



Serviço Nacional de Aprendizagem Industrial

PELO FUTURO DO TRABALHO



React - Desenvolvendo com Java Script IV

Rafael C. Ventura

O que iremos aprender?

- Manipular e transformar arrays de objetos
- Filtrar itens
- Renderização condicional

Nossas referências

- **W3 schools** (<https://www.w3schools.com>)
- **Mozilla** (<https://developer.mozilla.org/pt-BR/docs/Learn>)
- **React** (<https://create-react-app.dev>)

Material de Apoio

- Material de apoio: Prop Driling
- Arquivos do projeto (Css)

Projeto Desafio: Organograma interativo

- Iremos criar um formulário que cadastra colaboradores e/ou alunos separando em times de forma interativa

ORGANIZATIONAL CHART

INFOGRAPHIC



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec eu cursus turpis. Ut diam erat, ultrices vulputate placerat ut,

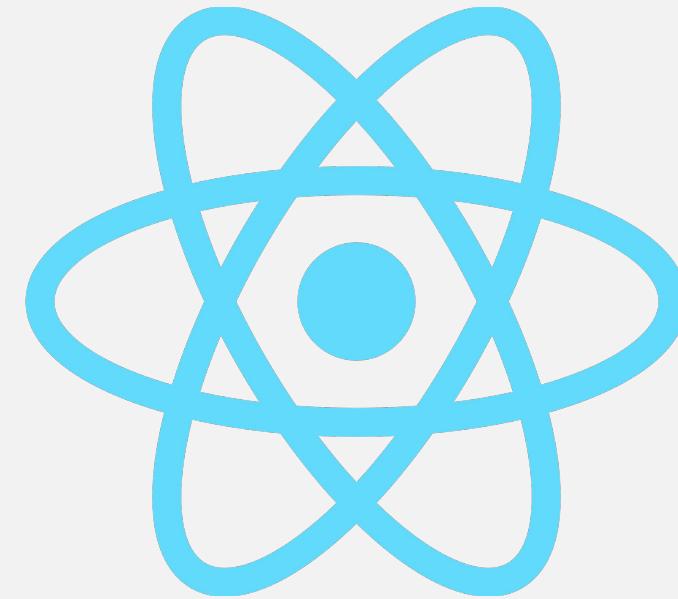
Vamos agora montar os times ...



Precisamos de um novo componente ...

**Este componente precisa receber um time
de colaboradores e ainda novos
colaboradores ...**

Criando o componente time



ORGANIZATIONAL CHART

INFOGRAPHIC



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec eu cursus turpis. Ut diam erat, ultrices vulputate placerat ut,



Então crie uma pasta Time

Depois crie um arquivo index.js

Crie um arquivo Time.css

Crie um jsx do componente usando uma <section>

Passe um nome dinâmico para um <h3>

<https://github.com/rc-ventura/react-senai>



Renderize o número de times no app.js

Analise no devTools ...

The screenshot shows the Visual Studio Code interface with the following details:

- Explorer View:** On the left, the file structure is displayed. A red box highlights the "Time" folder under "src/Componentes". Inside "Time", there are two files: "index.js" and "# Time.css".
- Code Editor:** The main area shows the content of "index.js". A red box highlights the entire code block:

```
1 import './Time.css'  
2  
3 const Time = () => {  
4     return (  
5           
6     )  
7 }  
8  
9 export default Time;
```
- Status Bar:** At the bottom, the status bar displays: "LN 9, COL 21", "SPACES: 4", "UTF-8", "CRLF", "{} JavaScript", "AWS: profile:terraform", and "Quokka".

The screenshot shows the Visual Studio Code interface with the following details:

- Explorer Bar:** On the left, it shows a tree view of the project structure under "FORMS-INTERATIVO". The "Time" folder is expanded, showing files like "index.js" and "# Time.css". Other collapsed categories include "node_modules" and "public".
- Code Editor:** The main area displays the content of "index.js".

```
1 import './Time.css'
2
3 const Time = (props) => {
4     return (
5         <section className='time'>
6             <h3>{props.nome}</h3>
7         </section>
8     )
9 }
10
11 }
12
13 export default Time;
```
- Decorations:** A large red rectangular box highlights the entire block of code from line 3 to line 10, which defines the "Time" component's structure.
- Status Bar:** At the bottom, it shows file statistics ("0 0 0"), AWS profile information ("AWS: profile:terraform"), and the Quokka extension status.
- Bottom Right:** Includes status icons for line count (Ln 13, Col 21), spaces (Spaces: 4), encoding (UTF-8 CRLF), file type (JavaScript), and live preview (Go Live).

