

Projeto UA – Team

Um breve resumo da aplicação:

1. **App:** Este é o componente principal do aplicativo. Ele mantém o estado de todos os times e colaboradores. Ele também fornece funções para adicionar e remover colaboradores e times.
2. **Banner:** Este componente renderiza um banner no topo da página.
3. **Formulario:** Este componente renderiza dois formulários - um para adicionar um novo colaborador e outro para adicionar um novo time. Ele recebe funções do componente `App` para atualizar o estado quando um novo colaborador ou time é adicionado.
4. **Campo:** Este é um componente reutilizável que renderiza um campo de entrada. Ele é usado no componente `Formulario`.
5. **ListaSuspensa:** Este é um componente reutilizável que renderiza uma lista suspensa. Ele é usado no componente `Formulario` para selecionar um time ao adicionar um novo colaborador.
6. **Botao:** Este é um componente reutilizável que renderiza um botão. Ele é usado no componente `Formulario`.
7. **Time:** Este componente renderiza um time e todos os seus colaboradores. Ele recebe um time e uma lista de colaboradores desse time como props. Ele também recebe funções para deletar um colaborador e mudar a cor do time.
8. **Colaborador:** Este componente renderiza um colaborador. Ele recebe um colaborador como prop e uma função para deletar o colaborador.
9. **Rodape:** Este componente renderiza um rodapé na parte inferior da página.

O aplicativo permite que você adicione times com uma cor específica. Para cada time, você pode adicionar colaboradores com um nome, cargo e imagem. Você pode deletar colaboradores e mudar a cor de um time. O aplicativo organiza os colaboradores por time e exibe-os em cartões coloridos.

Aqui está uma visão geral do que cada componente faz:

```
import { useState } from "react";
import Banner from "../componentes/Banner";
import Formulario from "../componentes/Formulario";
import Rodape from "../componentes/Rodape";
import Time from "../componentes/Time";
import { v4 as uuidv4 } from 'uuid';
```

Nesta parte do código, você está importando várias bibliotecas e componentes. `useState` é um Hook do React que permite adicionar o estado do React a componentes de função. `Banner`, `Formulario`, `Rodape` e `Time` são componentes React que você criou e estão sendo importados para serem usados neste arquivo. `uuidv4` é uma função da biblioteca 'uuid' que gera um ID único.

```
function App() {
  const [times, setTimes] = useState([
    // array de times
  ]);
  const [colaboradores, setColaboradores] = useState(inicial)
```

Aqui você está declarando a função `App` que é o componente principal do seu aplicativo React. Dentro desta função, você está usando o Hook `useState` para criar estados para `times` e `colaboradores`. `times` é um array de objetos onde cada objeto representa um time. `colaboradores` é um array de objetos onde cada objeto representa um colaborador.

```
function deletarColaborador(id) {
  setColaboradores(colaboradores.filter(colaborador =>
    colaborador.id !== id));
}
```

Esta é uma função que permite deletar um colaborador. Ela recebe um `id` como parâmetro e atualiza o estado `colaboradores` para um novo array que não inclui o colaborador com o `id` fornecido.

```
function mudarCorDoTime(cor, id) {
  setTimes(times.map(time => {
    if (time.id === id) {
      time.cor = cor;
    }
    return time;
  }));
}
```

A função `mudarCorDoTime` recebe uma `cor` e um `id` como parâmetros. Ela atualiza o estado `times`, alterando a cor do time cujo `id` corresponde ao `id` fornecido.

```
function cadastrarTime(novoTime) {
  setTimes([...times, { ...novoTime, id: uuidv4() }])
}
```

A função `cadastrarTime` recebe um objeto `novoTime` como parâmetro e adiciona esse novo time ao estado `times`. O novo time é adicionado com um id único gerado pela função `uuidv4`.

```
function resolverFavorito(id) {
  setColaboradores(colaboradores.map(colaborador => {
    if (colaborador.id === id) colaborador.favorito =
!colaborador.favorito;
    return colaborador;
  })))
}
```

A função `resolverFavorito` recebe um `id` como parâmetro e atualiza o estado `colaboradores`, invertendo o valor da propriedade `favorito` do colaborador cujo `id` corresponde ao `id` fornecido.

```
return (
  <div>
    <Banner />
    <Formulario
      cadastrarTime={cadastrarTime}
      times={times.map(time => time.nome)}
      aoCadastrar={colaborador => setColaboradores([...colaboradores,
colaborador])}
    />
    <section className="times">
      <h1>Minha organização</h1>
      {times.map((time, indice) =>
        <Time
          aoFavoritar={resolverFavorito}
          mudarCor={mudarCorDoTime}
          key={indice}
          time={time}
          colaboradores={colaboradores.filter(colaborador =>
colaborador.time === time.nome)}
          aoDeletar={deletarColaborador}
        />)}
      </section>
      <Rodape />
    </div>
  );
```

Esta é a renderização do componente `App`. Ele retorna um elemento JSX que inclui os componentes `Banner`, `Formulario`, `Time` (para cada `time` em `times`) e `Rodape`. As funções `cadastrarTime`, `resolverFavorito`, `mudarCorDoTime` e `deletarColaborador` são passadas como props para os componentes filhos, permitindo que eles interajam com o estado do componente `App`.

```
export default App;
```

Finalmente, o componente `App` é exportado para ser usado em outros arquivos.

