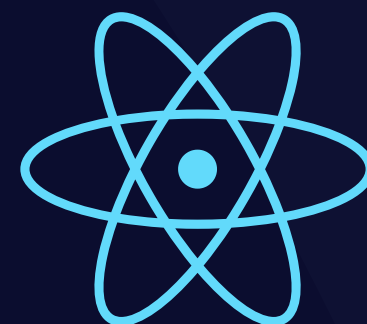


# React JS



## PRIMEIROS COMPONENTES

PROF. SINAIDE BEZERRA

CURSOS: ADS/SI

# CARACTERÍSTICAS DE UM COMPONENTE

Um componente deve ter:

## **Imports**

As bibliotecas/componentes necessários devem ser importados para que possam ser utilizados pelo componente.

## **Código**

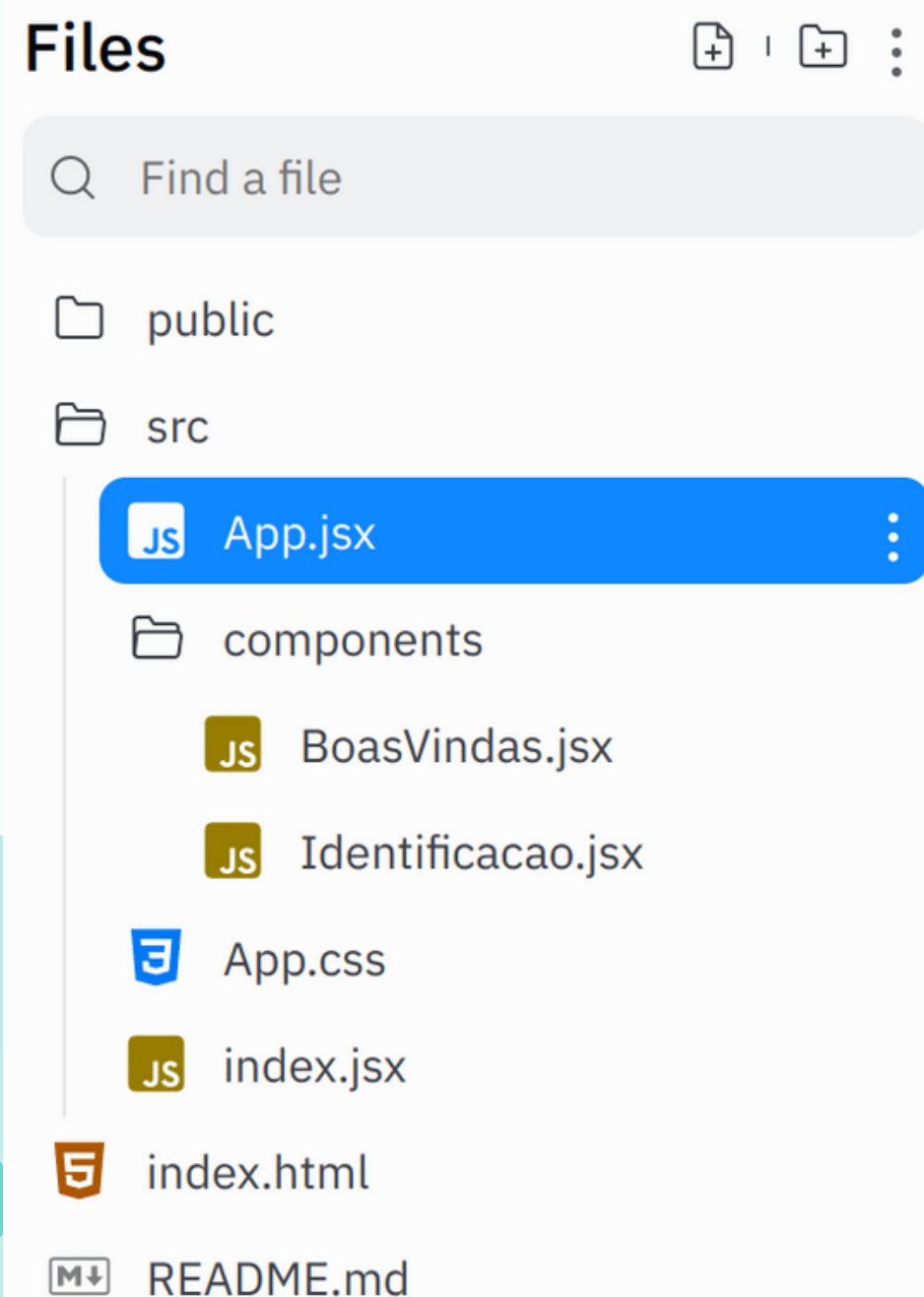
Códificação do componente

## **Exportação**

Para que o componente seja utilizado externamente, ele deve ser exportado.