EXPLORER ... JS index.js ...\\Formulario # Formulario.css JS index.js ...\\Time JS App.js X # Time.css ...

src > JS App.js > App

```
6 function App() {  
7  
8     const [colaboradores, setColaboradores] = useState([]);  
9  
10    const aoNovoColaboradorAdicionado = (colaborador) => {  
11        console.log(colaborador);  
12        setColaboradores([...colaboradores, colaborador]);  
13    }  
14  
15    return (  
16        <div className="App">  
17            <Banner/>  
18            <Formulario aoColaboradorCadastrado={colaborador => aoNovoColaboradorAdicionado(colaborador)}>  
19                <Time nome="Pedagógica"/>  
20                <Time nome="Administrativa"/>  
21                <Time nome="Pedagógica"/>  
22                <Time nome="Instrutores"/>  
23                <Time nome="Discentes"/>  
24            </Formulario>  
25        </div>  
26    );  
27}  
28  
29 }  
30  
31 export default App;
```

x +

localhost:3000

Digite seu nome

Cargo

Digite seu cargo

Imagen

Digite o endereço da sua imagem

Time

Pedagógica

Criar Card

Pedagógica

Administrativa

Pedagógica

Instrutores

Discentes

Aplicando o Css para o Time



No Ava temos na aula 08 o arquivo Css ...



Novamente precisaremos elevar o estado do componente Time



Agora temos um componente Time.

Index

Ainda melhor se o App.js pudesse controlar a lista de times

TIME

Faz sentido associarmos a lista de times ao componente Time

App

Banner

Formulario

A lista de times está no Formulário

CampoTexto

Botao

ListaSuspensa

Mas agora a nossa lista de times não será uma string e sim um objeto.

Iremos criar variáveis (chave) passando estilizações

Pois além dos cards precisamos passar uma cor também

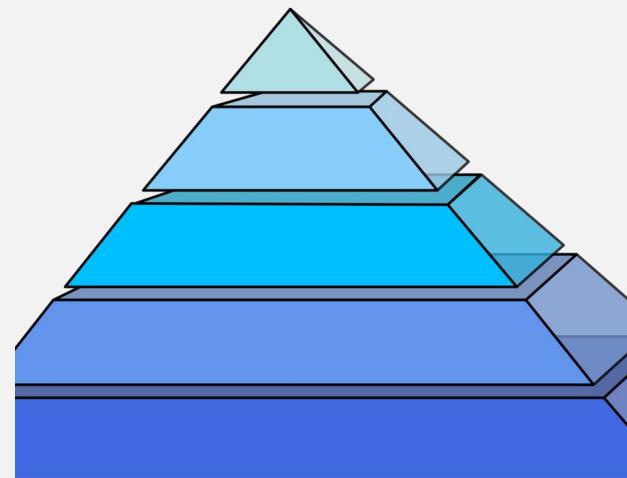
ORGANIZATIONAL CHART

INFOGRAPHIC



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec eu cursus turpis. Ut diam erat, ultrices vulputate placerat ut,

então vamos elevar o estado novamente ...





Então vá no App.js e crie um array de objetos times

defina uma variável (chave): nome

defina mais uma variável (chave); corDeFundo ;

crie 5 objetos no array e dê o nome de cada time.

JS index.js JS App.js X

src > JS App.js > App > times > nome

```
8 const times = [
9   {
10    nome: '',
11    corDeFundo: '',
12  },
13  {
14    nome: '',
15    corDeFundo: '',
16  },
17  {
18    nome: '',
19    corDeFundo: '',
20  },
21  {
22    nome: '',
23    corDeFundo: '',
24  },
25  {
26    nome: '',
27    corDeFundo: '',
28  },
29 ],
30 ]
```

const [colaboradores, setColaboradores] = useState()

JS index.js ...\\Formulario X JS index.js ...\\Time JS index.js src

src > Componentes > Formulario > JS index.js > Formulario > times

```
1 import { useState } from 'react'
2 import Botao from '../Botao'
3 import CampoTexto from '../CampoTexto'
4 import ListaSuspensa from '../ListaSuspensa'
5 import './Formulario.css'
6
7 export const Formulario = (props) => {
8
9   const times = [
10    'Pedagógica',
11    'Admnistrativa',
12    'Instrutores',
13    'Coordenação',
14    'Discentes'
15  ]
16
17
18  const [nome, setNome] = useState('');
19  const [cargo, setCargo] = useState('');
20  const [imagem, setImagem] = useState('');
21  const [time, setTime] = useState('');
22
23
24  const aoEnviar = (evento) => {
25    evento.preventDefault();
```