Componentes

Banner

```
import './banner.css'
```

Aqui, você está importando um arquivo CSS chamado `banner.css`. Este arquivo contém os estilos que serão aplicados ao componente `Banner`.

```
const Banner = () => {  
  return (<header className="cabecalho">  
    <img src='/imagens/banner.png' alt='Logo do Organo' />  
  </header>)  
}
```

Aqui, você está declarando uma função chamada `Banner` que retorna um elemento JSX. Este elemento é um cabeçalho (`<header>`) com a classe CSS `cabecalho`. Dentro do cabeçalho, há uma imagem (``) com o caminho do arquivo de imagem `/imagens/banner.png` e um texto alternativo `Logo do Organo`. O texto alternativo é usado por leitores de tela e em situações onde a imagem não pode ser carregada.

```
export default Banner
```

Finalmente, você está exportando o componente `Banner` para que ele possa ser importado e usado em outros arquivos.

Formulário

```
import { useState } from 'react'  
import Botao from '../Botao'  
import Campo from '../Campo'  
import ListaSuspensa from '../ListaSuspensa'  
import './formulario.css'
```

Aqui, você está importando várias bibliotecas e componentes. `useState` é um Hook do React que permite adicionar o estado do React a componentes de função. `Botao`, `Campo` e `ListaSuspensa` são componentes React que você criou e estão sendo importados para serem usados neste arquivo. `formulario.css` é um arquivo CSS que contém os estilos para este componente.

```
const Formulario = ({ aoCadastrar, times, cadastrarTime }) => {  
  const [nome, setNome] = useState('')  
  const [cargo, setCargo] = useState('')  
  const [imagem, setImagem] = useState('')  
  const [time, setTime] = useState('')
```

```
const [nomeTime, setNomeTime] = useState('')
const [corTime, setCorTime] = useState('')
```

Aqui, você está declarando a função `Formulario` que é um componente funcional do React. Este componente recebe três props: `aoCadastrar`, `times` e `cadastrarTime`. Dentro desta função, você está usando o Hook `useState` para criar estados para `nome`, `cargo`, `imagem`, `time`, `nomeTime` e `corTime`.

```
const aoSubmeter = (evento) => {
  evento.preventDefault()
  console.log('form enviado', nome, cargo, imagem, time)
  aoCadastrar({
    nome,
    cargo,
    imagem,
    time
  })
  setNome('')
  setCargo('')
  setImagem('')
  setTime('')
}
```

A função `aoSubmeter` é chamada quando o formulário é submetido. Ela impede o comportamento padrão do formulário (que é recarregar a página), imprime alguns valores no console para fins de depuração, chama a função `aoCadastrar` com um objeto contendo os valores atuais dos estados `nome`, `cargo`, `imagem` e `time`, e então limpa esses estados.

```
return (
  <section className="formulario-container">
    <form className="formulario" onSubmit={aoSubmeter}>
      <h2>Cadastre os dados do colaborador.</h2>
      <Campo
        obrigatorio
        label='Nome'
        placeholder='Digite seu nome '
        valor={nome}
        aoAlterado={valor => setNome(valor)} />
      <Campo
        obrigatorio
        label='Cargo'
        placeholder='Digite seu cargo '
        valor={cargo}
        aoAlterado={valor => setCargo(valor)} />
      <Campo
        label='Imagem'
        placeholder='Informe o endereço da imagem '
        valor={imagem}
        aoAlterado={valor => setImagem(valor)} />
      <ListaSuspensa
        obrigatorio
        label='Times'
```

```

        items={times}
        valor={time}
        aoAlterado={valor => setTime(valor)} />
      <Botao texto='Criar card' />
    </form>
    <form className="formulario" onSubmit={(evento) => {
      evento.preventDefault()
      cadastrarTime({
        nome: nomeTime,
        cor: corTime
      })
      setNomeTime('')
      setCorTime('')
    }}>
      <h2>Cadastre os dados do novo time.</h2>
      <Campo
        obrigatorio
        label='Nome'
        placeholder='Digite o nome do time '
        valor={nomeTime}
        aoAlterado={valor => setNomeTime(valor)} />
      <Campo
        obrigatorio
        type='color'
        label='Cor'
        placeholder='Digite a cor do time '
        valor={corTime}
        aoAlterado={valor => setCorTime(valor)} />
      <Botao texto='Criar time' />
    </form>
  </section>
)