# EXEMPLO DE DE UM COMPONENTE

Crie o arquivo BoasVindas.jsx em:



## Arquivo BoasVindas.jsx

```
import React from "react";

class BoasVindas extends React.Component {
  render() {
    return (
      <div>
        <strong>Bem vindo ao primeiro componente com React</strong>
      </div>
    )
  }
}

export default BoasVindas;
```

# EXEMPLO DE DE UM COMPONENTE

Altere/Insira no arquivo APP.jsx o código:

```
//bibliotec de estilo
import './App.css';
//import do componente BoasVindas
import BoasVindas from './components/BoasVindas';
import Identificacao from './components/Identificacao';

export default function App() {
  return (
    <main>
      <BoasVindas></BoasVindas>
    </main>
  )
}
```



Veja o que  
acontece!

# COMPONENTE IDENTIFICAÇÃO

Altere/Insira no arquivo **Identificacao.jsx** o código:

```
import React from "react";
class Identificacao extends React.Component {
  render() {
    return (
      <div>
        <p>Meu nome é Sinaide</p>
        <p>Meu apelido é Bezerra</p>
      </div>
    )
  }
}
export default Identificacao;
```

# INSERIR NOVO COMPONENTE

Altere/Insira no arquivo APP.jsx o código:

```
//bibliotec de estilo
import './App.css';
//import do componente BoasVindas
import BoasVindas from './components/BoasVindas';
import Identificacao from './components/Identificacao';

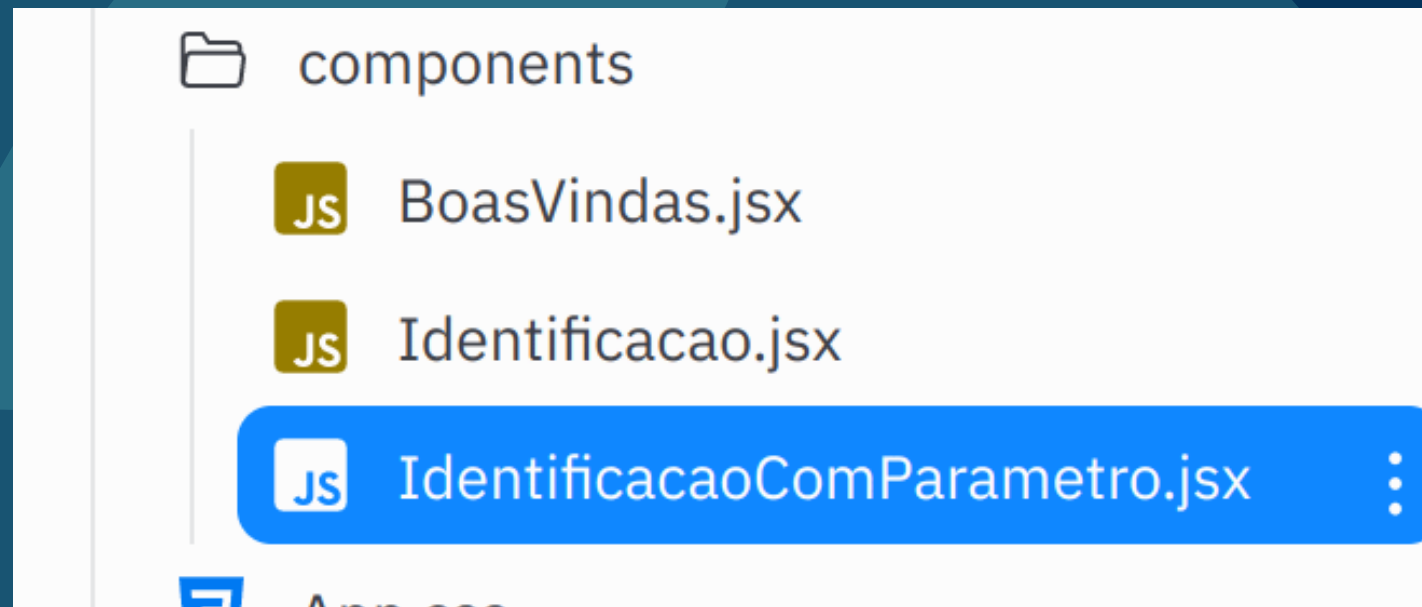
export default function App() {
  return (
    <main>
      <BoasVindas></BoasVindas>
      <Identificacao></Identificacao>
    </main>
  )
}
```



Veja o que  
acontece!