Agora iremos alternar o valor da variável corDeFundo

Alterne seu valor para:

CorDeFundo: '#5b9bd5 ; '#e7e6e6'

JS index.js JS App.js X ... JS index.js ...\\Formulario X # Time.css JS index.js ...\\Time ...

src > JS App.js > App > times > corDeFundo

```
8 const times = [
9   {
10     nome: 'Pedagógica',
11     corDeFundo: '#5b9bd5',
12   },
13   {
14     nome: 'Administrativa',
15     corDeFundo: '#e7e6e6',
16   },
17   {
18     nome: 'Instrutores',
19     corDeFundo: '#5b9bd5',
20   },
21   {
22     nome: 'Coordenação',
23     corDeFundo: '#e7e6e6',
24   },
25   {
26     nome: 'Discentes',
27     corDeFundo: '#5b9bd5',
28   },
29 ],
30 ]
31
32 const [colaboradores, setColaboradores] = useState()
```

src > Componentes > Formulario > JS index.js > Formulario > times

```
1 import { useState } from 'react'
2 import Botao from '../Botao'
3 import CampoTexto from '../CampoTexto'
4 import ListaSuspensa from '../ListaSuspensa'
5 import './Formulario.css'
6
7 export const Formulario = (props) => {
8
9   const times = [
10     'Pedagógica',
11     'Administrativa',
12     'Instrutores',
13     'Coordenação',
14     'Discentes'
15   ]
16
17
18   const [nome, setNome] = useState('');
19   const [cargo, setCargo] = useState('');
20   const [imagem, setImagem] = useState('');
21   const [time, setTime] = useState('');
22
23
24   const aoEnviar = (evento) => {
25     evento.preventDefault();
```

**Agora precisamos renderizar o componente
Time passando uma lista de times ...**



Apague todos os componentes time no App.js



Agora precisamos fazer um map de times iterando um time que renderiza o componente Time

Ainda precisamos de uma props nome e corDoFundo de forma dinâmica.

Não esqueça de passar a key

JS index.js JS App.js X

src > JS App.js > App

```
34
35
36     const aoNovoColaboradorAdicionado = (colaborador) => {
37         console.log(colaborador);
38         setColaboradores([...colaboradores, colaborador]);
39     }
40
41     return (
42         <div className="App">
43             <Banner/>
44             <Formulario aoColaboradorCadastrado={colaborador => aoNovoColaboradorAdicionado(colaborador)} />
45             {times.map(time => <Time key={time.nome} nome={time.nome}>/)}
46         </div>
47     );
48 }
49
50
51 export default App;
52
```

...



Analise o dev Tools ...

Administrativa

Instrutores

Coordenação

Discentes



Agora precisamos também passar uma props com a variável corDoFundo do objeto time.

```
JS index.js JS App.js X
src > JS App.js > App
35
36     const aoNovoColaboradorAdicionado = (colaborador) => {
37         console.log(colaborador);
38         setColaboradores([...colaboradores, colaborador]);
39     }
40
41
42     return (
43         <div className="App">
44             <Banner/>
45             <Formulario aoColaboradorCadastrado={colaborador => aoNovoColaboradorAdicionado(colaborador)} />
46             {times.map(time => <Time key={time.nome} nome={time.nome} corDoFundo={time.corDeFundo}/>)
47         </div>
48     );
49 }
50
51 export default App;
52
```