```

Esta é a renderização do componente `Formulario`. Ele retorna um elemento JSX que inclui dois formulários. O primeiro formulário é para cadastrar um novo colaborador e o segundo é para cadastrar um novo time. Cada formulário contém vários componentes `Campo` para entrada de dados, um componente `ListaSuspensa` para selecionar um time, e um componente `Botao` para submeter o formulário.

```
export default Formulario
```

Finalmente, o componente `Formulario` é exportado para ser usado em outros arquivos.

Campo

```
import './campo.css'
```

Aqui, você está importando um arquivo CSS chamado `campo.css`. Este arquivo contém os estilos que serão aplicados ao componente `Campo`.

```

const Campo = ({ type = 'text', label, placeholder, valor, aoAlterado,
  obrigatorio = false }) => {
  return (<div className={`campo campo-${type}`}>

```

```

    <label>{label}</label>
    <input
      type={type}
      value={valor}
      onChange={evento => aoAlterado(evento.target.value)}
      required={obrigatorio}
      placeholder={placeholder}
    />
  </div>)
}

```

Aqui, você está declarando a função `Campo` que é um componente funcional do React. Este componente recebe várias props: `type`, `label`, `placeholder`, `valor`, `aoAlterado` e `obrigatorio`. Dentro desta função, você está retornando um elemento JSX que inclui uma `div` com uma classe CSS que depende do tipo do campo, e um elemento `input` que tem vários atributos definidos pelas props.

```
export default Campo
```

Finalmente, o componente `Campo` é exportado para ser usado em outros arquivos.

Botão

```
import './botao.css'
```

Aqui, você está importando um arquivo CSS chamado `botao.css`. Este arquivo contém os estilos que serão aplicados ao componente `Botao`.

```

const Botao = ({texto}) => {
  return <button className="botao">
    {texto}
  </button>
}

```

Aqui, você está declarando a função `Botao` que é um componente funcional do React. Este componente recebe uma prop: `texto`. Dentro desta função, você está retornando um elemento JSX que é um botão (`<button>`) com a classe CSS `botao`. O texto do botão é definido pela prop `texto`.

```
export default Botao
```

Finalmente, o componente `Botao` é exportado para ser usado em outros arquivos.

Lista Suspensa

```
import './lista-suspensa.css'
```

Aqui, você está importando um arquivo CSS chamado `lista-suspensa.css`. Este arquivo contém os estilos que serão aplicados ao componente `ListaSuspensa`.

```

const ListaSuspensa = ({label, items, valor, aoAlterado, obrigatorio = false}) => {
  return (<div className="lista-suspensa">
    <label>{label}</label>

```

```

        <select required={obrigatorio} value={valor} onChange={evento}
=> aoAlterado(evento.target.value)}>
            <option />
            {items.map(item => <option key={item}>{item}</option>)}
        </select>
    </div>
}

```

Aqui, você está declarando a função `ListaSuspensa` que é um componente funcional do React. Este componente recebe várias props: `label`, `items`, `valor`, `aoAlterado` e `obrigatorio`. Dentro desta função, você está retornando um elemento JSX que inclui uma div com a classe CSS `lista-suspensa`, um elemento `label`, e um elemento `select`. O elemento `select` tem vários atributos definidos pelas props e inclui uma lista de opções baseada na prop `items`.

Finalmente, o componente `ListaSuspensa` é exportado para ser usado em outros arquivos.

Colaborador

```

import { AiFillCloseCircle, AiFillHeart, AiOutlineHeart } from
"react-icons/ai";
import './colaborador.css'

```

Aqui, você está importando ícones da biblioteca `react-icons` e um arquivo CSS chamado `colaborador.css`. Este arquivo contém os estilos que serão aplicados ao componente `Colaborador`.

```

const Colaborador = ({ colaborador, corDeFundo, aoDeletar, aoFavoritar
}) => {
    function favoritar() {
        aoFavoritar(colaborador.id);
    }

    const propsfavorito = {
        size: 25,
        onClick: favoritar
    }
}

```

Aqui, você está declarando a função `Colaborador` que é um componente funcional do React. Este componente recebe várias props: `colaborador`, `corDeFundo`, `aoDeletar` e `aoFavoritar`. Dentro desta função, você está declarando uma função `favoritar` que chama a função `aoFavoritar` com o id do colaborador, e um objeto `propsfavorito` que define as propriedades do ícone de favorito.

```

return (<div className="colaborador">
    <AiFillCloseCircle
        size={25}
        className='deletar'
        onClick={() => aoDeletar(colaborador.id)}
    />

