TUTORIAL - API GITHUB

SUMARIO

TUTORIAL - API GITHUB.....1
01 - OBJETIVO..... 5
02 - ENDPOINT DA API GITHUB..... 5
03 - INSTALAÇÃO DO NODEJS.....6
BAIXAR.....6
EXECUTAR.....6
TESTAR NO TERMINAL..... 10
04 - YARN..... 10
INSTALAR.....10
TESTAR..... 10
05 - GIT..... 11
BAIXAR..... 11
INSTALAR..... 11
TESTAR..... 12
06 - CRIAR O PROJETO API_GITHUB.....13
DEFINIR O DIRETÓRIO PARA O PROJETO.....13
CRIAR O PROJETO..... 13
ATUALIZAR..... 14
ABRIR COM O CODE..... 15
INICIAR O PROJETO..... 15
VISUALIZAÇÃO WEB.....16
ENXUGAR O PROJETO NO VSCODE.....16
main.tsx..... 16
index.css.....17
App.css..... 17
App.tsx.....18
assets..... 18
index.html..... 19
VISUALIZAÇÃO WEB.....19
GitHub-1.....20
07 - IMPLEMENTAR EXTENSÕES DO VSCODE.....21

GitHub-2.....21
08 - IMPLEMENTAR ESTILOS DEFAULT..... 22
GitHub-3.....24
09 - DESENVOLVER A API..... 25
COMPONENTE HEADER..... 25
Implementar o “components” e o “Header”..... 25
Chamar o Header no App.tsx..... 25
Estilizar o Header..... 26
index.....26
Criar e implementar o styles..... 26
Visualização web.....27
GitHub-4.....27
PÁGINA HOME.....28
Chamar o Home no App.tsx..... 28
Estilizar o Home..... 29
index.....29
Criar e implementar o styles..... 29
Visualização web.....30
GitHub-5.....30
COMPONENTE BUTTON_PRIMARY..... 31
Implementar o ButtonPrimary.....31
Estilizar o ButtonPrimary..... 31
Chamar o ButtonPrimary, no Home.....32
Visualização web.....32
GitHub-6.....32
PÁGINA GIT_SEARCH..... 33
Implementar o GitSearch.....33
Chamar o GitSearch no App.tsx.....33
Estilizar o GitSearch..... 34
Refatorar o código e chamar o ButtonPrimary no GitSearch..... 34
Visualização web.....35
GitHub-7.....35
COMPONENTE RESULT_GIT..... 36
Implementar o ResultGit.....36
Chamar o ResultGit, no GitSearch.....37
Estilo no GitResearch..... 37
Estilizar o ResultGit..... 38
Estylo no ResultGit.....38

Visualização web.....40
GitHub-8.....40
10 - ROTAS.....41
INSTALAR O REACT-ROUTER-DOM.....41
VERIFICAR NO PROJETO.....41
CRIAR O ROTES.....42
Implementar as rotas das páginas, no main.....42
Implementar o App.....43
Visualização web.....43
Implementar o link para os botões.....44
Botão “Começar”, do Home.....44
Nome “API GitHub” e “Entrar ”, do Header.....45
Visualização web.....45
GitHub-9.....46
11 - TRATAR EVENTOS DE FORMULÁRIO.....47
IMPLEMENTAR O INPUT, NO GIT_SEARCH.....47
Implementar o onChange.....47
Implementar a função handleChange.....47
Visualização web.....48
IMPLEMENTAR O SUBMIT, NO GIT_SEARCH.....48
Implementar o onSubmit.....48
Implementar a função handleSubmit.....49
CONTROLANDO O ESTADO DOS VALORES.....49
Implementar o formData.....49
Implementar o useState.....50
Implementar o input.....50
Visualização web.....51
ALTERANDO O ESTADO DA FUNÇÃO.....51
Implementar a função handleChange.....51
Implementar a função handleSubmit.....52
Visualização web.....52
GitHub-10.....52
12 - INTEGRANDO COM A API GITHUB.....53
INTEGRAR COM A API.....54
Implementar o tipo.....54
Implementar o useState.....54
Implementar o ResultGti, no GitSearch.....55
Implementar um condicional para renderizar o card.....55
AXIOS.....56
Instalar o Axios.....56

Implementar o axios, no gitSearch..... 56
Visualização web..... 57
TRATANDO ERRO (PERFIL INVÁLIDO)..... 57
Implementar o catch..... 57
Visualização web..... 58
GitHub-11..... 58
13 - REPOSITÓRIOS DO PERFIL, COMO COMPONENTE..... 59
COMPONENTE RESULT_GIT_REPO..... 59
Criar e implementar o componente ResultGitRepo..... 59
Implementar o estilo do ResultGitRepo..... 60
Implementar o estilo do card, no GitSearch..... 61
CHAMADA NA PAGINA GIT_SEARCH..... 61
Implementar o tipo..... 61
Implementar o useState para os repositórios..... 62
Implementar o handleSubmit para os repositórios..... 62
Implementar o map e a chamada do ResultGitRepo..... 63
Implementar o useState para o condicional do ResultGitRepo..... 63
Implementar o condicional findRepo..... 64
Visualização web..... 65
BOTÃO DO REPOSITORIO..... 66
Implementar o ButtonPrimary, no ResultSearch..... 66
Implementar estilo..... 66
Implementar a função handleSubmitRepos..... 67
Visualização web..... 68
GitHub-12..... 69
FIM..... 69

01 - OBJETIVO

Sistema que disponibiliza um formulário para buscar os dados de um determinado usuário do Github, utilizando a API pública do Github.

Sistema desenvolvido utilizando o ReactJS como frontend.

02 - ENDPOINT DA API GITHUB

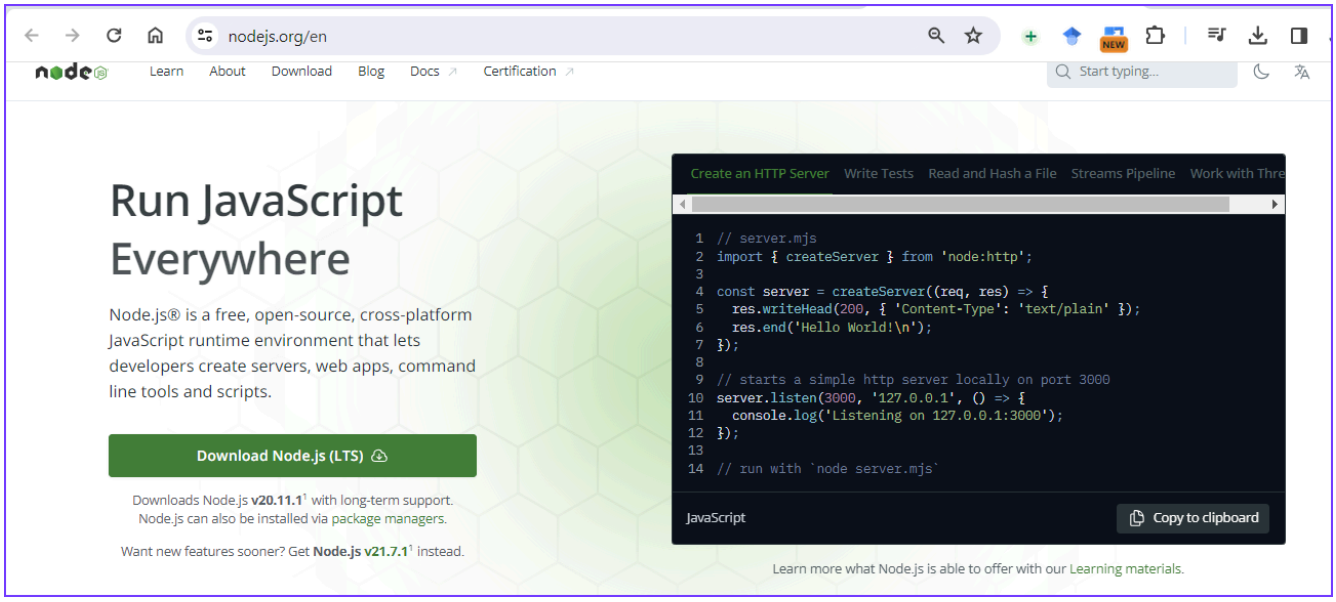
- https://api.github.com/users/auriceliof

```
// 20240326094438
// https://api.github.com/users/auriceliof

{
  "login": "auriceliof",
  "id": 4201131,
  "node_id": "MDQ6VXNlcjQyMDExMzE=",
  "avatar_url": "https://avatars.githubusercontent.com/u/4201131?v=4",
  "gravatar_id": "",
  "url": "https://api.github.com/users/auriceliof",
  "html_url": "https://github.com/auriceliof",
  "followers_url": "https://api.github.com/users/auriceliof/followers",
  "following_url": "https://api.github.com/users/auriceliof/following{/other_user
  "gists_url": "https://api.github.com/users/auriceliof/gists{/gist_id}",
```

