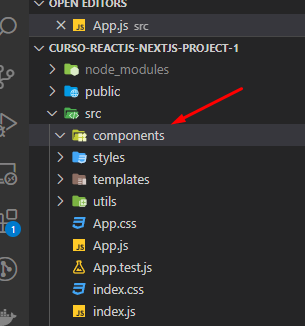
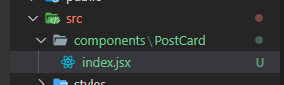
Props e organização dos componentes - Parte 1

Vamos organizar o projeto

Criar uma pasta chamada components dentro do src



Dentro vamos criar uma outra pasta com o nome do nosso componente (um dos padrões utilizados pelos devs) e um arquivo dentro chamado index.jsx padrão:



o index vai ficar assim:

export const PostCard = (props) => {

    return (

        <h1>Teste</h1>

    );

}

Quando formos exportar alguma coisa que não for default, como estamos fazendo, temos que tratar da seguinte maneira:

import {} from ''

“importe alguma coisa de algum lugar”. Depois de importar, vamos passar as props e pegar no index

se mandarmos o post todo assim:

 <section className='container'>

        <div className="posts">

          {posts.map(post => (

            <PostCard

              post={post}

            />

          ))

          }

        </div>

      </section>

no index teremos que pegar assim :

export const PostCard = ({post}) => {

    return (

        <div className="post">

            <img src={post.cover} alt={post.title} />

            <div key={post.id} className="post-content">

                <h1>{post.title}</h1>

                <p>{post.body}</p>

            </div>

        </div>

    );

}

podemos usar:

const post = props.post

que é igual a

export const PostCard = (props) => {

const { post } = props;

usando destructuring

ou ainda assim:

export const PostCard = ({post}) => {

mas podemos mandar os props separadamente que parece ser melhor maneira

 <div className="posts">

          {posts.map(post => (

              <PostCard

              key = {post.id}

              title={post.title}

              body={post.body}

              cover={post.cover}

              id={post.id}

              />

            ))

          }

        </div>

E PEGARMOS DIRETAMENTE AS PROPS com o nome e aplicar direto

export const PostCard = ({title,cover, body, id }) => {

    return (

        <div className="post">

            <img src={cover} alt={title} />

            <div key={id} className="post-content">

                <h1 >{title}</h1>

                <p>{body}</p>

            </div>

        </div>

    );

}

Como este componente não tem nada entre o início e o return, o ideal é reduzir assim:

export const PostCard = ({title,cover, body, id }) =>  (

        <div className="post">

            <img src={cover} alt={title} />

            <div key={id} className="post-content">

                <h1 >{title}</h1>

                <p>{body}</p>

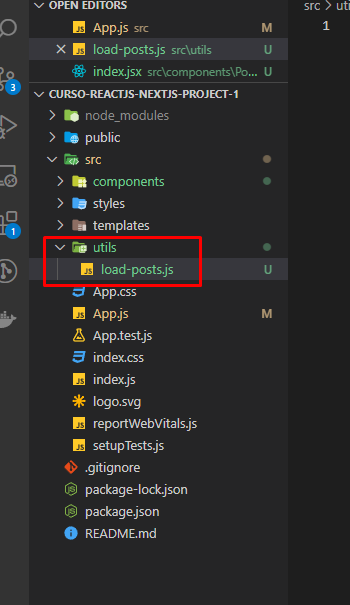
            </div>

        </div>

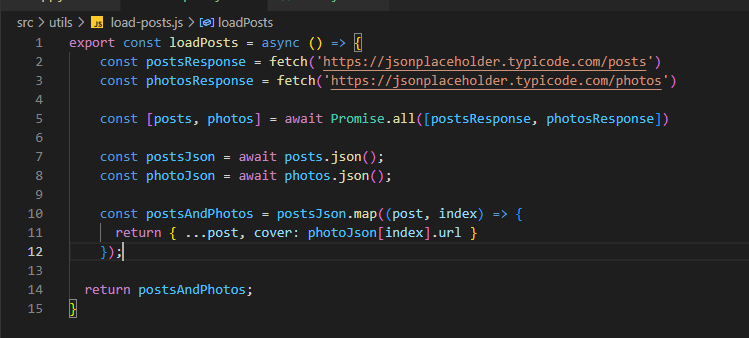
    );

- Continuando:

Vamos criar uma pasta chamada utils e dentro vamos criar um arquivo chamado load-posts