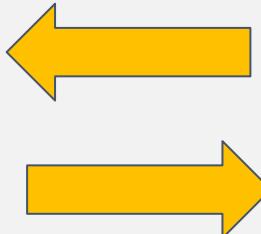


**Vamos aprender algo novo, vamos pegar uma props
CorDoFundo e estilizar um componente fazendo um
CSS INLINE**

props = nome = string

props = CorDoFundo = hexadecimal

TIME



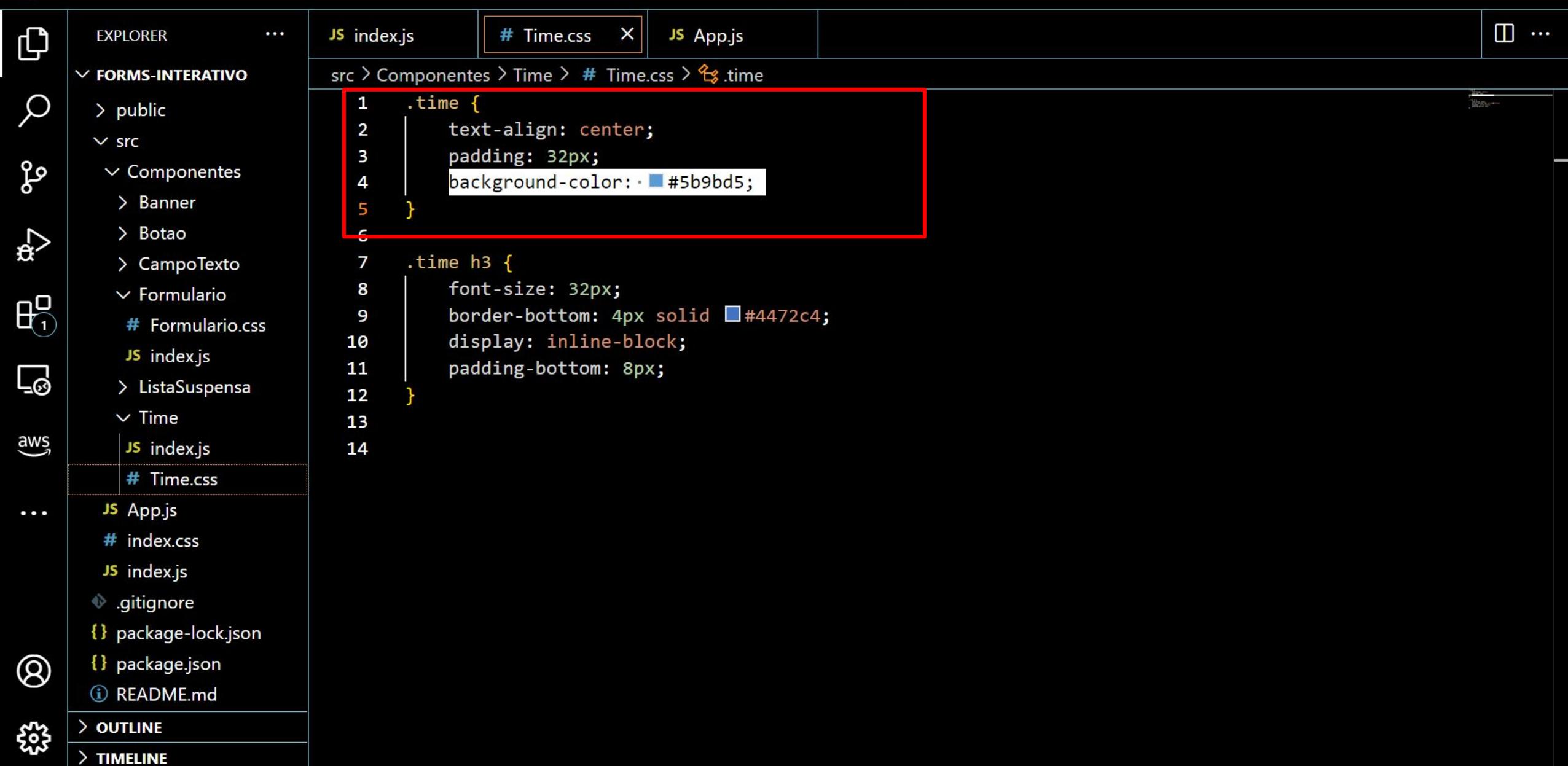
App

Estilizar o componente

CSS IN-LINE



Primeiro vamos no Time.Css apagar o background color do componente Time ...



The screenshot shows the Visual Studio Code interface with the following details:

- Explorer Bar:** On the left, it displays the project structure under "FORMS-INTERATIVO". The "src" folder contains "Componentes" (Banner, Botao, CampoTexto), "Formulario" (Formulario.css, index.js), "ListaSuspensa", "Time" (index.js, Time.css), "App.js", "index.css", "index.js", ".gitignore", "package-lock.json", "package.json", and "README.md".
- Editor:** The main area shows the "Time.css" file with the following content:

```
1 .time {  
2     text-align: center;  
3     padding: 32px;  
4     background-color: #5b9bd5;  
5 }  
  
6  
7 .time h3 {  
8     font-size: 32px;  
9     border-bottom: 4px solid #4472c4;  
10    display: inline-block;  
11    padding-bottom: 8px;  
12 }  
13  
14
```
- Status Bar:** At the bottom, it shows "Ln 5, Col 1 (28 selected)" and other settings like "Spaces: 4", "UTF-8", "CRLF", "CSS", "Go Live", and icons for AWS, Quokka, and timeline.

```
<section className='time' Style={variavel-react} >
```

```
<section className='time' Style={{objeto-javaScript}} >
```

```
<section className='time' Style={{backgroundColor: props.corDeFundo}} >
```

```
<section className='time' Style={variavel-react} >
```

```
<section className='time' Style={{objeto-javaScript}} >
```

```
const css = {backgroundColor: props.corDoFundo}
```

```
return (
```

```
<section className='time' Style={{css}} >
```

EXPLORER ...

