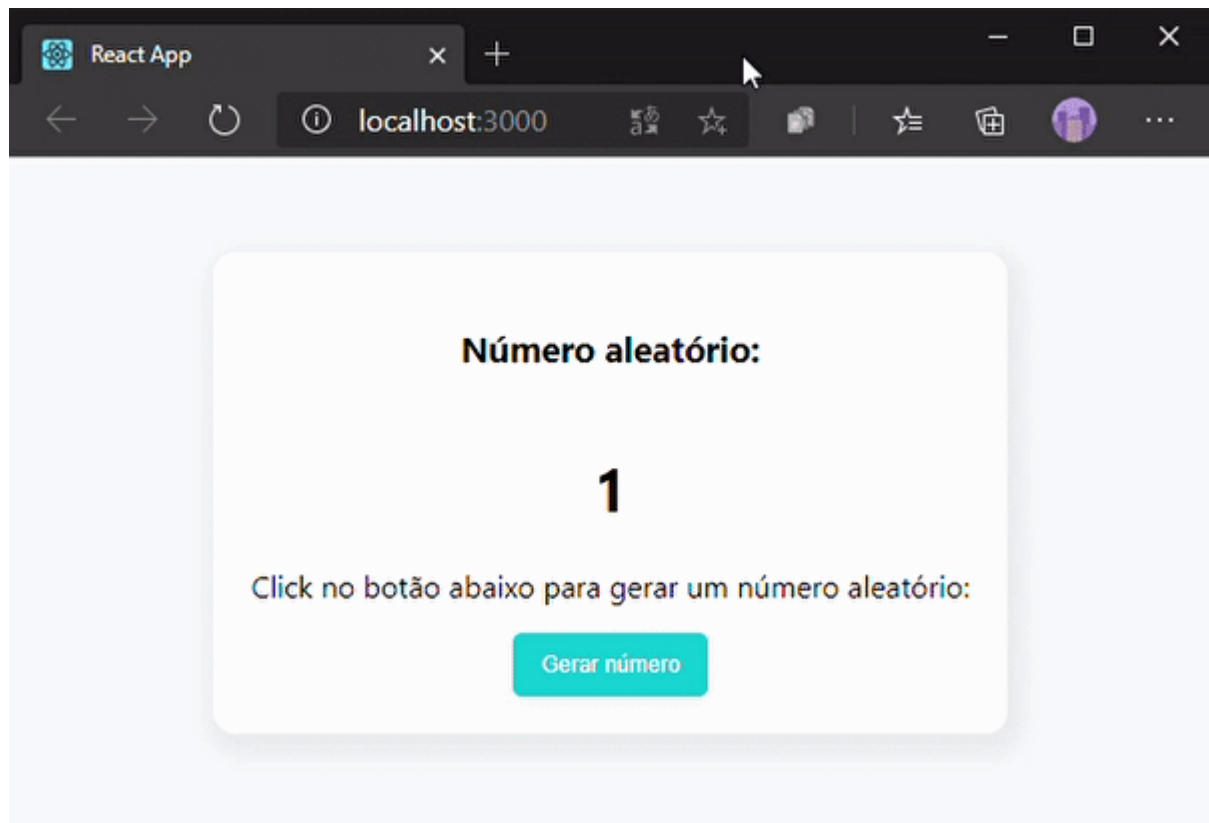
Último passo Alterar App.js arquivo principal do React JS

App.JS

```
1 import React from 'react';
2 import './App.css';
3 import PaginaInicial from './componentes/PaginaInicial';
4
5 function App() {
6   return (
7     <div className="App">
8       <PaginaInicial />
9     </div>
10   );
11 }
12
13 export default App;
14
```

App.css

```
src > # App.css > ...
1 .App {
2   display: flex;
3   justify-content: center;
4   align-items: center;
5   flex-direction: column;
6   width: 100vw;
7   height: 100vh;
8   background-color: #f8fafb;
9 }
10
```



Animação2. Aplicação funcionando

1 Para criar projeto no React

[Getting Started | Create React App \(create-react-app.dev\)](https://create-react-app.dev/)

```
npx create-react-app appaleatorio
```