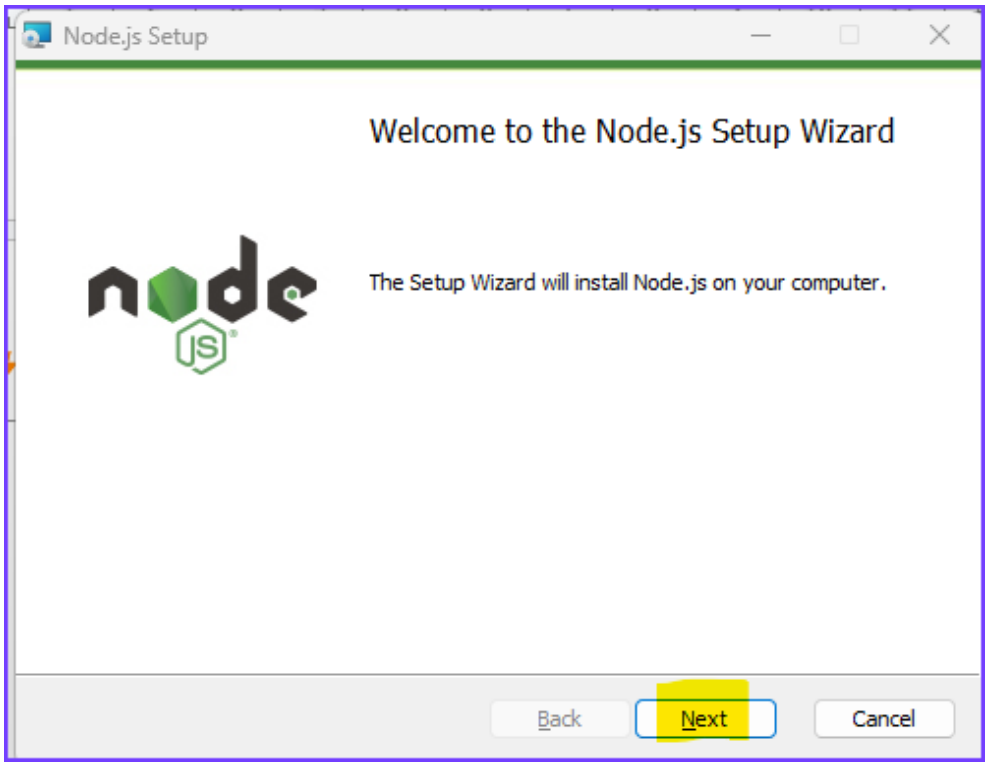
03 - INSTALAÇÃO DO NODEJS

BAIXAR

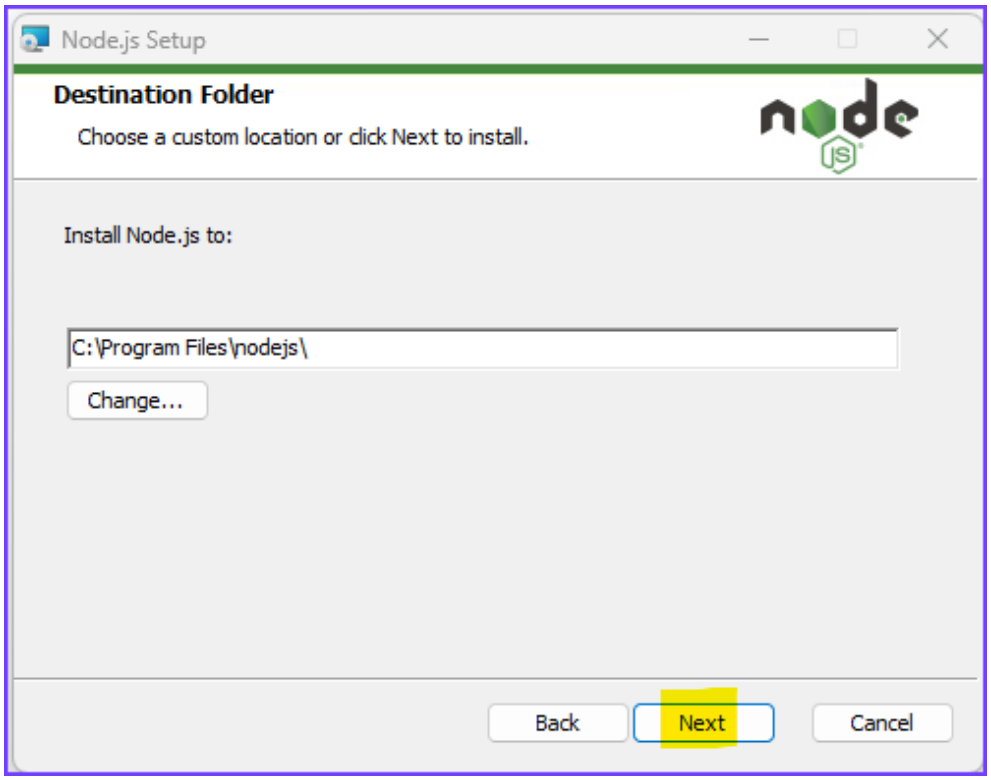
- https://nodejs.org

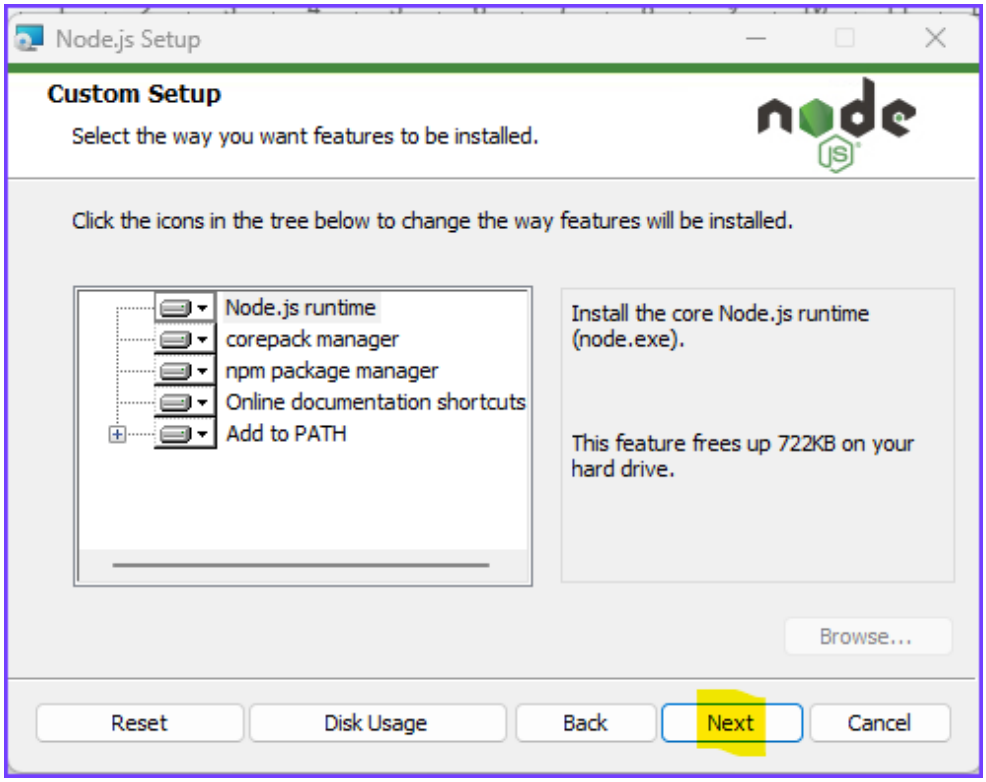


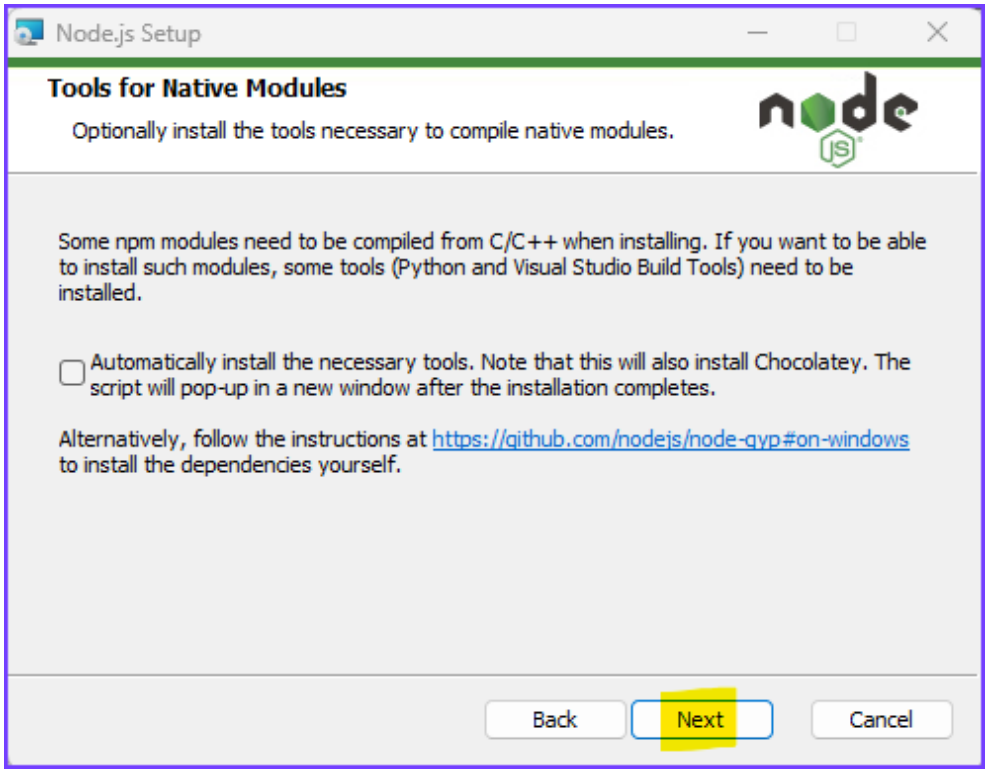
EXECUTAR

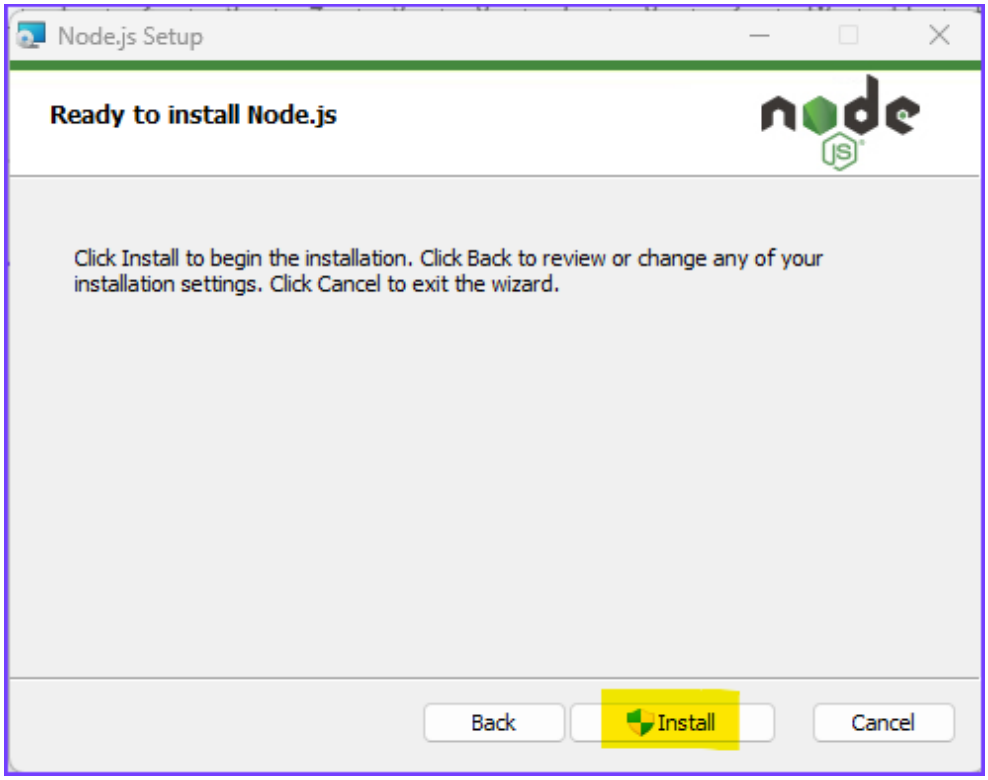


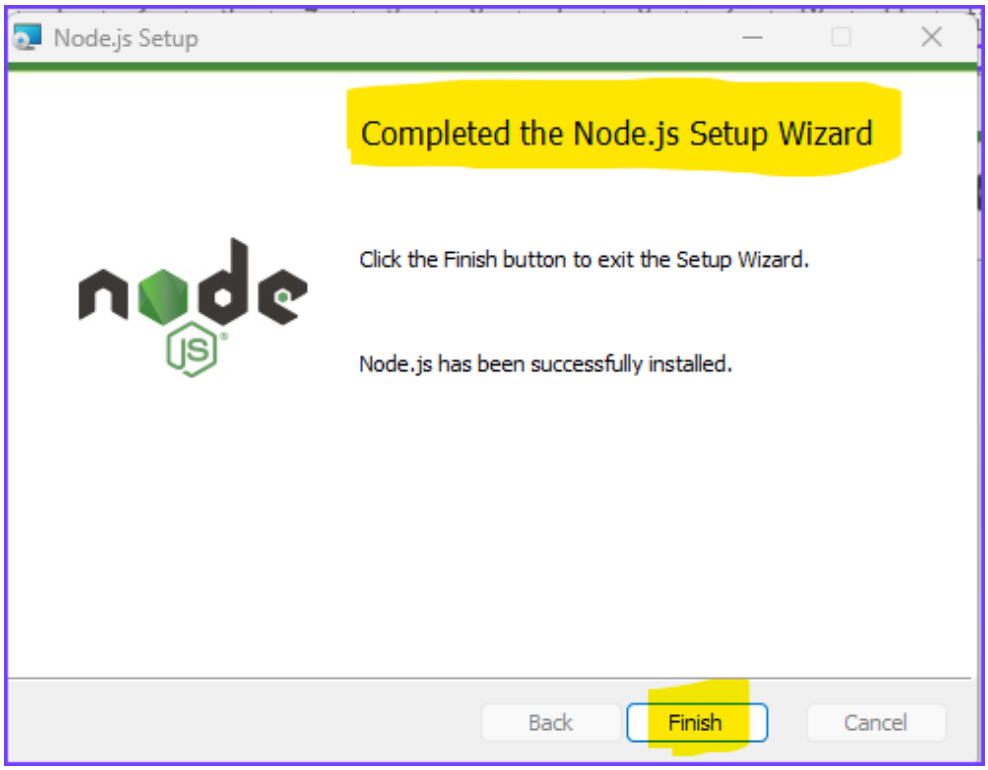

Node.js Setup
End-User License Agreement
Please read the following license agreement carefully
Node.js is licensed for use as follows:
Copyright Node.js contributors. All rights reserved.
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject
I accept the terms in the License Agreement
Print
Back
Next
Cancel


Node.js Setup
Destination Folder
Choose a custom location or click Next to install.
Install Node.js to:
C:\Program Files\nodejs\
Change...
Back
Next
Cancel


Node.js Setup
Custom Setup
Select the way you want features to be installed.
Click the icons in the tree below to change the way features will be installed.
Node.js runtime
corepack manager
npm package manager
Online documentation shortcuts
Add to PATH
Install the core Node.js runtime (node.exe).
This feature frees up 722KB on your hard drive.
Browse...
Reset
Disk Usage
Back
Next
Cancel

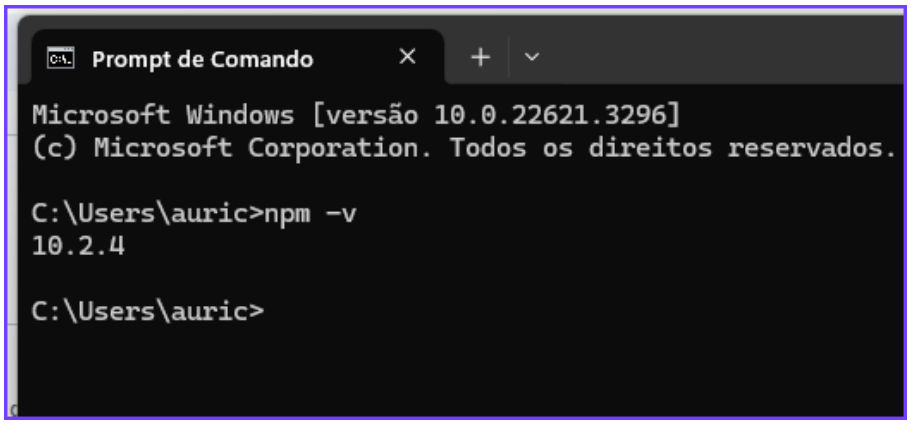

Node.js Setup
Tools for Native Modules
Optionally install the tools necessary to compile native modules.
Some npm modules need to be compiled from C/C++ when installing. If you want to be able to install such modules, some tools (Python and Visual Studio Build Tools) need to be installed.
Automatically install the necessary tools. Note that this will also install Chocolatey. The script will pop-up in a new window after the installation completes.
Alternatively, follow the instructions at https://github.com/nodejs/node-gyp#on-windows to install the dependencies yourself.
Back
Next
Cancel


Node.js Setup
Ready to install Node.js
Click Install to begin the installation. Click Back to review or change any of your installation settings. Click Cancel to exit the wizard.
Back
Install
Cancel


Node.js Setup
Completed the Node.js Setup Wizard
Click the Finish button to exit the Setup Wizard.
Node.js has been successfully installed.
Back
Finish
Cancel