FORMS-INTERATIVO

- > public
- src
 - Componentes
 - > Banner
 - > Botao
 - > CampoTexto
 - Formulario
 - # Formulario.css
 - JS index.js
 - > ListaSuspensa
 - Time
 - JS index.js
 - # Time.css
- ...
 - JS App.js
 - # index.css
 - JS index.js
 - .gitignore
 - { } package-lock.json
 - { } package.json
 - README.md
- > OUTLINE
- > TIMELINE

JS index.js ...\\Time X JS index.js src # Time.css JS App.js

src > Componentes > Time > JS index.js > [?] Time

```
1 import './Time.css'
2
3
4 const Time = (props) => {
5
6     const css = {backgroundColor: props.corDeFundo}
7
8
9     return (
10        <section className='time' style={css}>
11            <h3>{props.nome}</h3>
12        </section>
13    )
14}
15
16
17
18 export default Time;
```

Vamos lá no Dev Tools ...

Pedagógica

Administrativa

Instrutores

Coordenação

Discentes

Refatorando o código ...

Apague a lista de times no Formulário



Agora no App.js tente agora passar a lista de times para o componente Formulário.

Use o Map.

Lembre que queremos apenas o nome do objeto time

File Edit Selection View Go Run Terminal Help index.js - forms-interativo - Visual Studio Code

EXPLORER ... JS index.js ...\\Time JS index.js ...\\Formulario X # Botao.css JS index.js ...\\ListaSuspensa # Time.css JS App.js

FORMS-INTERATIVO

> node_modules
> public
src
 Componentes
 > Banner
 Formulario
 # Botao.css
 JS index.js
 > CampoTexto
 Formulario
 # Formulario.css
 JS index.js
 ListaSuspensa
 JS index.js
 # ListaSuspensa.css
 Time
 JS index.js
 # Time.css
 JS App.js
 # index.css
 JS index.js
 .gitignore
 package-lock.json
 package.json
 README.md

src > Componentes > Formulario > JS index.js > [x] Formulario > [x] times

```
1 import { useState } from 'react'
2 import Botao from '../Botao'
3 import CampoTexto from '../CampoTexto'
4 import ListaSuspensa from '../ListaSuspensa'
5 import './Formulario.css'
6
7 export const Formulario = (props) => {
8   const times = [
9     'Pedagógica',
10    'Administrativa',
11    'Instrutores',
12    'Coordenação',
13    'Discentes'
14  ]
15
16  .....
17
18  const [nome, setNome] = useState('');
19  const [cargo, setCargo] = useState('');
20  const [imagem, setImagen] = useState('');
21  const [time, setTime] = useState('');
22
23
24  const aoEnviar = (evento) => {
25    evento.preventDefault();
26    props.aoColaboradorCadastrado({
27      nome,
28      cargo,
29      imagem,
30      time
31    })
32  }
33
34
35
36  return (
37    <section className='formulario'>
38      <form onSubmit={aoEnviar}>
39        <h2>Preencha os dados para criar o card do colaborador:</h2>
40        <CampoTexto>
```

Deletar

File Edit Selection View Go Run Terminal Help App.js - forms-interativo - Visual Studio Code

EXPLORER ... JS index.js ...\\Time JS index.js ...\\Formulario JS index.js src # Botao.css # Time.css JS App.js X

src > JS App.js > ...

```
27 |     nome: 'Discentes',
28 |     corDeFundo: '#5b9bd5',
29 |   },
30 | ]
31 |
32 | const [colaboradores, setColaboradores] = useState([]);
33 |
34 |
35 |
36 | const aoNovoColaboradorAdicionado = (colaborador) => {
37 |   console.log(colaborador);
38 |   setColaboradores([...colaboradores, colaborador]);
39 |
40 |
41 | return (
42 |   <div className="App">
43 |     <Banner/>
44 |     <Formulario times={times.map(time => time.nome)} aoColaboradorCadastrado={colaborador => aoNovoColaboradorAdicionado(colaborador)}/>
45 |     {times.map(time => <time key={time.nome} nome={time.nome} corDeFundo={time.corDeFundo}>)}
46 |   </div>
47 |
48 | );
49 |
50 |
51 | export default App;
52 | 
```

aws

Y

gitpod

JS App.js
index.css
JS index.js
.gitignore
{ package-lock.json
{ package.json
① README.md