E vamos levar todo conteúdo da função loadPosts do nosso App.js para este arquivo:



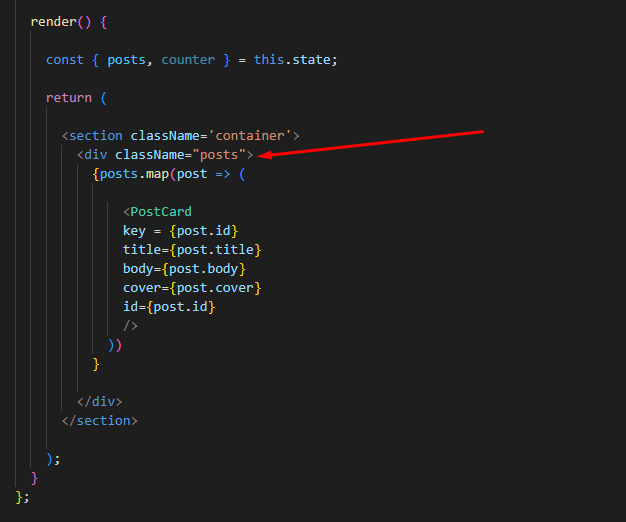
Depois vamos importar o load posts:

import {loadPosts} from './utils/load-posts'

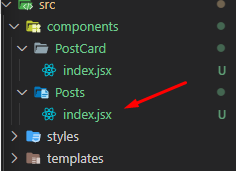
e chamar a função do import dentro da loadPosts local:



Em seguida, podemos limpar o render, criando mais um componente para a div posts:



Vamos criar a pasta Posts dentro da components com um index dentro, igual ao anterior:



E no return do App vamos chamar o Posts enviando posts como props:

 return (

      <section className='container'>

        <Posts  posts={ posts } />

      </section>

    );

Logicamente vamos ter que importar antes:

import { Posts } from './components/Posts';

Enfim no src\components\Posts\index.jsx vamos importar o postCard que agora não precisamos mais importar no App e colar o que tínhamos no return do app

import { PostCard } from '../PostCard';

export const Posts = ({ posts }) => (

    <div className="posts">

          {posts.map(post => (

              <PostCard

              key = {post.id}

              title={post.title}

              body={post.body}

              cover={post.cover}

              id={post.id}

              />

            ))

          }

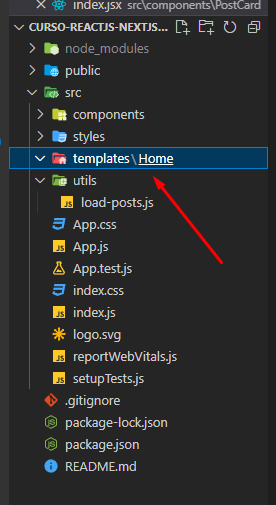
        </div>

);

Props e organização dos componentes - Parte 3

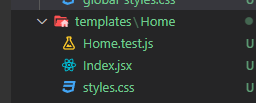
Vamos criar uma pasta chamada template e dentro vamos por as paginas.

Posteriormente não vamos precisar porque o Next.js cria uma pasta chamada pages.



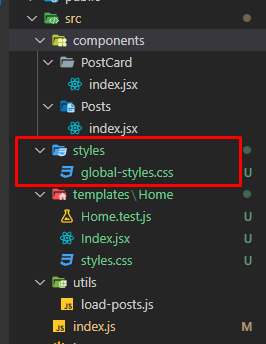
Arrastamos aos arquivos App pra dentro desta pasta (App.js, App.css e o App.test.js )

E vamos trocar seus nomes para:

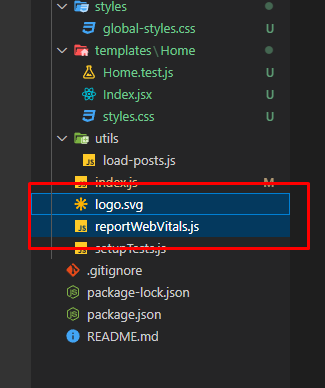


Sempre o componente vai se chamar index e o css styles dentro da pasta.

Vamos arrumar as importações e criar a pasta styles dentro de src e colocar o index.css global la e chamar de global-styles.css.



Vamos deletar o logo e o report:



Parei de anotar na aula Props e organização dos componentes - Parte 3

para ir mais rápido.