# COMO PASSAR PARÂMETROS (PROPS)

Em components crie o arquivo IdentificacaoComParametro.jsx



IdentificacaoComParametro.jsx

```
import React from "react";
```

```
export default function
```

```
IdentificacaoComParametro(props){
```

```
  return (
```

```
    <div>
```

```
      <p>Meu nome é {props.nome}</p>
```

```
      <p>Meu apelido é {props.apelido}</p>
```

```
      <p>Tenho {props.idade} anos</p>
```

```
    </div>
```

```
  )
```

```
}
```

# COMO PASSAR PARÂMETROS (PROPS)

Altere/Insira no arquivo APP.jsx o código:

```
import BoasVindas from './components/BoasVindas';
import IdentificacaoComParametro from './components/IdentificacaoComParametro';

export default function App() {
  return (
    <main>
      <BoasVindas></BoasVindas>
      <IdentificacaoComParametro
        nome="Sinaide"
        apelido = "Bezerra"
        idade = {48} />
    </main>
  )
}
```



Quais as diferenças  
entre o código  
atual e o anterior?



# INFORMAÇÕES GERAIS

Props são somente leitura

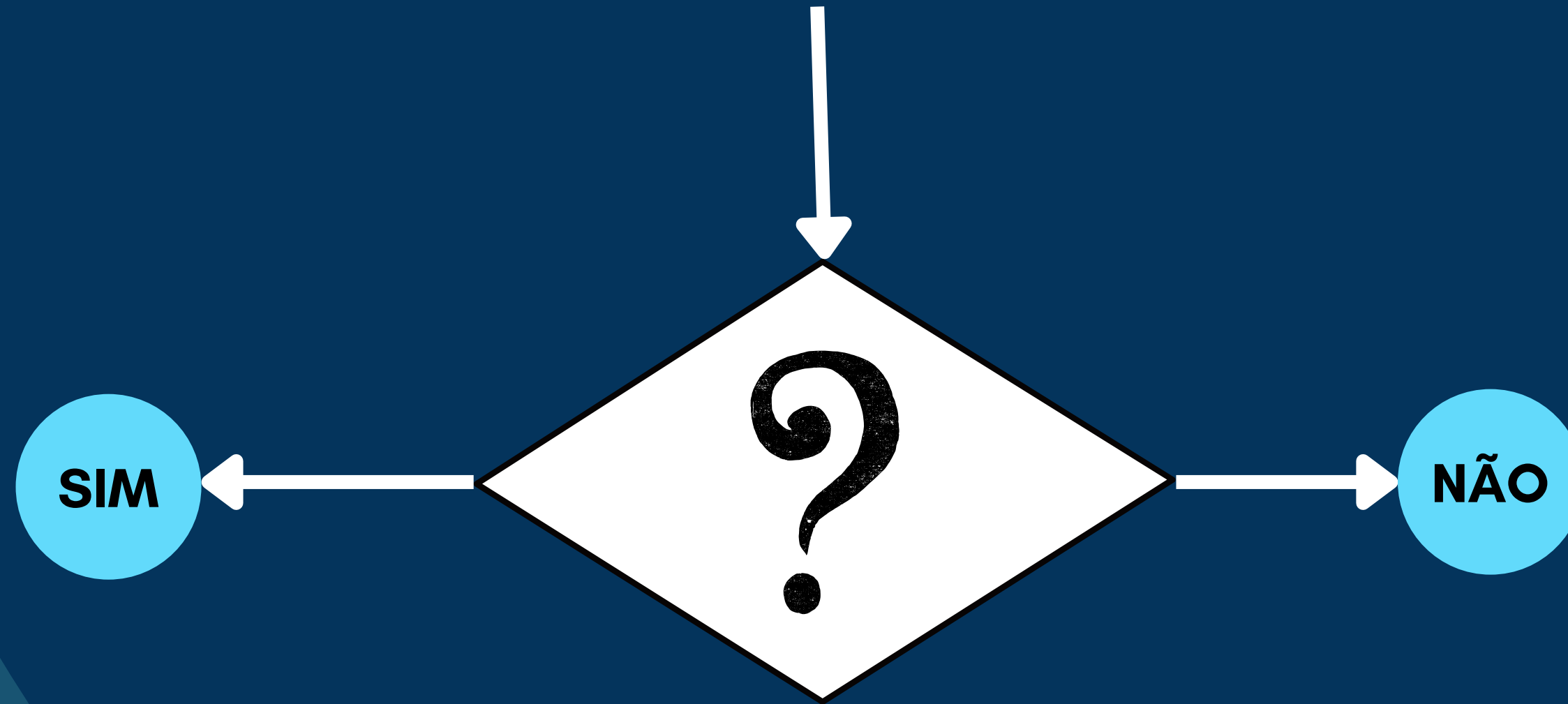
Comentários

- `{/*<Componente></Componente>* /}` ou `//`

Documentação no site React

- <https://pt-br.reactjs.org/docs/getting-started.html>

# ESTRUTURA CONDICIONAL



# CONDICIONAL EM REACT

Crie o arquivo onente APP.jsx o código:

```
import React from "react";

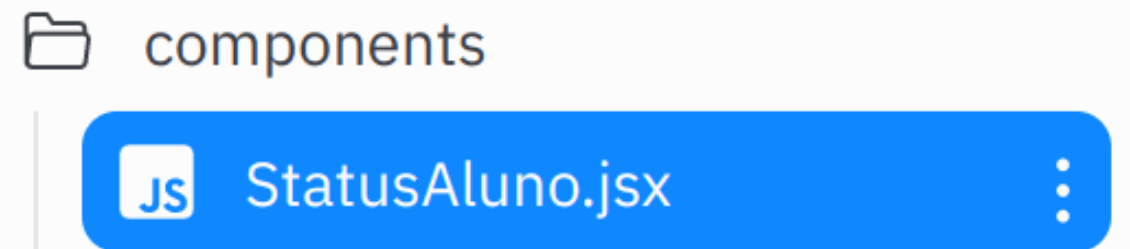
export default function StatusAluno(props) {
  const status = props.nota >= 6 ? "Aprovado" : "Reprovado";
  return (
    <div>
      <p>Meu nome é {props.nome}</p>
      <p>Minha nota na disciplina foi {props.nota}</p>
      <p>Eu fui {status}!</p>
    </div>
  )
}
```



Veja o que  
acontece!

# CONDICIONAL EM REACT

Crie o arquivo:



Altere App.jsx:

```
import StatusAluno from
'./components/StatusAluno';

export default function App() {
  return (
    <main>
      <StatusAluno
        nome="Sinaide"
        nota={8} />
    </main>
  )
}
```