TESTAR NO TERMINAL

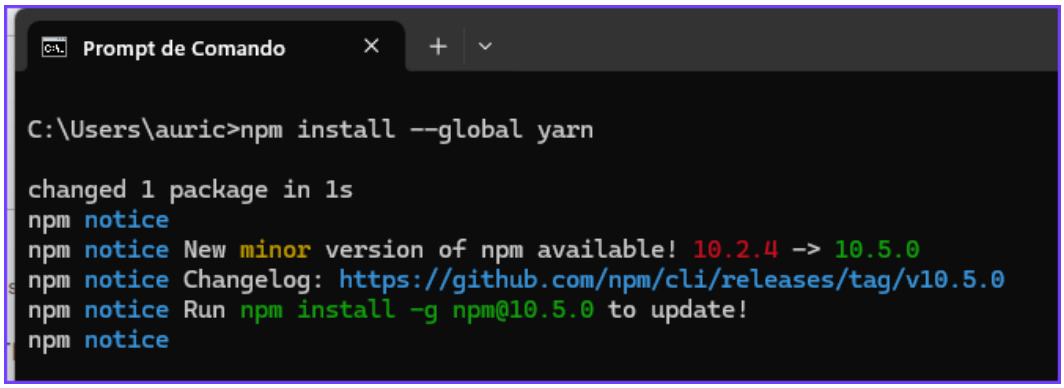
- npm -v



04 - YARN

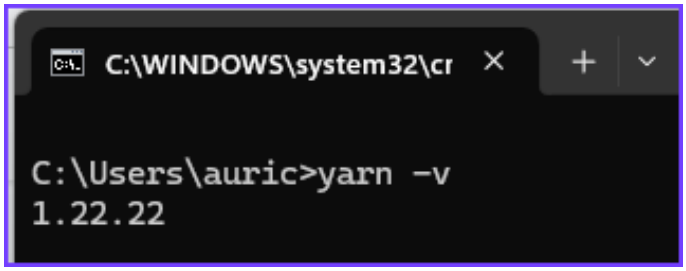
INSTALAR

- npm install --global yarn



TESTAR

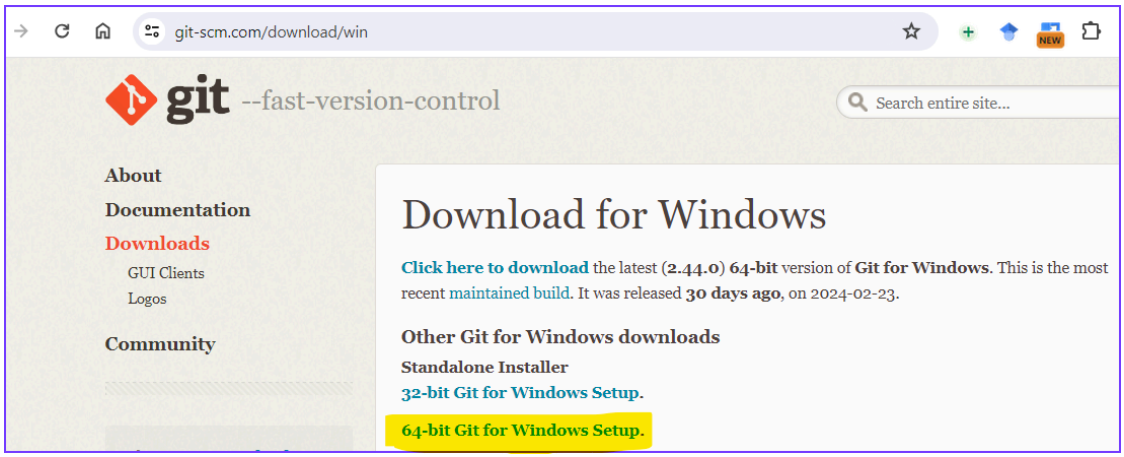
- yarn -v



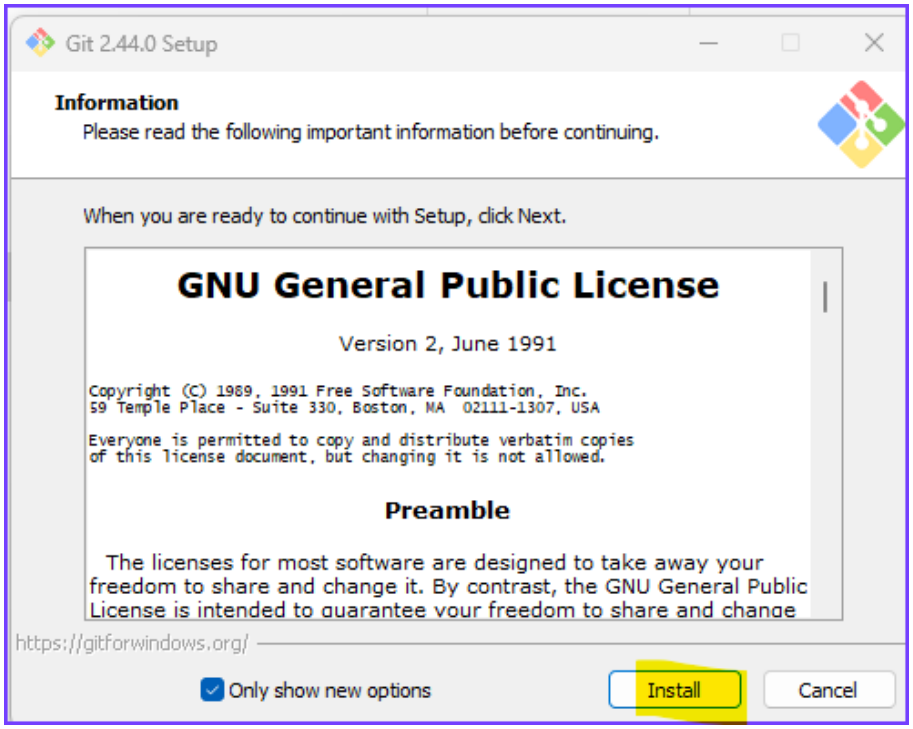
05 - GIT

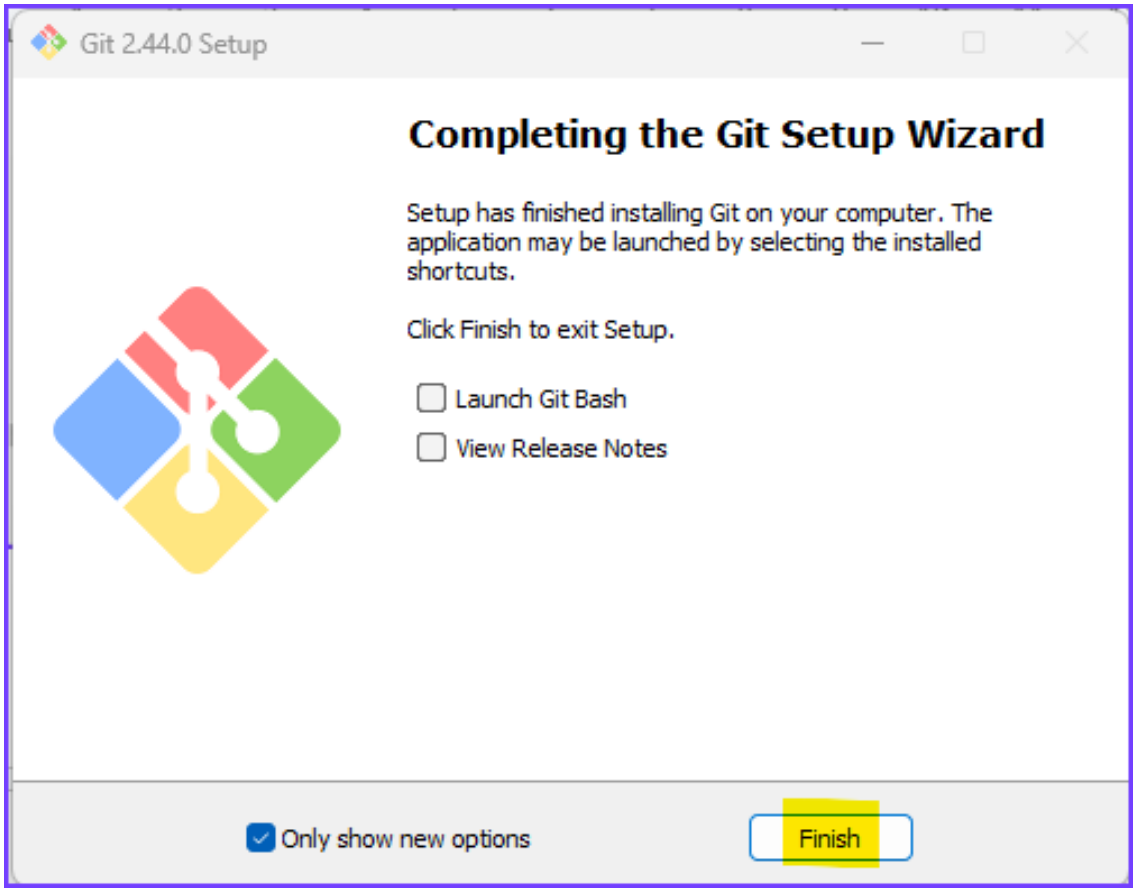
BAIXAR

- https://git-scm.com/download/win



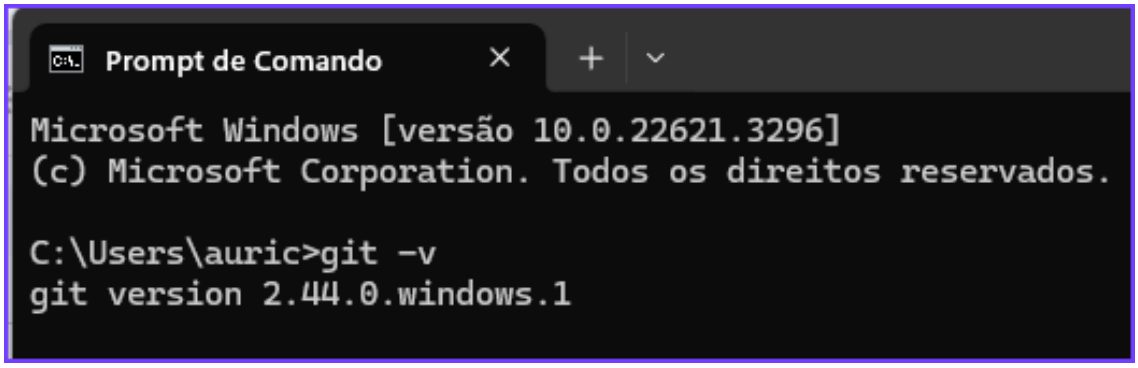
INSTALAR




Git 2.44.0 Setup
Completing the Git Setup Wizard
Setup has finished installing Git on your computer. The application may be launched by selecting the installed shortcuts.
Click Finish to exit Setup.
Launch Git Bash
View Release Notes
Only show new options
Finish

TESTAR

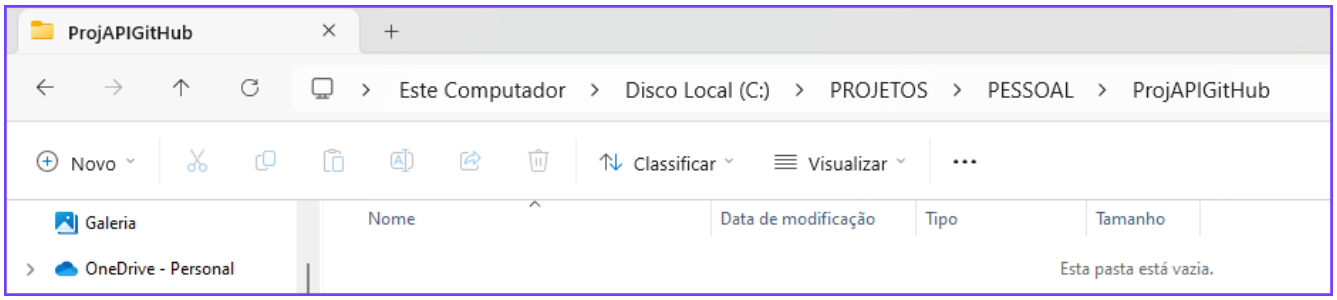
- git -v


Prompt de Comando
Microsoft Windows [versão 10.0.22621.3296]
(c) Microsoft Corporation. Todos os direitos reservados.
C:\Users\auric>git -v
git version 2.44.0.windows.1