OUTLINE
TIMELINE

Ln 52, Col 1 Spaces: 2 UTF-8 LF {} JavaScript ⚡ Go Live 🔍

```
<CampoTexto  
eObrigatorio={true}  
label='Cargo'  
placeholder='Digite seu cargo'  
valor={cargo}  
aoAlterar={valor => setCargo(valor)}  
/>  
  
<CampoTexto  
label='Imagen'  
placeholder='Digite o endereço da sua imagem'  
valor={imagem}  
aoAlterar={valor => setImagen(valor)}  
/>  
  
<ListaSuspensa  
label = 'Time'  
|   itens={props.times} highlight  
|   valor={time}  
|   aoAlterar={valor => setTime(valor)}  
|  
|   />  
<Botao>  
|   Criar Card  
</Botao>  
</form>  
</section>  
}
```

Criando o componente Card ...

ORGANIZATIONAL CHART

INFOGRAPHIC



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec eu cursus turpis. Ut diam erat, ultrices vulputate placerat ut,



Crie a pasta Colaborador ...

Dentro da pasta, crie um index.js e um arquivo Colaborador.css

Crie três blocos de div: container; outro com um e o outro com <h4> e <h5>

Renderize o Card ??? Onde iremos renderizá-lo?

EXPLORER ... JS index.js ...\\Time JS index.js ...\\Formulario JS index.js ...\\Colaborador X # Colaborador.css JS index.js src # Botao.css # Time.css J ⋮

src > Componentes > Colaborador > JS index.js > [e] Colaborador

```
1 import './Colaborador.css'
2
3 const Colaborador = () => {
4   return (
5     <div>
6       <div>
7         <img src='https://github.com/rc-ventura.png' alt='foto do colaborador' />
8       </div>
9       <div>
10        <h4>Rafael Ventura</h4>
11        <h5>Instrutor</h5>
12      </div>
13    </div>
14  )
15
16 }
17
18
19
20 export default Colaborador
```

EXPLORER ... JS index.js ...\\Time X JS index.js ...\\Formulario JS index.js ...\\Colaborador # Colaborador.css JS index.js src # Botao.css # Time.css J ⋮

src > Componentes > Time > JS index.js > [e] Time

```
1 import Colaborador from '../Colaborador';
2 import './Time.css'
3
4
5 const Time = (props) => {
6
7     const css = {backgroundColor: props.corDeFundo}
8
9
10    return (
11        <section className='time' style={css}>
12            <h3>{props.nome}</h3>
13            <Colaborador/>
14        </section>
15    )
16 }
17
18
19 export default Time;
```

Pedagógica



Rafael Ventura
Instrutor

Agora vamos aplicar o Css para o componente Colaborador



Pegue o arquivo Css no Ava e copie dentro do Colaborador.Css

Coloque o ClassName para as três <div>:

- **colaborador**
- **cabecalho**
- **rodape**

EXPLORER ... JS index.js ...\\Time JS index.js ...\\Formulario JS index.js ...\\Colaborador X # Colaborador.css JS index.js src # Botao.css # Time.css J ...

src > Componentes > Colaborador > JS index.js > [e] Colaborador

```
1 import './Colaborador.css'
2
3 const Colaborador = () => {
4   return (
5     <div className='colaborador'>
6       <div className='cabecalho'>
7         <img src='https://github.com/rc-ventura.png' alt='foto do colaborador' />
8       </div>
9       <div className='rodape'>
10        <n4>Katael Ventura</n4>
11        <h5>Instrutor</h5>
12      </div>
13    </div>
14
15  )
16}
17
18
19
20 export default Colaborador
```

EXPLORER ...

FORMS-INTERATIVO

- > node_modules
- > public
- src
 - Componentes
 - > Banner
 - Botao
 - # Botao.css
 - JS index.js
 - > CampoTexto
 - Colaborador
 - # Colaborador.css
 - JS index.js
 - Formulario
 - # Formulario.css
 - JS index.js
 - ListaSuspensa
 - JS index.js
 - # ListaSuspensa.css
 - Time
 - JS index.js
 - # Time.css
 - App.js
 - # index.css
 - JS index.js
 - .gitignore
 - { package-lock.json
 - { package.json
 - ① README.md

```
JS index.js ...\Time JS index.js ...\Formulario JS index.js ...\Colaborador # Colaborador.css X JS index.js src # Botao.css # Time.css JS App.js
```