## Arquivo StatusAluno.jsx

```
import React from "react";

export default function StatusAluno(props) {

  const status = props.nota >= 6 ? "Aprovado" : "Reprovado";

  return (
    <div>
      <p>Meu nome é {props.nome}</p>
      <p>Minha nota na disciplina foi {props.nota}</p>
      <p>Eu fui {status}!</p>
    </div>
  )
}
```

# SOBRE USO DE COMPONENTES

## Objetivo único

Um componente só pode ter um objetivo.

Componentes com mais de um objetivo são fadados à separação.

Exemplo:

Um formulário com três tags `<input>` de texto e um botão.

Pode ser separado em cinco componentes distintos:

Um deles seria o próprio formulário – para agrupar os demais.

Livro Learning React: A Hands-On Guide to Building Maintainable, High-Performing Web Application User Interfaces Using the React JavaScript Library.

Airbnb React/JSX Style Guide: <https://github.com/airbnb/javascript/tree/master/react/>

# SOBRE USO DE COMPONENTES

## Stateless ou Stateful

Um componente pode ter ou não estado.

Por estado, entenda que o componente precisar guardar valores dentro de si para viabilizar sua renderização. Componentes com estado (stateful) são mais complexos por natureza.

Componente que agrupa pequenos componentes que necessitam de seus próprios estados --> bom candidato à separação.

Um agrupador de componentes sem estado pode ser tratado como um único componente.

# SOBRE USO DE COMPONENTES

## Reutilização

Um componente muito grande é difícil de ser reutilizado.  
Um componente muito simplório também é.

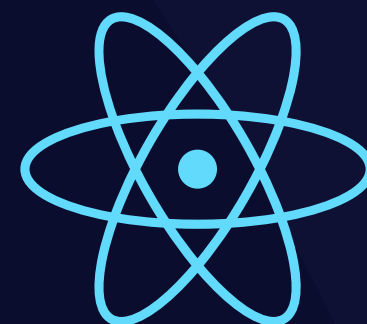
Essa análise precisa ser feita com foco na aplicação.

O fato de um componente ter parte de sua estrutura diferente não significa necessariamente a separação.

Exemplo:

Um botão, que ora terá o texto Cancelar, ora o texto Salvar, não precisa ser dividido em dois botões.

# React JS



## OBRIGADO E BONS ESTUDOS!

PROF. SINAIDE BEZERRA

CURSOS: ADS/SI