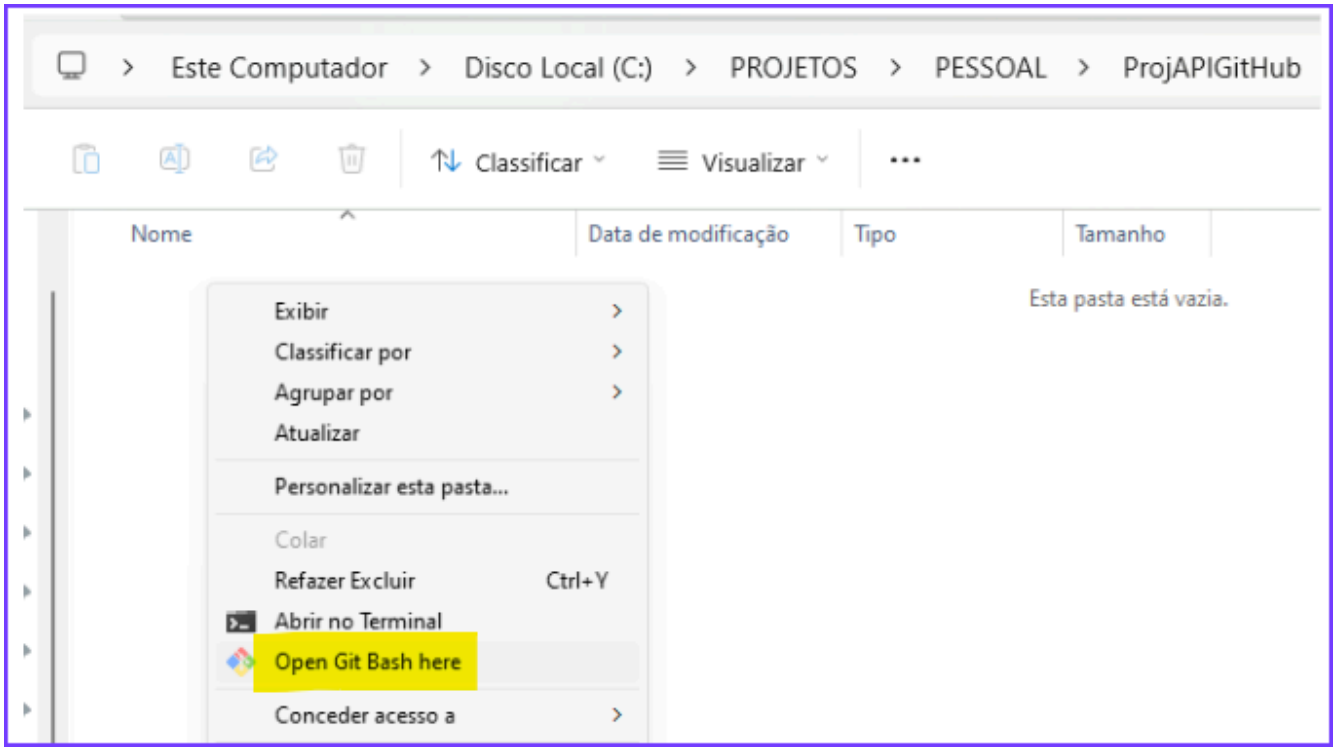
06 - CRIAR O PROJETO API_GITHUB

DEFINIR O DIRETÓRIO PARA O PROJETO

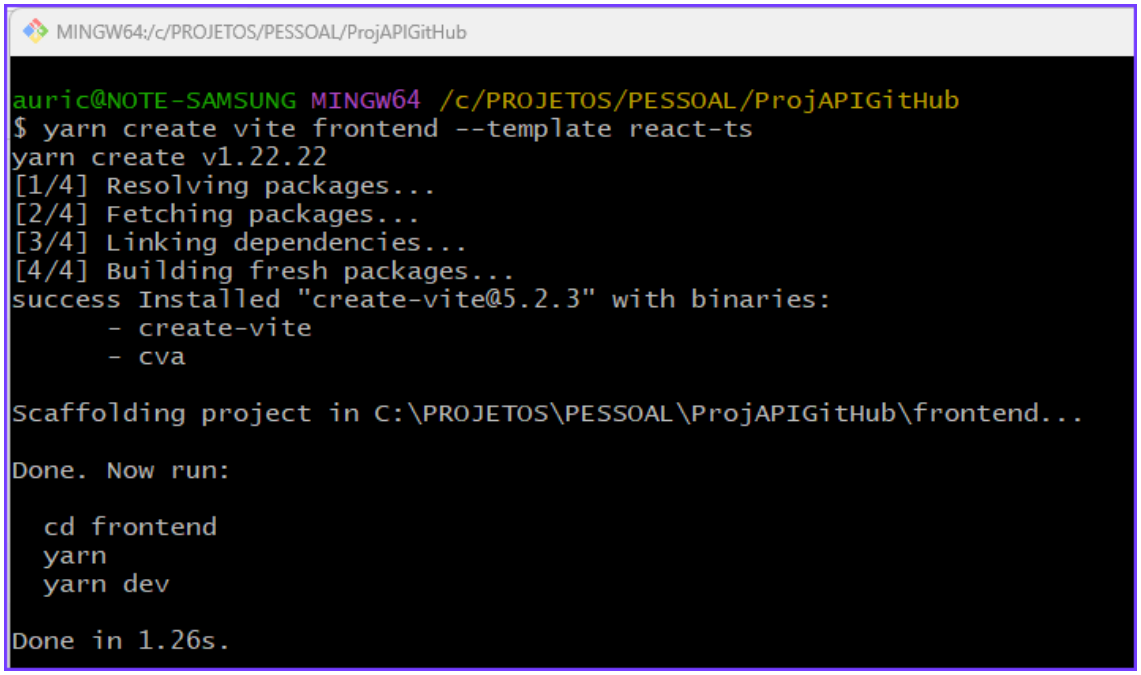
- C:\PROJETOS\PESSOAL\ProjAPIGitHub



CRIAR O PROJETO

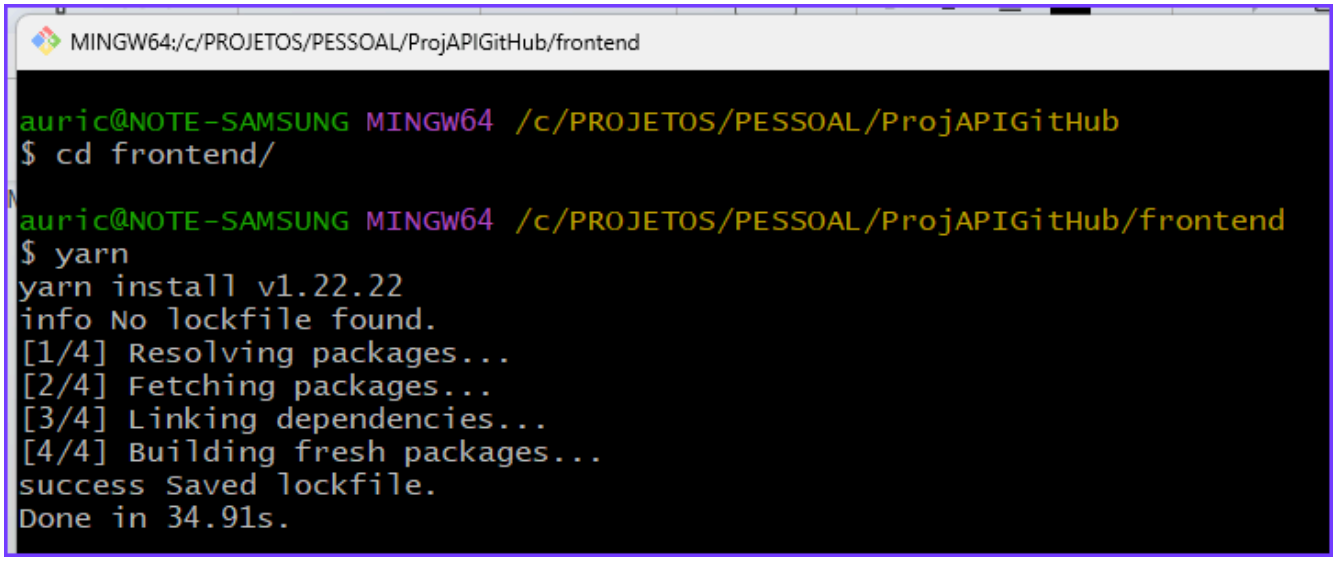


- yarn create vite frontend --template react-ts



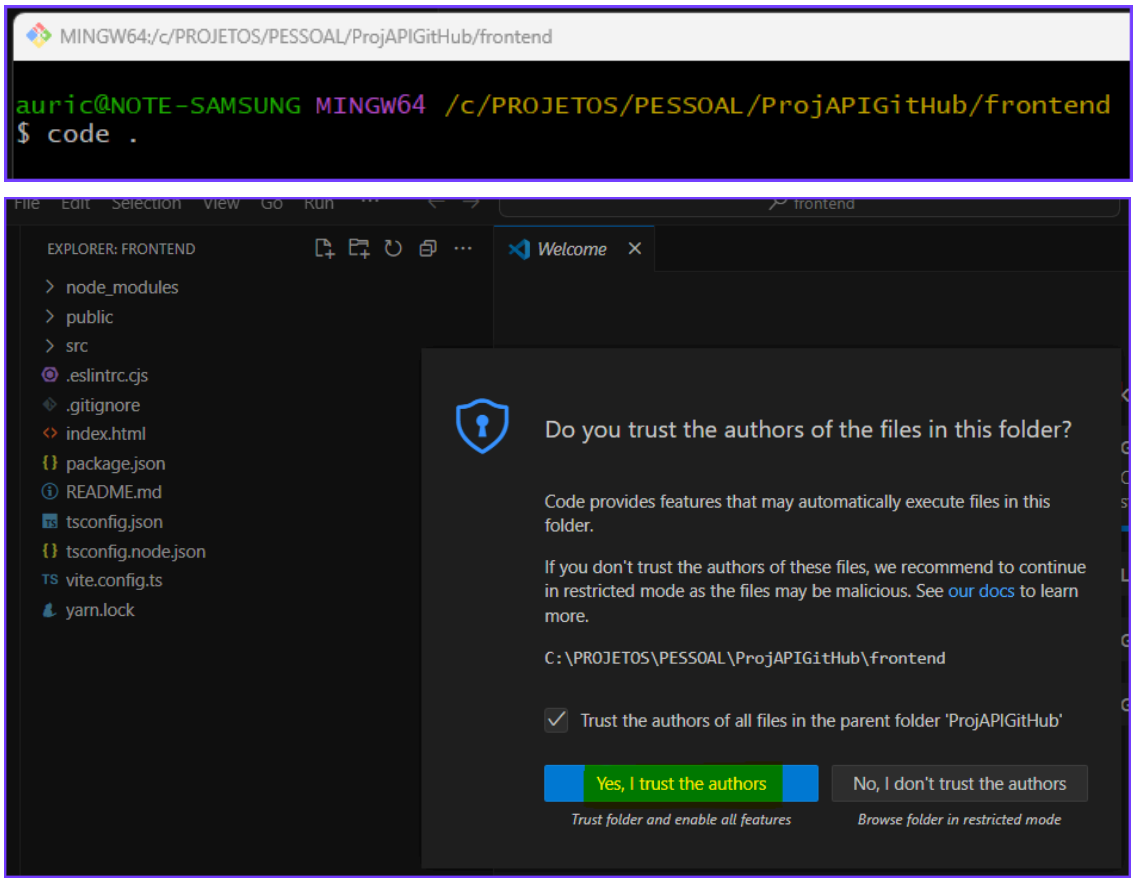
ATUALIZAR

- cd frontend/
- yarn



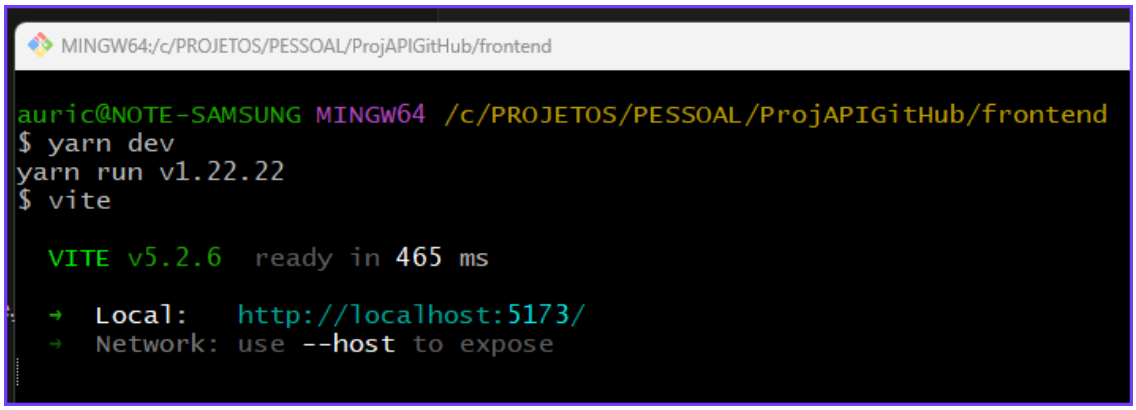
ABRIR COM O CODE

- code .

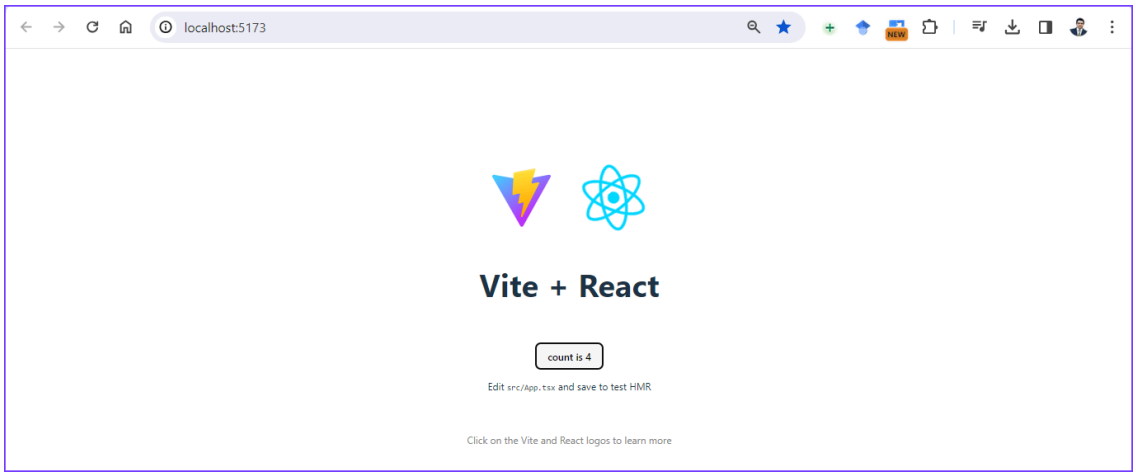


INICIAR O PROJETO

- yarn dev

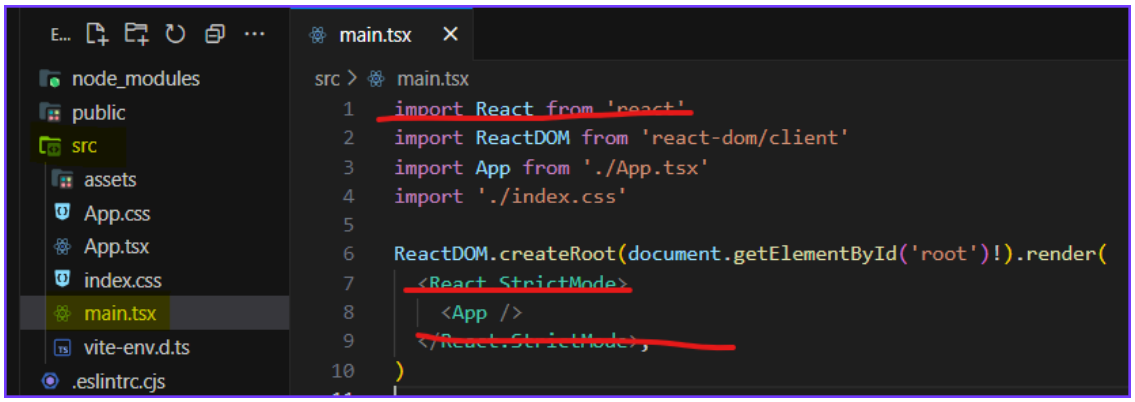


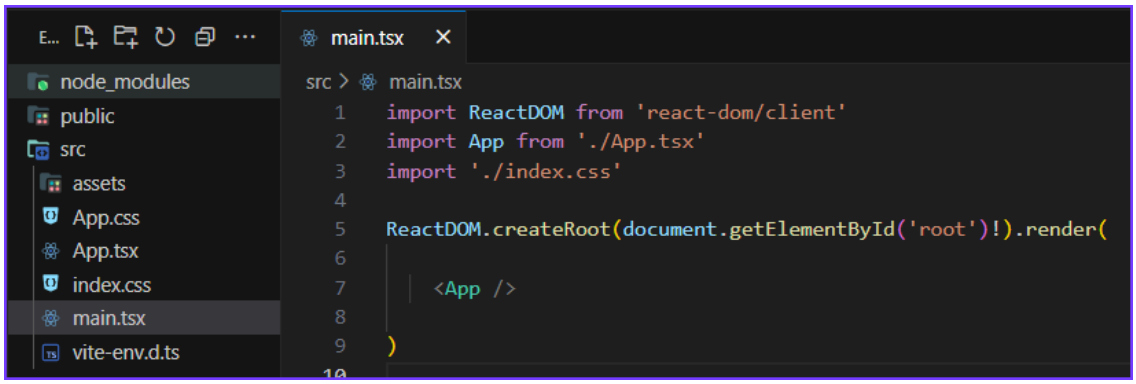
VISUALIZAÇÃO WEB



ENXUGAR O PROJETO NO VSCODE

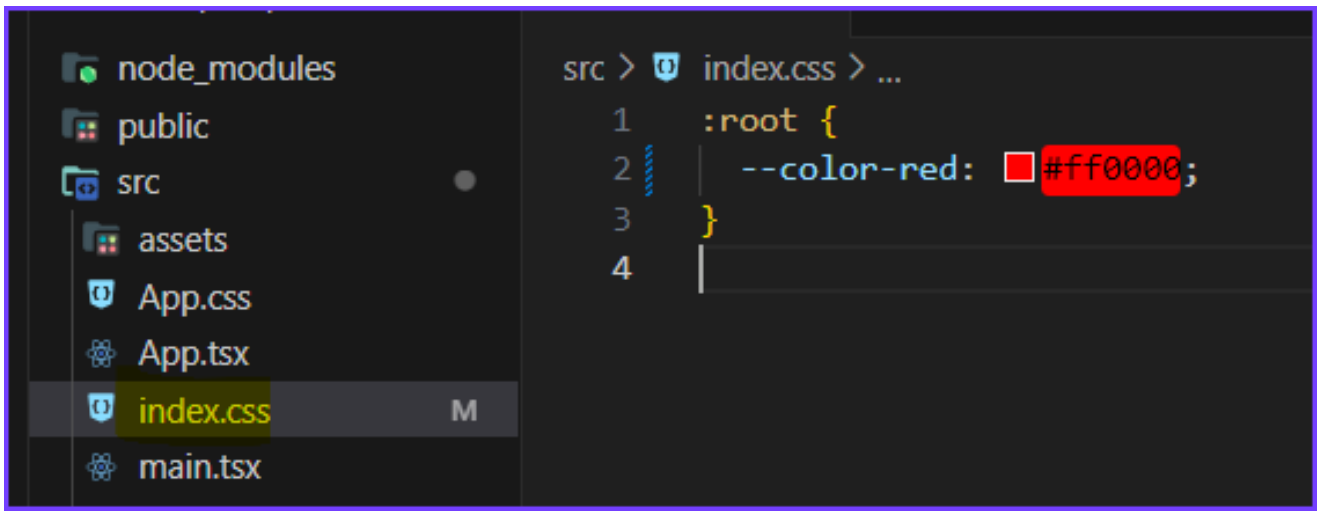
main.tsx





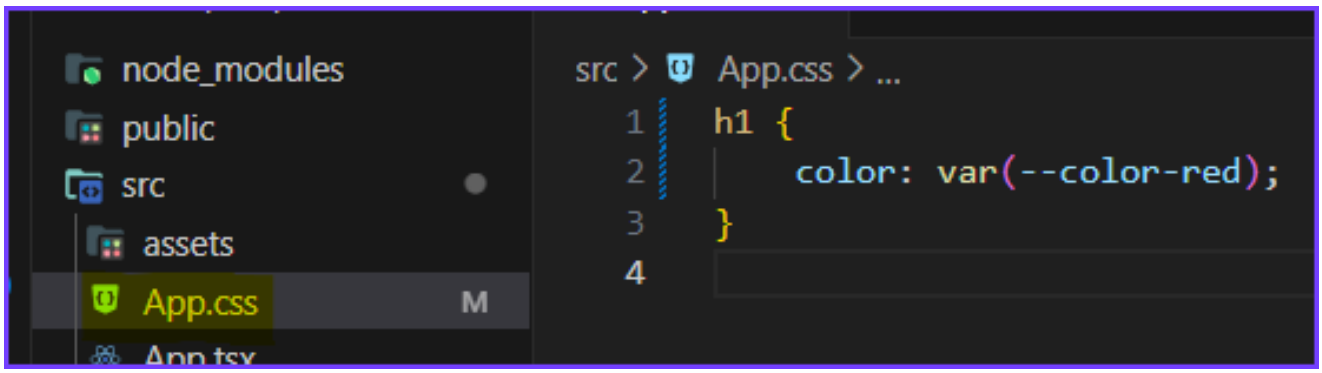
index.css

- Apaga todo o conteúdo, deixando apenas o mínimo.



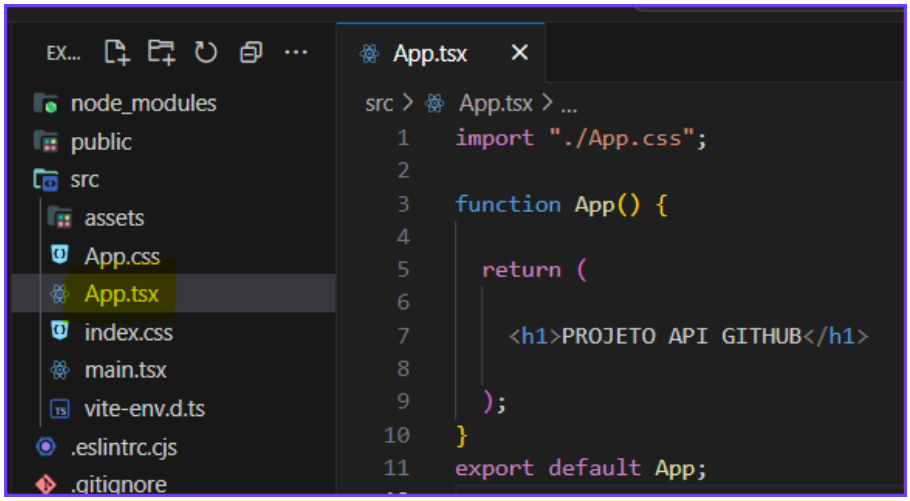
App.css

- Apaga todo o css, puxando a variável do index.css.



