

```
export const Formulario = (props) => {  
  return (  
    <form>  
      <input type="text" name="nome" />  
    </form>  
  )  
}
```

```
export const Formulario = (props) => {  
  return (  
    <form>  
      <input type="text" name="nome" />  
    </form>  
  )  
}
```

```
import { useState } from 'react';  
  
export const Formulario = (props) => {  
  const [nomeValor, setNomeValor] = useState(''); // definimos como uma  
  string vazia  
  return (  
    <form>  
      <input type="text" name="nome" />  
    </form>  
  )  
}
```

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

```
export const Formulario = (props) => {  
  const [nomeValor, setNomeValor] = useState(""); // definimos como uma string vazia  
  return (  
    <form>  
      <input type="text" name="nome" />  
    </form>  
  )  
}
```

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

export const Formulario = (props) => {
  const [nomeValor, setNomeValor] = useState('');
  return (
    <form>
      <input type="text" name="nome" value={nomeValor} />
    </form>
  )
}
```

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

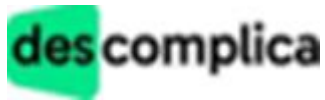
```
export const Formulario = (props) => {
  const [nomeValor, setNomeValor] = useState('');
  return (
    <form>
      <input type="text" name="nome" value={nomeValor} />
    </form>
  )
}
```

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

export const Formulario = (props) => {
  const [nomeValor, setNomeValor] = useState('');

  // Função que vai ser invocada toda vez que o usuário digitar no input
  const handleChange = (event) => {
    setNomeValor(e.target.value);
  };

  return (
    <form>
      <input
        type="text"
        name="nome"
        value={nomeValor}
        onChange={handleChange} // Evento que é invocado toda vez que o
        campo altera (ou deveria alterar)
      />
    </form>
  )
}
```



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

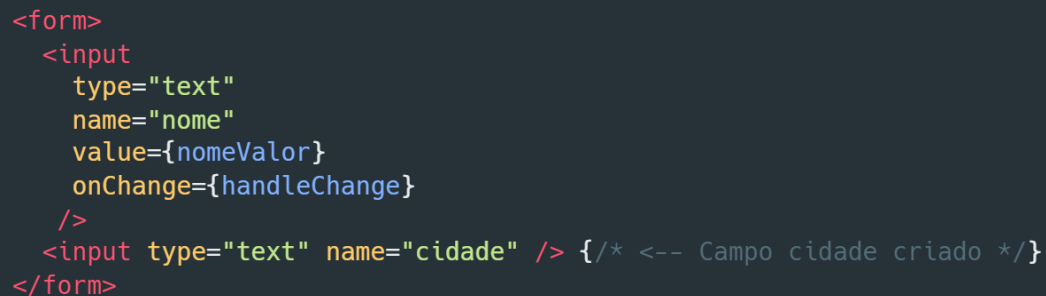
```
export const Formulario = (props) => {  
  const [nomeValor, setNomeValor] = useState("");
```

```
  // Função que vai ser invocada toda vez que o usuário digitar no input
```

```
  const handleChange = (event) => {  
    setNomeValor(e.target.value)  
  };
```

```
  return (  
    <form>  
      <input  
        type="text"  
        name="nome"  
        value={campoValor}  
        onChange={handleChange} // Evento que é invocado toda vez que o campo altera (ou  
deveria alterar)  
      />  
    </form>  
  )  
}
```

—



```
<form>  
  <input  
    type="text"  
    name="nome"  
    value={nomeValor}  
    onChange={handleChange}  
  />  
  <input type="text" name="cidade" /> { /* <-- Campo cidade criado */ }  
</form>
```

```
<form>  
  <input  
    type="text"  
    name="nome"  
    value={nomeValor}  
    onChange={handleChange}
```



```
    />
    <input type="text" name="cidade" /> { /* <-- Campo cidade criado */ }
  </form>
```

```
const [campoValor, setCampoValor] = useState({
  nome: '',
  cidade: ''
});
```

```
const [campoValor, setCampoValor] = useState({
  nome: '',
  cidade: ''
});
```

```
<form>
  <input
    type="text"
    name="nome"
    value={campoValor.nome}
    onChange={handleChange}
  />
  <input type="text" name="cidade" value={campoValor.cidade} />
</form>
```

```
<form>
  <input
    type="text"
    name="nome"
    value={campoValor.nome}
    onChange={handleChange}
  />
  <input type="text" name="cidade" value={campoValor.cidade} />
</form>
```

```
const handleChange = (event) => {  
  setCampoValor({  
    ...campoValor,  
    [event.target.name]: event.target.value  
  })  
};
```

```
const handleChange = (event) => {  
  setCampoValor({  
    ...campoValor,  
    [event.target.name]: event.target.value  
  })  
};
```

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

export const Formulario = (props) => {
  const [campoValor, setCampoValor] = useState({
    nome: '',
    cidade: ''
  });

  const handleChange = (event) => {
    setCampoValor({
      ...campoValor,
      [event.target.name]: event.target.value
    })
  };

  return (
    <form>
      <input
        type="text"
        name="nome"
        value={campoValor.nome}
        onChange={handleChange}
      />
      <input
        type="text"
        name="cidade"
        value={campoValor.cidade}
        onChange={handleChange}
      />
    </form>
  )
}
```

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