```



```

    <div className="cabecalho" style={{ backgroundColor: corDeFundo
}}>
      <img src={colaborador.imagem} alt={colaborador.nome} />
    </div>
    <div className="rodape">
      <h4>{colaborador.nome}</h4>
      <h5>{colaborador.cargo}</h5>
      <div className="favoritar">
        { colaborador.favorito ?
          <AiFillHeart {...propsfavorito} color="#ff0000"/>
          : <AiOutlineHeart {...propsfavorito} />
        }
      </div>
    </div>
  </div>
</div>)

```

Esta é a renderização do componente `Colaborador`. Ele retorna um elemento JSX que inclui um ícone de fechar que, quando clicado, chama a função `aoDeletar` com o id do colaborador, um cabeçalho com uma imagem do colaborador, e um rodapé com o nome e cargo do colaborador e um ícone de favorito que, quando clicado, chama a função `favoritar`.

```
export default Colaborador
```

Finalmente, o componente `Colaborador` é exportado para ser usado em outros arquivos.

Time

```

import Colaborador from '../Colaborador'
import './time.css'
import hexToRgba from 'hex-to-rgba';

```

Aqui, você está importando o componente `Colaborador`, um arquivo CSS chamado `time.css` que contém os estilos para o componente `Time`, e a função `hexToRgba` que converte cores em formato hexadecimal para o formato `rgba`.

```
const Time = ({ time, colaboradores, aoDeletar, mudarCor, aoFavoritar
}) => {
```

Aqui, você está declarando a função `Time` que é um componente funcional do React. Este componente recebe várias

`props: time, colaboradores, aoDeletar, mudarCor e aoFavoritar.`

```

return (
  <div>
    {colaboradores.length > 0 && <section className='time' style={{
      backgroundImage: 'url(/imagens/fundo.png)', backgroundColor:
      hexToRgba(time.cor, '0.6') }}>
      <input onChange={evento => mudarCor(evento.target.value,
      time.id)} value={time.cor} type='color' className='input-cor' />
      <h3 style={{ borderColor: time.cor }}>{time.nome}</h3>
      <div className='colaboradores'>
        {colaboradores.map((colaborador, indice) => {
          return (
            <Colaborador
              key={indice}

```

```

        colaborador={colaborador}
        corDeFundo={time.cor}
        aoDeletar={aoDeletar}
        aoFavoritar={aoFavoritar}
      />
    );
  }
}
</div>
</section>
)

```

Esta é a renderização do componente `Time`. Ele retorna um elemento JSX que inclui uma seção (`<section>`) com a classe CSS `time` e um estilo que define a imagem de fundo e a cor de fundo. Dentro desta seção, há um elemento `input` que permite ao usuário alterar a cor do time, um elemento `h3` que exibe o nome do time, e uma `div` que contém uma lista de componentes `Colaborador` para cada colaborador no time.

```
export default Time
```

Finalmente, o componente `Time` é exportado para ser usado em outros arquivos.

Rodape

```
import './rodape.css'
```

Aqui, você está importando um arquivo CSS chamado `rodape.css`. Este arquivo contém os estilos que serão aplicados ao componente `Rodape`.

```

const Rodape = () => {
  return (
    <footer className="footer">
      <section>
        <ul>
          <li>
            <a href="facebook.com" target="_blank">
              
            </a>
          </li>
          <li>
            <a href="twitter.com" target="_blank">
              
            </a>
          </li>
          <li>
            <a href="instagram.com" target="_blank">
              
            </a>
          </li>
        </ul>
      </section>
      <section>
        
      </section>
    </footer>
  )
}

```

```

    </section>
    <section>
      <p>
        Desenvolvido por Thiago e Heitor.
      </p>
    </section>
  </footer>)
}

```

Aqui, você está declarando a função `Rodape` que é um componente funcional do React. Este componente não recebe nenhuma prop. Dentro desta função, você está retornando um elemento JSX que é um rodapé (`<footer>`) com a classe CSS `footer`. Dentro do rodapé, há três seções (`<section>`):

- A primeira seção contém uma lista não ordenada (``) de links (`<a>`) para as páginas do Facebook, Twitter e Instagram. Cada link contém uma imagem (``) que provavelmente é o ícone da respectiva rede social.
- A segunda seção contém uma imagem (``) que provavelmente é o logotipo do site.
- A terceira seção contém um parágrafo (`<p>`) que indica que o site foi desenvolvido por Thiago e Heitor.

```
export default Rodape
```

Finalmente, o componente `Rodape` é exportado para ser usado em outros arquivos.