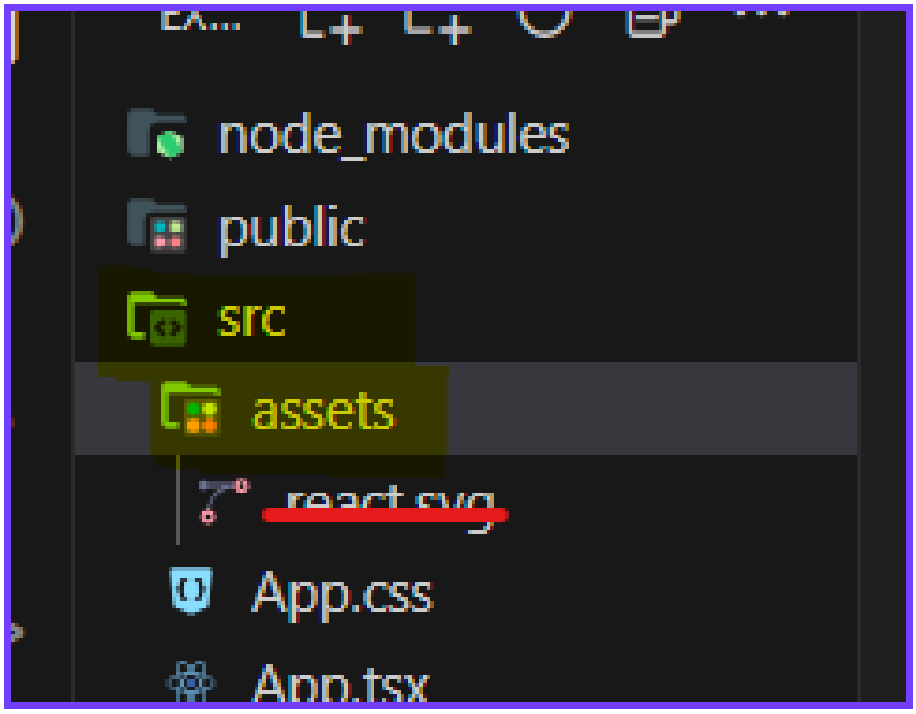
App.tsx

- Enxugar



assets

- Apagar

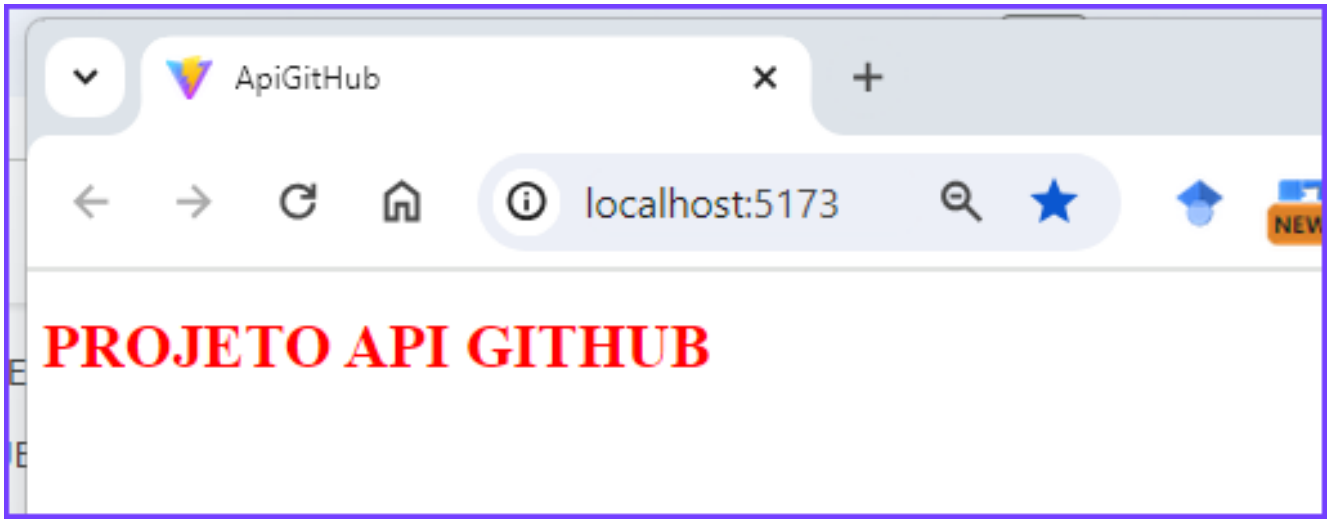


index.html

- Renomear

```
index.html > ...
1   <!doctype html>
2   <html lang="en">
3     <head>
4       <meta charset="UTF-8" />
5       <link rel="icon" type="image/svg+xml" href="/vite.svg" />
6       <meta name="viewport" content="width=device-width, initial-scale=1.0" />
7       <title>ApiGitHub</title>
8     </head>
9     <body>
10      <div id="root"></div>
11      <script type="module" src="/src/main.tsx"></script>
12    </body>
13  </html>
```

VISUALIZAÇÃO WEB



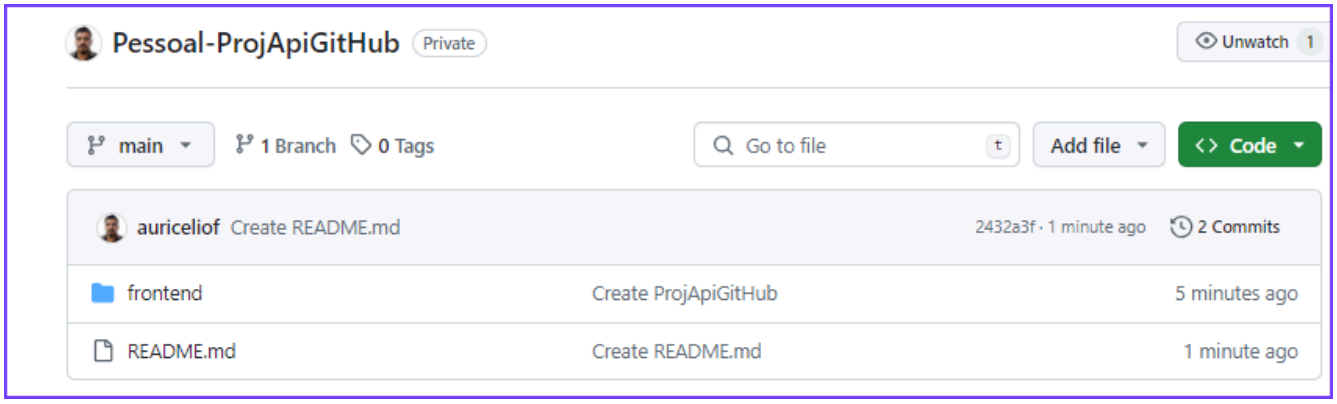
GitHub-1

Criar o projeto no Github

- auriceliof/Pessoal/ProjApiGitHub

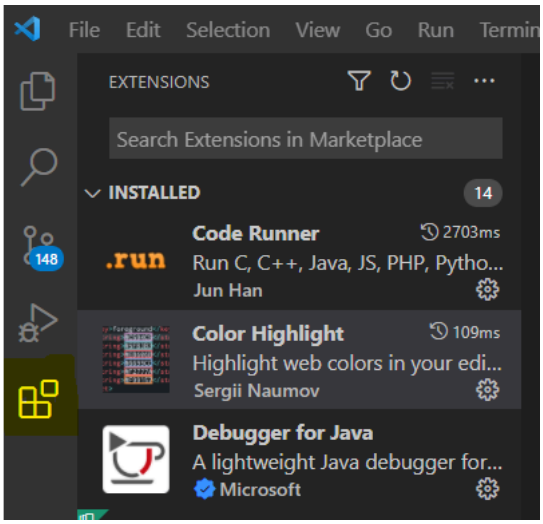
Associar o projeto local ao Github

- git init
- git add .
- git commit -m "Create ProjApiGitHub"
- git branch -M main
- git remote add origin https://github.com/auriceliof/Pessoal-ProjApiGitHub.git
- git push -u origin main



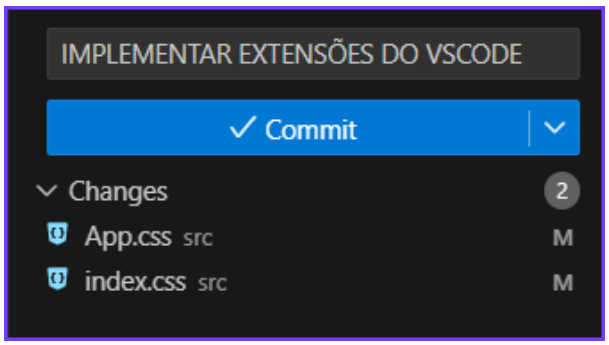
07 - IMPLEMENTAR EXTENSÕES DO VSCODE

- **Color Highlight**
- **JSX HTML <tags/>**
- **IntelliCode**
- **Code Runner**
- **Live Server**
- **Material Theme Icons**
- **Prettier – Code formatter**



GitHub-2

- IMPLEMENTAR EXTENSÕES DO VSCODE



08 - IMPLEMENTAR ESTILOS DEFAULT

```
@import url("https://fonts.googleapis.com/css2?family=Open+Sans:wght@400;700&display=swap");

:root {
  --pag-color-bg-primary: #e8e8e8;
  --pag-color-bg-secondary: #ffe500;
  --pag-color-bg-tertiary: #cac7c7;
  --pag-color-bg-quaternary:#c9d9f2;
  --pag-color-bg-quinternary:  #b8c6b9;

  --pag-color-card-bg: #fff;
  --pag-color-card-border: #d9d9d9;

  --pag-color-btn-primary: #0caf1d;
  --pag-color-btn-secondary: #f33;

  --pag-color-font-primary: #636363;
  --pag-color-font-secondary: #3483fa;
  --pag-color-font-tertiary: #fff;
  --pag-color-font-placeholder: #d9d9d9;

  --pag-color-error: #f33;
}

* {
  box-sizing: border-box;
  margin: 0;
  padding: 0;
  font-family: "Open Sans";
}

a,
a:hover {
  text-decoration: none;
  color: unset;
}

html,
body {
  background-color: var(--pag-color-bg-primary);
}

main {
  padding: 0 20px;
}

/*--------------------------------------------------------------*/
```

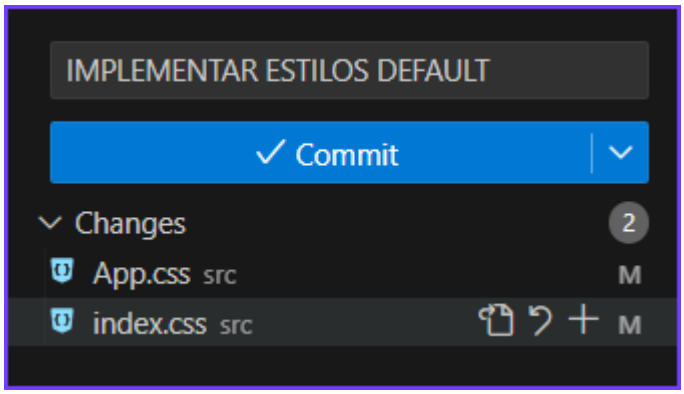
```
/* generic styles */

.pag-container {
  width: 100%;
  max-width: 960px;
  margin: 0 auto;
}

.pag-mb20 {
  margin-bottom: 20px;
}

.pag-mb40 {
  margin-bottom: 40px;
}

.pag-mt20 {
  margin-top: 20px;
}

.pag-mt40 {
  margin-top: 40px;
}

.pag-section-title {
  text-align: center;
  color: var(--pag-color-font-primary);
  font-size: 16px;
}

.pag-txt-left {
  text-align: left;
}

@media (min-width: 576px) {
  .pag-section-title {
    text-align: left;
    font-size: 24px;
  }
}
```

- src/index.css

```
src > index.css > ...
1   @import url("https://fonts.googleapis.com/css2?family=Open+Sans:wght@400;700&display=swap");
2
3   :root {
4     --pag-color-bg-primary: #e8e8e8;
5     --pag-color-bg-secondary: #ffe500;
6     --pag-color-bg-tertiary: #636363;
7     --pag-color-bg-quaternary: #c9d9f2;
8
9     --pag-color-card-bg: #fff;
10    --pag-color-card-border: #d9d9d9;
11
12    --pag-color-btn-primary: #0caf1d;
13    --pag-color-btn-secondary: #f33;
14
15    --pag-color-font-primary: #636363;
16    --pag-color-font-secondary: #3483fa;
17    --pag-color-font-tertiary: #fff;
18    --pag-color-font-placeholder: #d9d9d9;
19
20    --pag-color-error: #f33;
21  }
22
23  * {
24    box-sizing: border-box;
```

GitHub-3

- IMPLEMENTAR ESTILOS DEFAULT



09 - DESENVOLVER A API

COMPONENTE HEADER

Implementar o “components” e o “Header”

src > components > Header > index.tsx > ...

```tsx
export default function Header() {

    return (
        <header>
            <nav>
                <h1>API GitHub</h1>
                <div>
                    <a href="#">Entrar</a>
                </div>
            </nav>
        </header>
    );
}

```

***NOTA**: Neste momento, devemos criar o COMPONENTS, onde iremos alocar todos os componentes, a partir daqui.*

Chamar o Header no App.tsx

src > App.tsx > ...

```tsx
import "./App.css";
import Header from "./components/Header";

function App() {

  return (
    <>
      <Header />
      <h1>PROJETO API GITHUB</h1>
    </>

  );
}
export default App;
```

Estilizar o Header

index

src > components > Header > index.tsx > ...

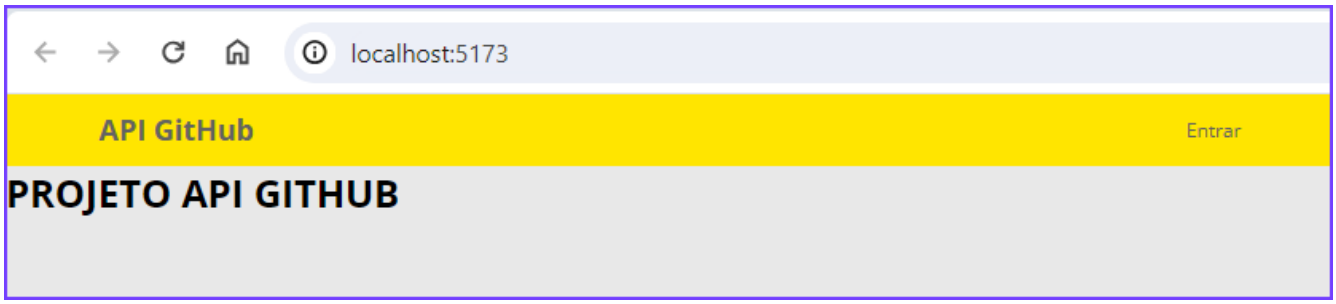
```tsx
import './styles.css';

export default function Header() {

    return (
        <header className="pag-header">
            <nav className="pag-container">
                <div className="pag-header-navbar">
                    <h1>API GitHub</h1>
                    <div>
                        <a href="#">Entrar</a>
                    </div>
                </div>
            </nav>
        </header>
    );
}
```

Criar e implementar o styles

src > components > Header > styles.css > ...