```
export const Formulario = (props) => {
  const [campoValor, setCampoValor] = useState({
    nome: "",
    cidade: ""
  });
```

```
  const handleChange = (event) => {
    setCampoValor({
      ...campoValor,
      [event.target.name]: event.target.value
    })
  };
}
```



```
return (  
  <form>  
    <input  
      type="text"  
      name="nome"  
      value={campoValor.nome}  
      onChange={handleChange}  
    />  
    <input  
      type="text"  
      name="cidade"  
      value={campoValor.cidade}  
      onChange={handleChange}  
    />  
  </form>  
)  
}
```

```
export const Formulario = (props) => {  
  // ... funções anteriores aqui  
  
  // Função que vai ser invocada quando formulário for enviado  
  const handleSubmit = (event) => {  
    event.preventDefault();  
  
    console.log(campoValor);  
  }  
  
  return (  
    <form onSubmit={handleSubmit}>  
      {/* campos aqui */}  
    </form>  
  )  
}
```

```
export const Formulario = (props) => {  
  // ... funções anteriores aqui  
  
  // Função que vai ser invocada quando formulário for enviado  
  const handleSubmit = (event) => {  
    event.preventDefault();  
  
    console.log(campoValor);  
  }  
}
```

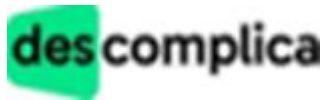


```
return (  
  <form onSubmit={handleSubmit}>  
    /* campos aqui */  
  </form>  
)  
}
```

```
export const Formulario = (props) => {  
  const handleSubmit = (event) => {  
    event.preventDefault();  
  }  
  
  return (  
    <form onSubmit={handleSubmit}>  
      <input  
        type="text"  
        name="nome"  
      />  
      <input  
        type="text"  
        name="cidade"  
      />  
    </form>  
  )  
}
```

```
export const Formulario = (props) => {  
  const handleSubmit = (event) => {  
    event.preventDefault();  
  }  
}
```

```
return (  
  <form onSubmit={handleSubmit}>  
    <input  
      type="text"  
      name="nome"  
    />  
    <input  
      type="text"  
      name="cidade"  
    />  
  </form>  
)
```

```
</form>
)
}
```

—

```
import { useRef } from 'react'; // importamos o hook

export const Formulario = (props) => {
  // Definimos um ref para cada campo, seu valor inicial é null pois aqui
  // estamos referenciando um elemento DOM e não o valor do input
  const nomeRef = useRef(null);
  const cidadeRef = useRef(null);

  const handleSubmit = (event) => {
    event.preventDefault();
  }

  return (
    <form onSubmit={handleSubmit}>
      <input
        type="text"
        name="nome"
        ref={nomeRef} // definimos que esse input é o nomeRef
      />
      <input
        type="text"
        name="cidade"
        ref={cidadeRef} // definimos que esse input é o cidadeRef
      />
    </form>
  )
}
```

```
import { useRef } from 'react'; // importamos o hook
```

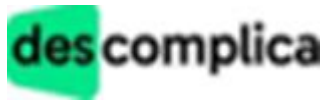
```
export const Formulario = (props) => {
  // Definimos um ref para cada campo, seu valor inicial é null pois aqui estamos
  // referenciando um elemento DOM e não o valor do input
  const nomeRef = useRef(null);
  const cidadeRef = useRef(null);

  const handleSubmit = (event) => {
    event.preventDefault();
  }
}
```



```
return (  
  <form onSubmit={handleSubmit}>  
    <input  
      type="text"  
      name="nome"  
      ref={nomeRef} // definimos que esse input é o nomeRef  
    />  
    <input  
      type="text"  
      name="cidade"  
      ref={cidadeRef} // definimos que esse input é o cidadeRef  
    />  
  </form>  
)  
}
```

```
import { useRef } from 'react';  
  
export const Formulario = (props) => {  
  const nomeRef = useRef(null);  
  const cidadeRef = useRef(null);  
  
  const handleSubmit = (event) => {  
    event.preventDefault();  
  
    // os ref recebem um objeto e quando associado a um elemento, ele  
    // preenche a propriedade current com o elemento  
    console.log(nomeRef.current.value, cidadeRef.current.value);  
  }  
  
  return (  
    <form onSubmit={handleSubmit}>  
      <input  
        type="text"  
        name="nome"  
        ref={nomeRef}  
      />  
      <input  
        type="text"  
        name="cidade"  
        ref={cidadeRef}  
      />  
    </form>  
  )  
}
```



```
import { useRef } from 'react';
```

```
export const Formulario = (props) => {  
  const nomeRef = useRef(null);  
  const cidadeRef = useRef(null);
```

```
  const handleSubmit = (event) => {  
    event.preventDefault();
```

```
    // os ref recebem um objeto e quando associado a um elemento, ele preenche a  
    propriedade current com o elemento
```

```
    console.log(nomeRef.current.value, cidadeRef.current.value);  
  }
```

```
  return (  
    <form onSubmit={handleSubmit}>  
      <input  
        type="text"  
        name="nome"  
        ref={nomeRef}  
      />  
      <input  
        type="text"  
        name="cidade"  
        ref={cidadeRef}  
      />  
    </form>  
  )
```

```
}
```