src > Componentes > Colaborador > # Colaborador.css > .colaborador img

```
1 .colaborador {  
2 |   width: 280px;  
3 }  
4  
5 .colaborador .cabecalho {  
6 |   background-color: #F0F0F0;  
7 |   border-radius: 10px 10px 0px 0px;  
8 }  
9 .colaborador img {  
10 |   width: 100px;  
11 |   border-radius: 50%;  
12 |   position: relative;  
13 |   bottom: -50px;  
14 }  
15  
16 .colaborador .rodape {  
17 |   background: #FFFFFF;  
18 |   box-shadow: 5px 5px 15px rgba(0, 0, 0, 0.08);  
19 |   border-radius: 0px 0px 10px 10px;  
20 |   padding-top: 90px;  
21 |   padding-bottom: 40px;  
22 }  
23  
24 .colaborador .rodape h4 {  
25 |   color: #6278F7;  
26 |   font-size: 18px;  
27 |   line-height: 22px;  
28 |   font-weight: bold;  
29 |   margin-bottom: 8px;  
30 }  
31  
32 .colaborador .rodape h5 {  
33 |   font-size: 18px;  
34 |   line-height: 22px;  
35 |   color: #212121;  
36 |   padding: 0 16px;  
37 }  
38
```



localhost:3001

Pedagógica



Rafael Ventura

Instrutor

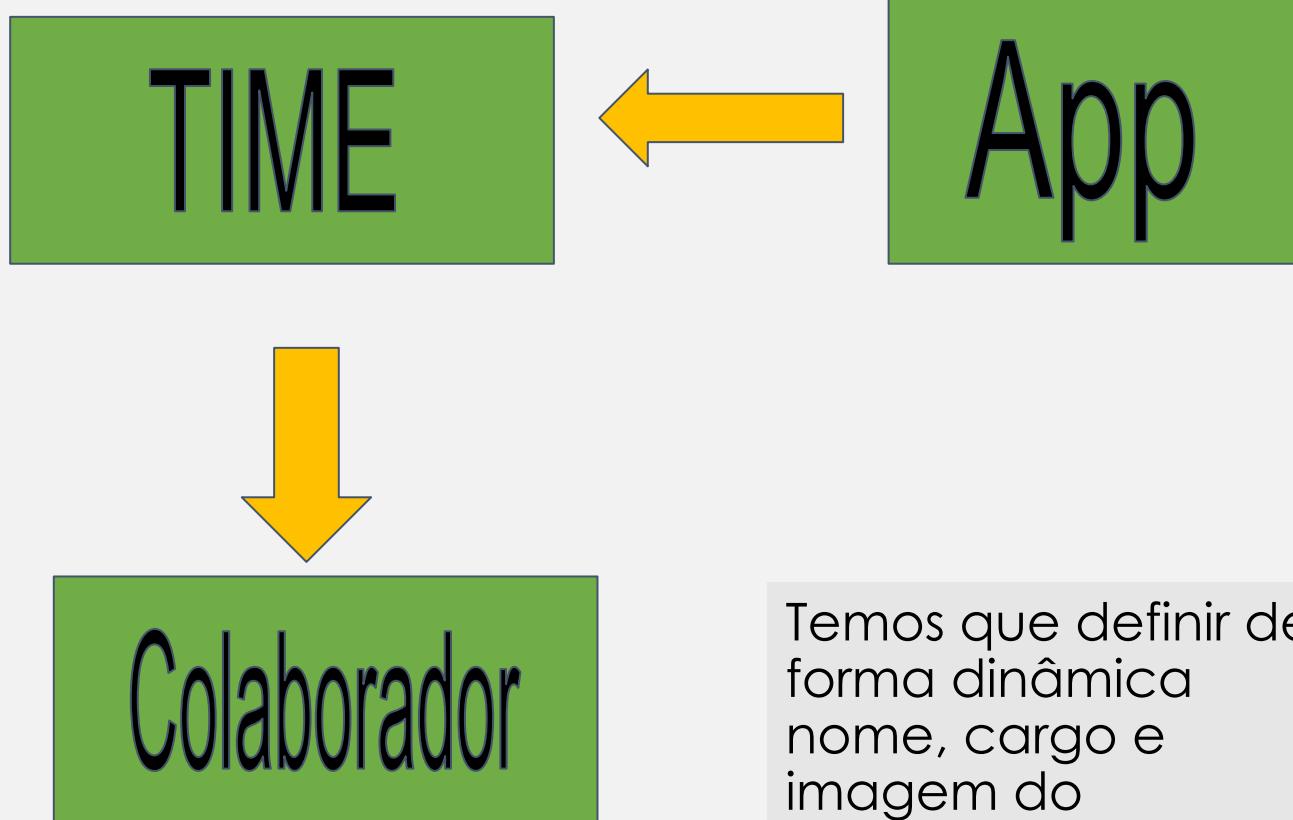
Administrativa



Rafael Ventura

Instrutor

Temos que renderizar
cada card do
colaborador de
forma dinâmica



Precisamos no App.js
passar o array de
colaboradores
cadastrados pelo
formulário

Temos que definir de
forma dinâmica
nome, cargo e
imagem do
colaborador



Vamos passar o array de Colaboradores para o Time



EXPLORER ... JS index.js ...\\Formulario JS index.js ...\\Colaborador # Colaborador.css JS index.js src # Botao.css # Time.css JS App.js X ⏹ ...