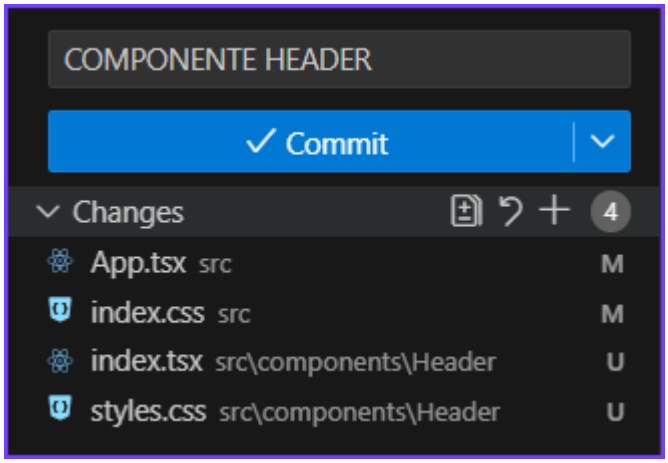
```css
.pag-header {
    background-color: var(--pag-color-bg-secondary);
    color: var(--pag-color-font-primary);
    height: 60px;
    display: flex;
    align-items: center;
}

.pag-header-navbar {
    display: flex;
    justify-content:space-between;
    align-items: center;
}
```

Visualização web



GitHub-4

- COMPONENTE HEADER



PÁGINA HOME

src > pages > Home > index.tsx > ...

```tsx
export default function Home() {

    return (
        <div>
            <div>
                <h1>Projeto API GitHub</h1>
            </div>
            <div>
                <h1>Buscar perfil de usuários do GitHub</h1>
            </div>
            <div>
                <button>Começar</button>
            </div>
        </div>
    );
}
```

***NOTA**: Neste momento, devemos criar o PAGES, onde iremos alocar todas as páginas, que irão receber rotas, a partir daqui.*

Chamar o Home no App.tsx

src > App.tsx > ...

```tsx
import "./App.css";
import Header from "./components/Header";
import Home from "./pages/Home";

function App() {

  return (
    <>
      <Header />
      <Home />
    </>

  );
}
export default App;

```

Estilizar o Home

index

src > pages > Home > index.tsx > ...

```tsx
import './styles.css';

export default function Home() {

    return (
        <div className="pag-container ">
            <div className="pag-mb20 pag-mt40">
                <h1>Projeto API GitHub</h1>
            </div>
            <div className="pag-mb40 pag-home-content">
                <h4>Buscar perfil de usuários do GitHub</h4>
            </div>
            <div>
                <button className='pag-home-btn'>Começar</button>
            </div>
        </div>
    );
}
```

Criar e implementar o styles

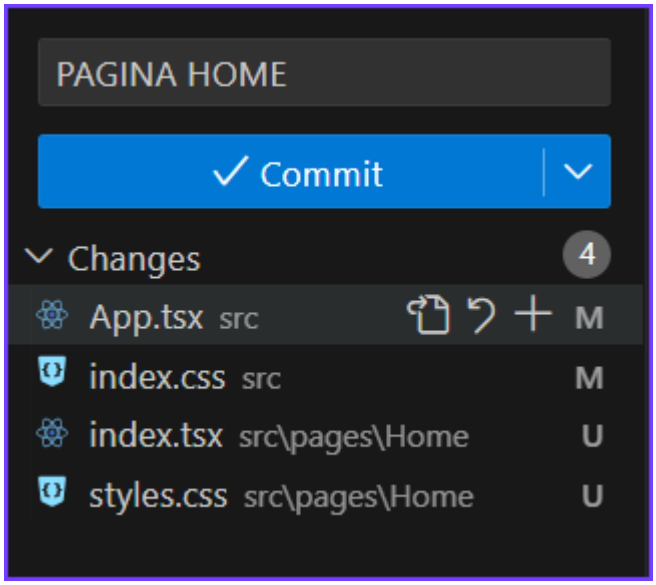
src > pages > Home > styles.css > ...

```css
.pag-home-content{
    color: var(--pag-color-font-primary);
}
```

Visualização web

GitHub-5

- PAGINA HOME



COMPONENTE BUTTON_PRIMARY

Implementar o ButtonPrimary

src > components > ButtonPrimary > index.tsx > ...

```tsx
import './styles.css';

type Props = {
    name: string;
}

export default function ButtonPrimary({name}: Props) {

    return(
        <button className="pag-btn">
            {name}
        </button>
    );
}
```

Estilizar o ButtonPrimary

src > components > ButtonPrimary > styles.css > ...

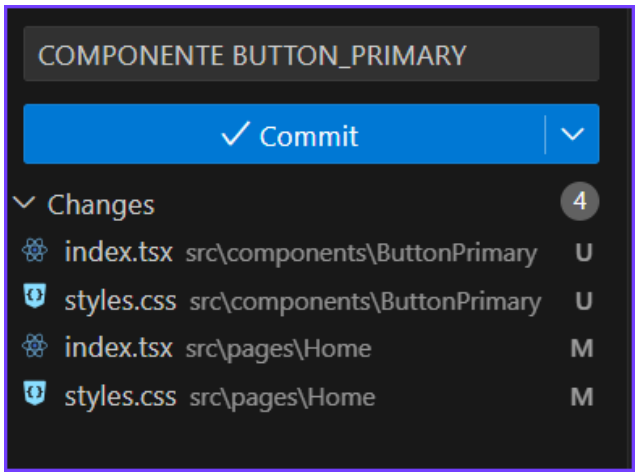
```css
.pag-btn {
    display: flex;
    justify-content: center;
    font-size: 1rem;
    font-weight: bold;
    background-color: var(--pag-color-btn-primary);
    color: var(--pag-color-font-tertiary);
    width: 150px;
    padding: 5px;
    border: none;
    border-radius: 5px;
    cursor: pointer;
}
```

Chamar o ButtonPrimary, no Home

Visualização web

GitHub-6

- COMPONENTE BUTTON_PRIMARY



PÁGINA GIT_SEARCH

Implementar o GitSearch

src > pages > GitSearch > index.tsx > ...

```tsx
import ButtonPrimary from '../../components/ButtonPrimary';
import './styles.css';

export default function GitSearch() {

    return (
        <div className="pag-gitsearch-container">
            <div className="pag-gitsearch-card">
                <div className="pag-mb20">
                    <h2>Encontre um perfil GitHub</h2>
                </div>
                <form className="pag-mb40">
                    <input
                        type="text"
                        className="pag-gitsearch-input"
                        placeholder="Usuário GitHub"
                    />
                </form>
                <div className="pag-gitsearch-btn">
                    <ButtonPrimary name="Encontrar"/>
                </div>
            </div>
        </div>
    );
}
```

Chamar o GitSearch no App.tsx

src > App.tsx > ...

```tsx
import "./App.css";
import Header from "./components/Header";
import GitSearch from "./pages/GitSearch";

function App() {

  return (
    <>
      <Header />
      <GitSearch />
    </>

  );
}
export default App;
```

Estilizar o GitSearch

src > pages > GitSearch > styles.css > ...

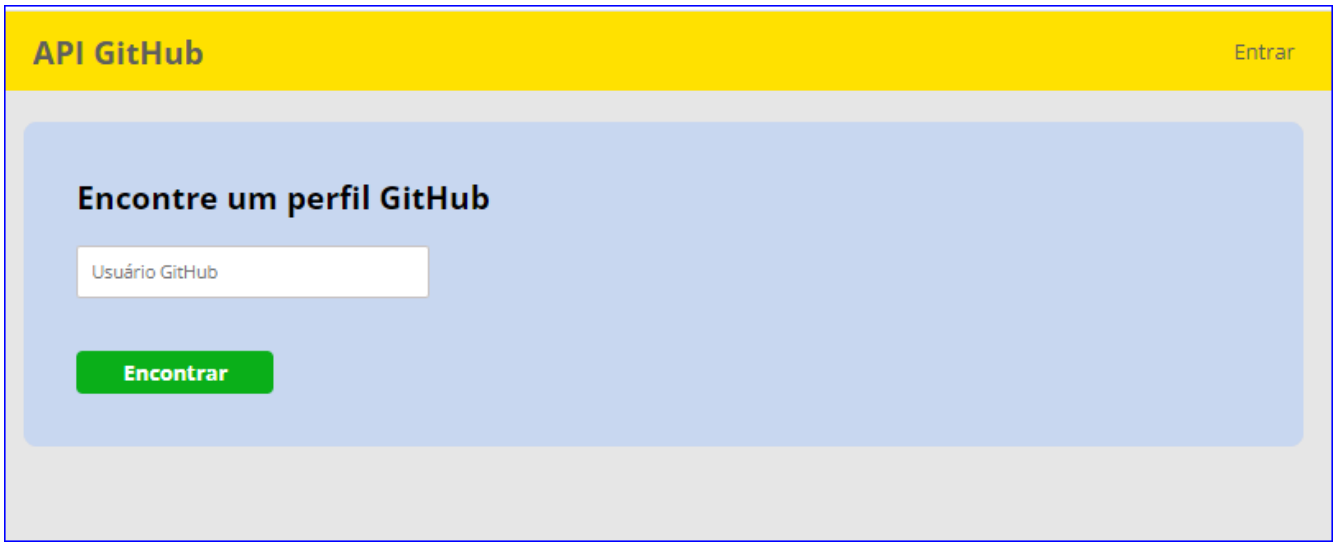
```css
.pag-gitsearch-container {
    display: flex;
    justify-content: center;
    align-items: center;
    flex-direction: column;
    position: relative;
    margin: 2% 10%;
}

.pag-gitsearch-card {
    width: 100%;
    background: var(--pag-color-bg-quaternary);
    border-radius: 10px;
    padding: 40px;
}

.pag-gitsearch-input {
    width: 30%;
    border: 1px solid var(--pag-color-bg-tertiary);
    border-radius: 3px;
    padding: 10px;
    color: var(--pag-color-font-secondary);
    font-weight: bold;
}

.pag-gitsearch-input::placeholder {
    font-weight: lighter;
}
```

Refatorar o código e chamar o ButtonPrimary no GitSearch

src > pages > GitSearch > index.tsx > ...

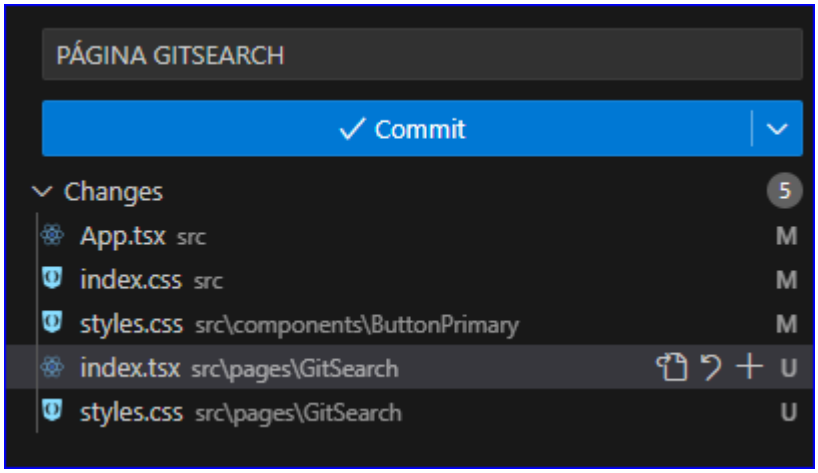
```tsx
export default function GitSearch() {

    return (
        <div className="pag-gitsearch-container">
            <div className="pag-gitsearch-card">
                <div className="pag-mb20">
                    <h2>Encontre um perfil GitHub</h2>
                </div>
                <form >
                    <div className="pag-mb40">
                        <input
                            type="text"
                            className="pag-gitsearch-input"
                            placeholder="Usuário GitHub"
                        />
                    </div>
                    <div className="pag-gitsearch-btn">
                        <ButtonPrimary name="Encontrar"/>
                    </div>
                </form>
            </div>
```

Visualização web



GitHub-7

- PÁGINA GITSEARCH



COMPONENTE RESULT_GIT

Implementar o ResultGit

src > components > ResultGit > index.tsx > ...

```tsx
import './styles.css';

type Props = {
    foto: any;
    nome?: string;
    perfil?: string;
    localidade?: string;
    seguidores?: number;
    repoPublicos?: number;
}
```

```tsx
export default function ResultGit({foto, nome, perfil, localidade, seguidores, repoPublicos}: Props) {

    return (
        <div className="pag-resultgit-container">
            <div className="pag-resultgit-card">
                <div className="pag-resultgit-foto">
                    <img src={foto} alt='foto' />
                </div>
                <div className="pag-resultgit-information">

                    <h4>Informações</h4>

                    <div className="pag-resultgit-content">
                        <h5>Nome: </h5>
                        <h6>{nome}</h6>
                    </div>
                    <div className="pag-resultgit-content">
                        <h5>Perfil: </h5>
                        <h6>{perfil}</h6>
                    </div>
                    <div className="pag-resultgit-content">
                        <h5>Localidade: </h5>
                        <h6>{localidade}</h6>
                    </div>
                    <div className="pag-resultgit-content">
                        <h5>Seguidores: </h5>
                        <h6>{seguidores}</h6>
                    </div>
                    <div className="pag-resultgit-content">
                        <h5>Repositórios públicos: </h5>
                        <h6>{repoPublicos}</h6>
                    </div>

                </div>
            </div>
        </div>
    );
}
```

Chamar o ResultGit, no GitSearch

pages > GitSearch > index.tsx > ...

```tsx
                    <div className="pag-mt20 pag-gitsearch-resultgit-card">
                        <ResultGit
                            foto="foto"
                            nome="nome"
                            perfil="perfil"
                            localidade="localidade"
                            seguidores={1}
                            repoPublicos={2}
                        />
                    </div>
                )}
            </div>
        );
    }
```

Estilo no GitResearch

src > pages > GitSearch > styles.css > ...

```css
29
30  .pag-gitsearch-resultgit-card {
31      width: 100%;
32      background: var(--pag-color-bg-tertiary);
33      border-radius: 10px;
34      padding: 40px;
35  }
```

Estilizar o ResultGit

Estylo no ResultGit

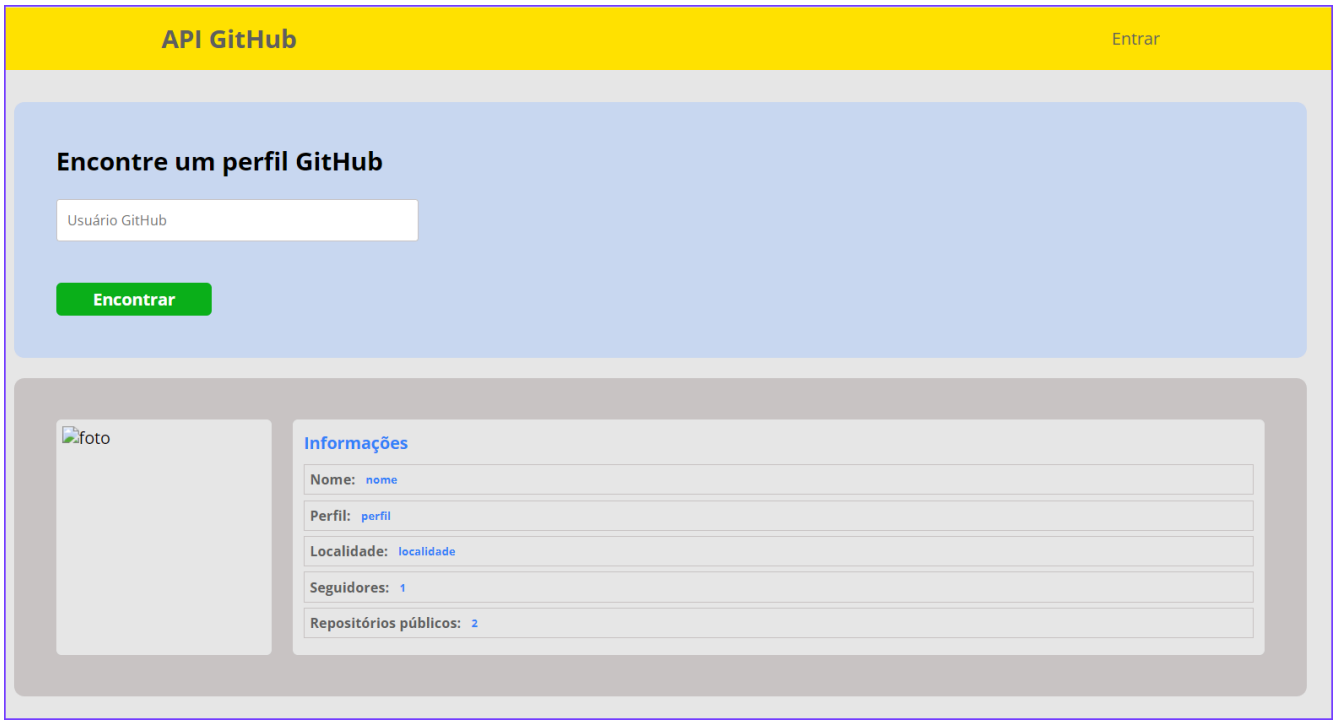
src > components > ResultGit > styles.css > ...

```css
.pag-resultgit-card {
    display: flex;

}

.pag-resultgit-foto {
    display: flex;
    justify-content: center;
    align-items: center;
    width: 20%;
    border: 1px solid var(--pag-color-bg-primary);
    border-radius: 5px;
    margin-right: 20px;
    box-sizing: border-box;
    background: var(--pag-color-bg-primary);
}

.pag-resultgit-foto img {
    width: 95%;
    height: 95%;
}

.pag-resultgit-information {
    width: 90%;
    padding: 10px;
    background: var(--pag-color-bg-primary);
    border: 1px solid var(--pag-color-bg-primary);
    border-radius: 5px;

}
```

src > components > ResultGit > styles.css > ...

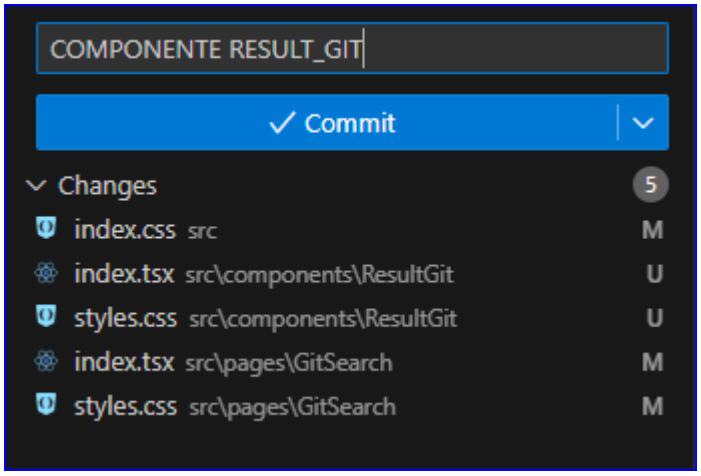
```
32  .pag-resultgit-information h4 {
33      color: var(--pag-color-font-secondary);
34      margin-bottom: 10px;
35  }
36
37  .pag-resultgit-content {
38      display: flex;
39      align-items: center;
40      border: 1px solid var(--pag-color-bg-tertiary);
41      padding: 5px;
42      margin-bottom: 5px;
43  }
44
45  .pag-resultgit-content h5 {
46      color: var(--pag-color-font-primary);
47      margin-right: 10px;
48  }
49
50  .pag-resultgit-content h6 {
51      color: var(--pag-color-font-secondary);
52  }
```

Visualização web



GitHub-8

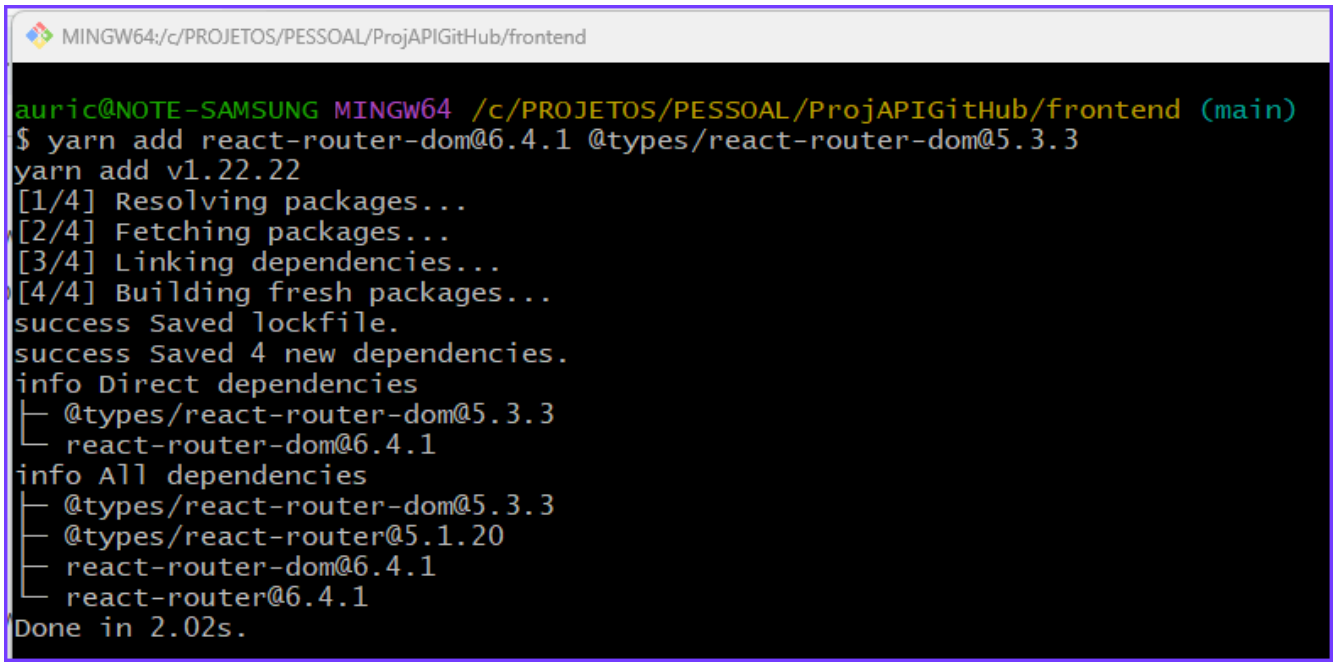
- COMPONENTE RESULT_GIT



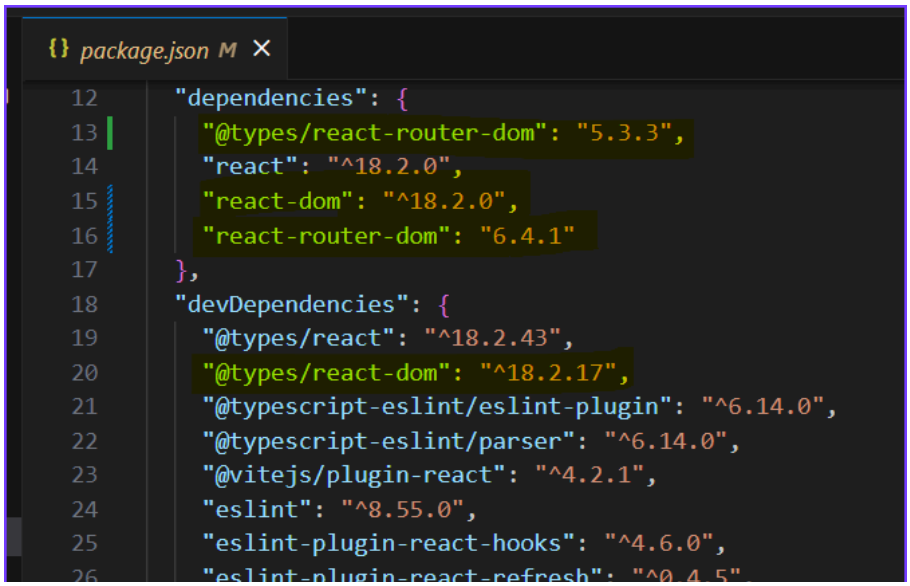
10 - ROTAS

INSTALAR O REACT-ROUTER-DOM

- yarn add react-router-dom@6.4.1 @types/react-router-dom@5.3.3



VERIFICAR NO PROJETO



CRIAR O ROTES

Implementar as rotas das páginas, no main

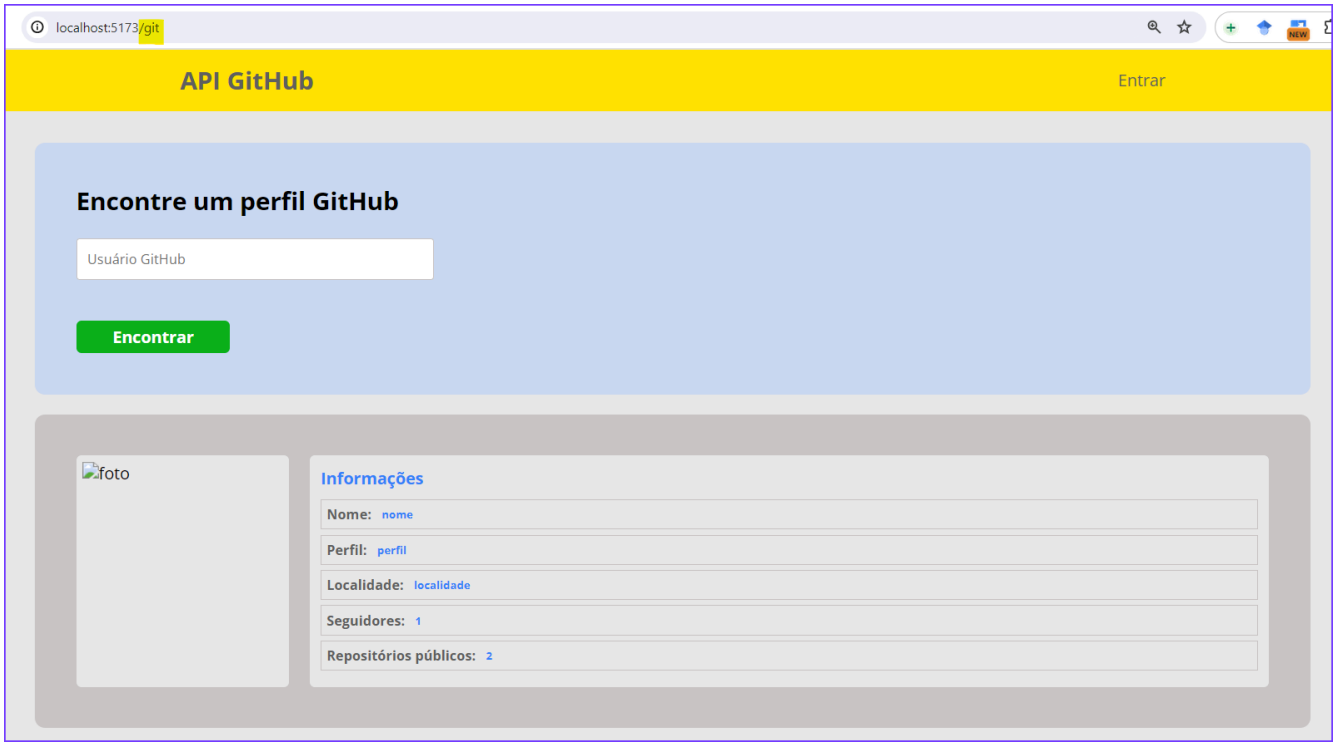
src > main.tsx

```tsx
import ReactDOM from 'react-dom/client'
import App from './App.tsx'
import './index.css'
import { BrowserRouter, Route, Routes } from 'react-router-dom'
import GitSearch from './pages/GitSearch/index.tsx'
import Home from './pages/Home/index.tsx'
import Header from './components/Header/index.tsx'

ReactDOM.createRoot(document.getElementById('root')!).render(



    <BrowserRouter>

        <Header />

        <Routes>
            <Route path="/" element={<App />} />
            <Route index element={<Home />} />
            <Route path="git" element={<GitSearch />} />
        </Routes>
    </BrowserRouter>
)
```

Implementar o App

```
src > App.tsx > ...
1  import { Outlet } from "react-router-dom";
2  import "./App.css";
3
4  function App() {
5
6    return (
7
8      <Outlet />
9
10   );
11 }
12 export default App;
13
```

Visualização web



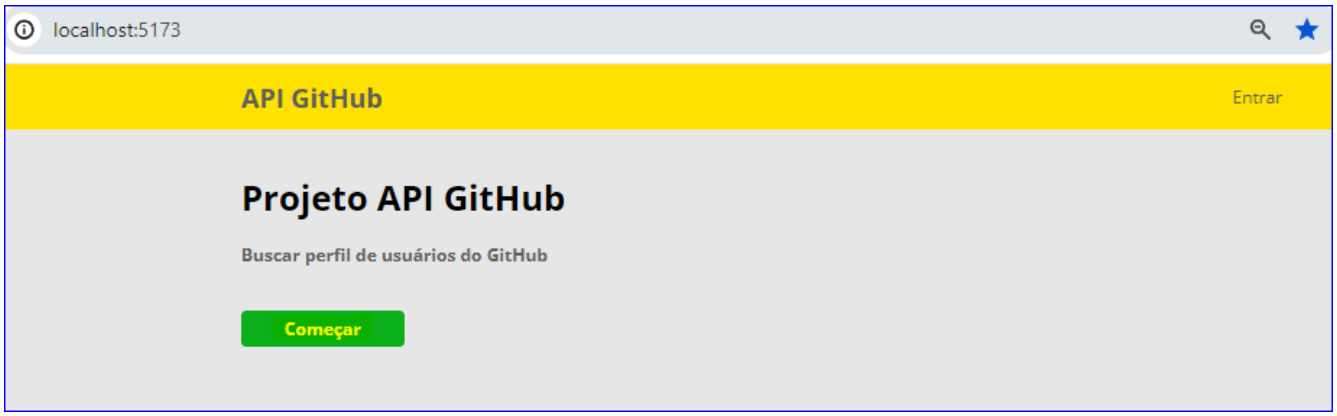
Implementar o link para os botões

Botão “Começar”, do Home

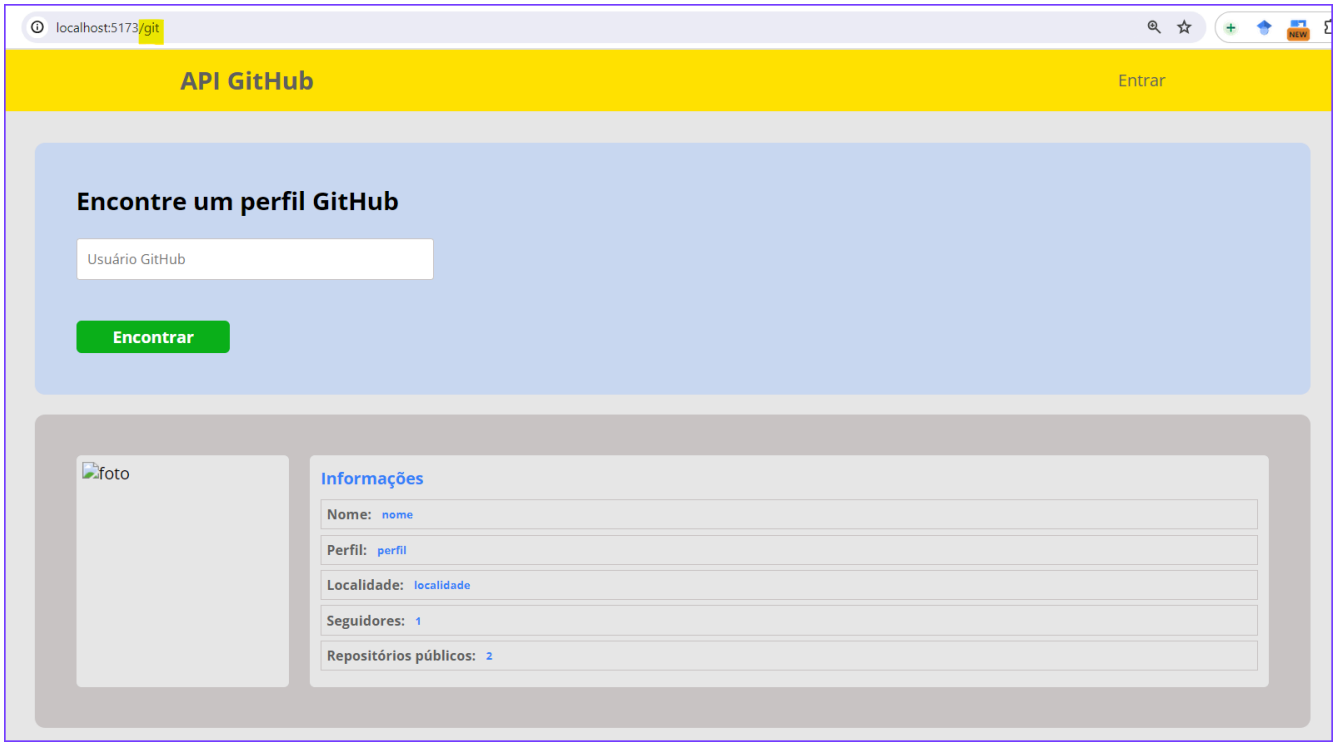
Nome “API GitHub” e “Entrar ”, do Header

```
src > components > Header > index.tsx > ...
  6       return (
  7           <header className="pag-header">
  8               <nav className="pag-container">
  9                   <div className="pag-header-navbar">
 10                       <Link to="/">
 11                           <h1>API GitHub</h1>
 12                       </Link>
 13                       <div>
 14                           <Link to="/">
 15                               <a href="#">Entrar</a>
 16                           </Link>
 17                       </div>
 18                   </div>
 19               </nav>
 20           </header>
 21       );
```

Visualização web



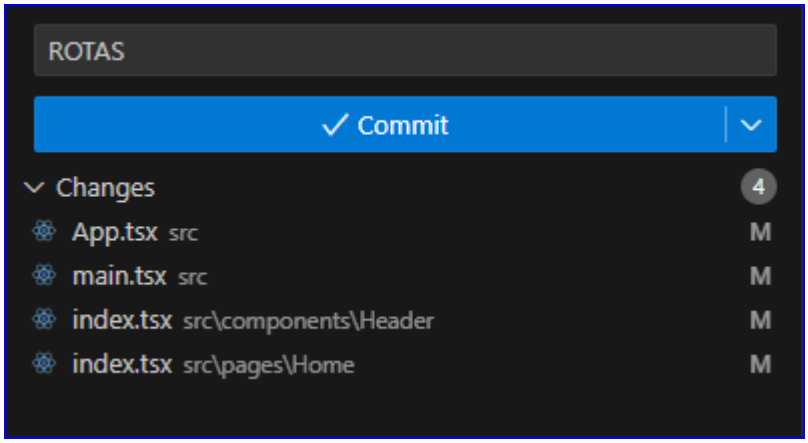
***NOTA**: Ao clicar no botão “Começar”, será redirecionado para a página “/git”.*



***NOTA**: Ao clicar no nome "API GitHub" e "Entrar" , será redirecionado para a página "/".*

GitHub-9

- ROTAS



11 - TRATAR EVENTOS DE FORMULÁRIO

IMPLEMENTAR O INPUT, NO GIT_SEARCH

Implementar o onChange

src > pages > GitSearch > index.tsx > ...