src > JS App.js > App

```
27 |     nome: 'Discentes',
28 |     corDeFundo: '#5b9bd5',
29 |
30 ]
31
32 const [colaboradores, setColaboradores] = useState([]);
33
34
35
36 const aoNovoColaboradorAdicionado = (colaborador) => {
37     console.log(colaborador);
38     setColaboradores([...colaboradores, colaborador]);
39 }
40
41
42 return (
43     <div className="App">
44         <Banner/>
45         <Formulario times={times.map(time => time.nome)} aoColaboradorCadastrado={colaborador => aoNovoColaboradorAdicionado} colaboradores={colaboradores}>
46             {times.map(time => <Time colaboradores={colaboradores}>
47                 key={time.nome}
48                 nome={time.nome}
49                 corDeFundo={time.corDeFundo}/>)
50
51             </div>
52         );
53     }
54 }
55
56 export default App;
```

Agora mexa no componente Time para que possamos renderizar vários Cards de Colaboradores de forma dinâmica.



Lembre-se use a função map.

Não esqueça de definir o valor das props no componente Time.



EXPLORER ... JS index.js ...\\Time X JS index.js ...\\ListaSuspensa JS index.js ...\\Formulario JS index.js ...\\Colaborador # Colaborador.css # Botao.css # Ti ...

src > Componentes > Time > JS index.js > [?] default

```
1 import Colaborador from '../Colaborador';
2 import './Time.css'
3
4
5 const Time = (props) => {
6     const css = {backgroundColor: props.corDeFundo}
7
8     return (
9         <section className='time' style={css}>
10            <h3>{props.nome}</h3>
11            {props.colaboradores.map(colaborador => <Colaborador/>)}
12        </section>
13    )
14 }
15
16
17 export default Time;
```

The code block shows a component named 'Time' that imports 'Colaborador' and 'Time.css'. It defines a function that takes 'props' and returns a section with a background color. Inside the section, it displays the 'nome' prop and a list of 'colaboradores' using a map function. The line 'props.colaboradores.map(colaborador => <Colaborador/>)' is highlighted with a red rectangle.

Precisamos agora refatorar componente Colaborador



Tente deixá-lo dinâmico, passando props.

Não esqueça de definir o valor das props no componente Time.



JS index.js ...\\Time

JS index.js ...\\ListaSuspensa

JS index.js ...\\Formulario

JS index.js ...\\Colaborador X

Colaborador.css

Botao.css

Time.css

JS App.js

...

src > Componentes > Colaborador > JS index.js > [x] Colaborador

```
1 import './Colaborador.css'  
2  
3 const Colaborador = (props) => {  
4     return (  
5         <div className='colaborador'>  
6             <div className='cabecalho'>  
7                 <img src={props.imagem} alt={props.nome} />  
8             </div>  
9             <div className='rodape'>  
10                <h4>{props.nome}</h4>  
11                <h5>{props.cargo}</h5>  
12            </div>  
13        </div>  
14    )  
15}
```

JS index.js X

src > Componentes > Time > JS index.js > ...

```
1 import Colaborador from '../Colaborador';  
2 import './Time.css'  
3  
4  
5 const Time = (props) => {  
6     const css = {backgroundColor: props.corDeFundo}  
7  
8     return (  
9         <section className='time' style={css}>  
10            <h3>{props.nome}</h3>  
11            [<Colaborador nome={colaborador.nome} imagem={colaborador.imagem} cargo={colaborador.cargo}/>]  
12        </section>  
13    )  
14}  
15
```



Ln 4, Col 1 Spaces: 4 UTF-8 CRLF {} JavaScript ⚡ Go Live ⚡ 🔍

Pedagógica



Mario Souto

Professor



Rafael Ventura

Professor



Serviço Nacional de Aprendizagem Industrial
PELO FUTURO DO TRABALHO

0800 048 1212 sc.senai.br

Rodovia Admar Gonzaga, 2765 - Itacorubi - 88034-001 - Florianópolis, SC