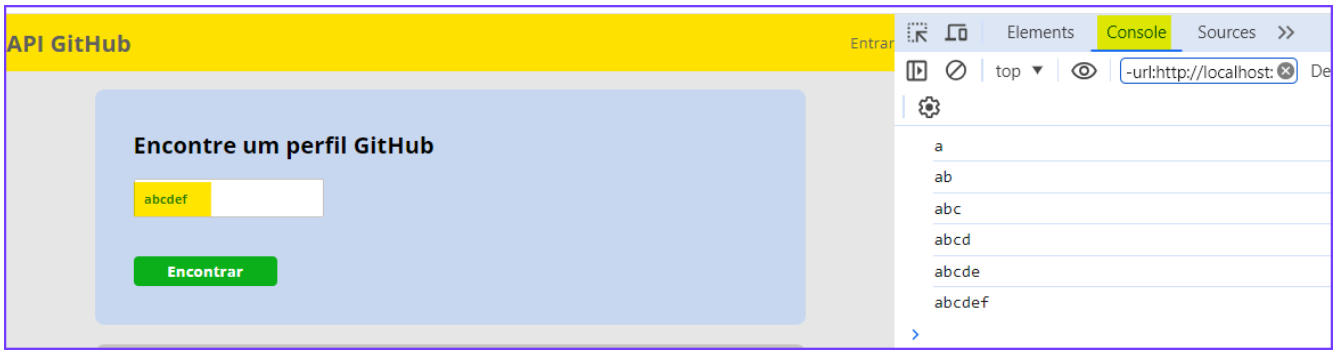
```
13                  <form className="pag-mb40">
14                      <input
15                          type="text"
16                          className="pag-gitsearch-input"
17                          placeholder="Usuário GitHub"
18                          onChange={handleChange}
19                      />
20                  </form>
```

Implementar a função handleChange

src > pages > GitSearch > index.tsx > ...

```
4
5   export default function GitSearch() {
6
7       function handleChange(event: any) {
8           console.log(event.target.value)
9       }
10
11      return (
```

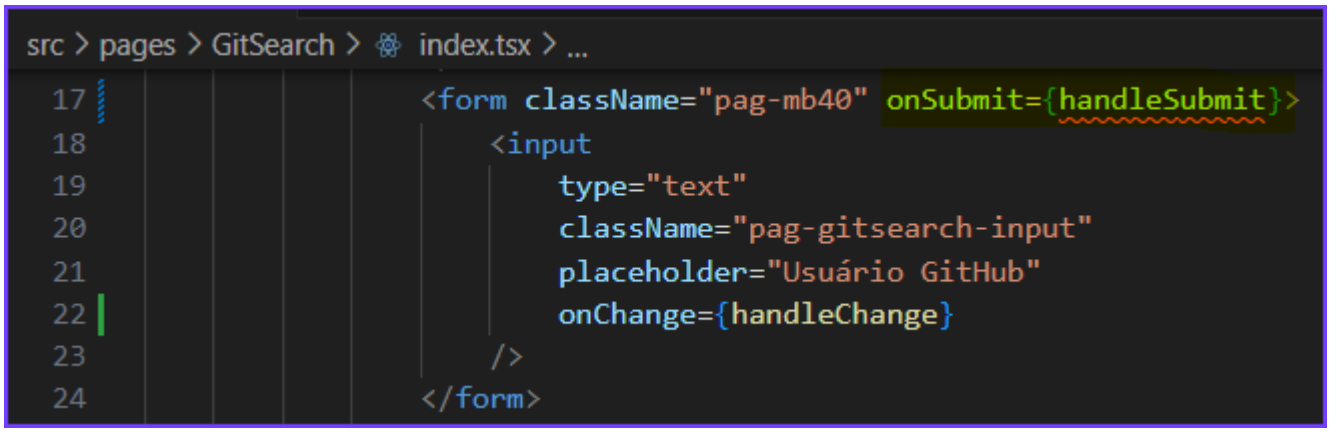
Visualização web



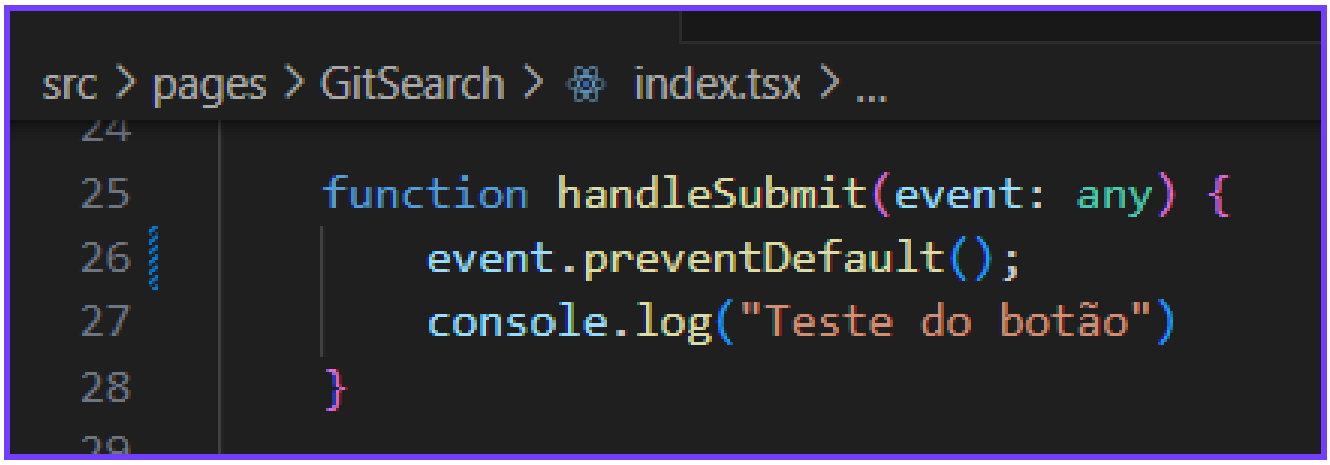
***NOTA**: Com o "console.log", podemos ver que ao digitar é mostrado no Console do navegador.*

IMPLEMENTAR O SUBMIT, NO GIT_SEARCH

Implementar o onSubmit



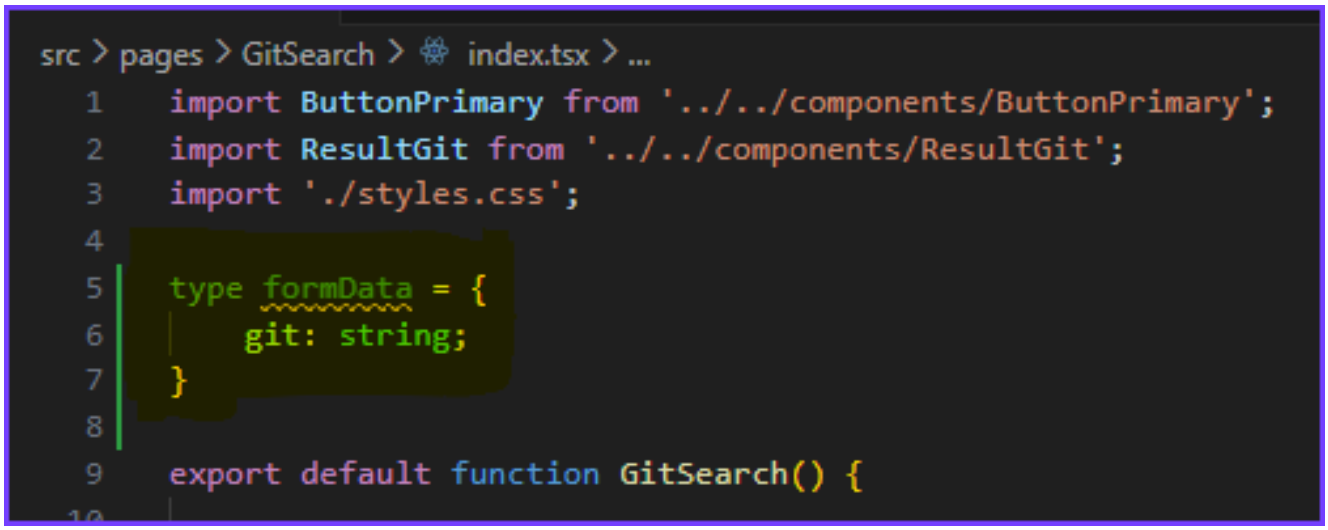
Implementar a função handleSubmit



***NOTA**: Sempre iniciamos com o event.preventDefault(), pois evita que o formulário reinicialize. Sem esse parâmetro, o componente não segura o estado no momento da pesquisa.*

CONTROLANDO O ESTADO DOS VALORES

Implementar o formData



Implementar o useState

src > pages > GitSearch > index.tsx > ...

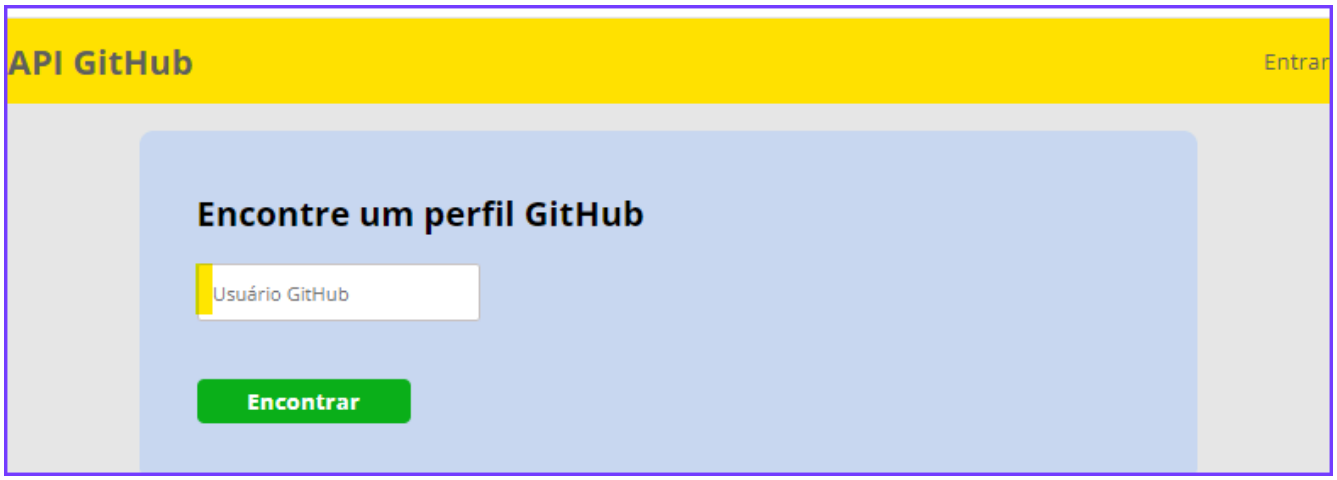
```tsx
export default function GitSearch() {

    const [ formData,  setFormData ] = useState<formData>({

        git: "",

    });
```

Implementar o input

src > pages > GitSearch > index.tsx > ...

```tsx
<form onSubmit={handleSubmit}>
    <div className="pag-mb40">
        <input
            value={formData.git}
            type="text"
            name="git"
            className="pag-gitsearch-input"
            placeholder="Usuário GitHub"
            onChange={handleChange}
        />
    </div>
    <div className="pag-gitsearch-btn">
        <ButtonPrimary name="Encontrar" />
    </div>
</form>
```

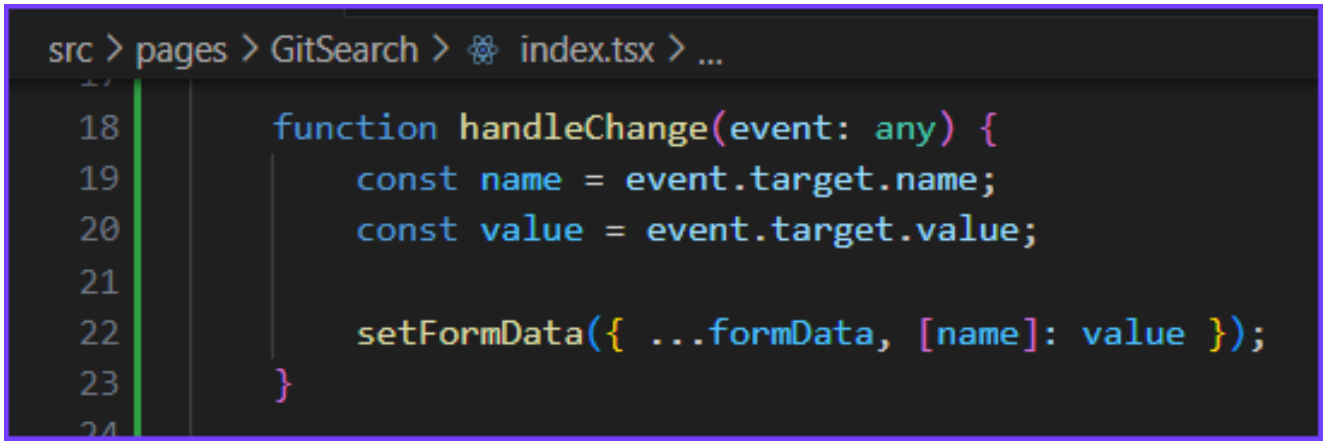
Visualização web



***NOTA**: Ao tentar digitar algo, nada acontece, visto que não colocamos o método para alterar o estado da função (setFormData).*

ALTERANDO O ESTADO DA FUNÇÃO

Implementar a função handleChange



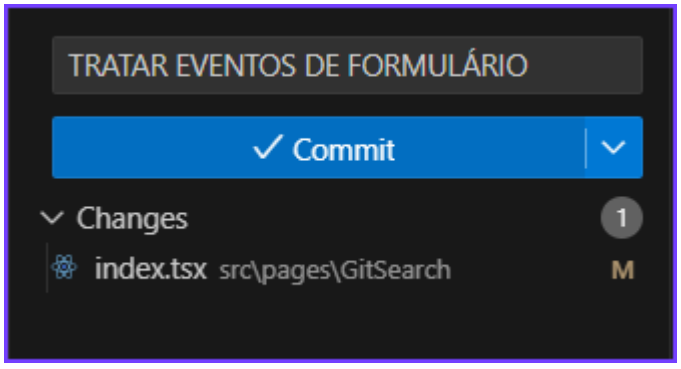
Implementar a função handleSubmit

```
src > pages > GitSearch > index.tsx > ...
24
25      function handleSubmit(event: any) {
26          event.preventDefault();
27          console.log(formData)
28      }
```

Visualização web

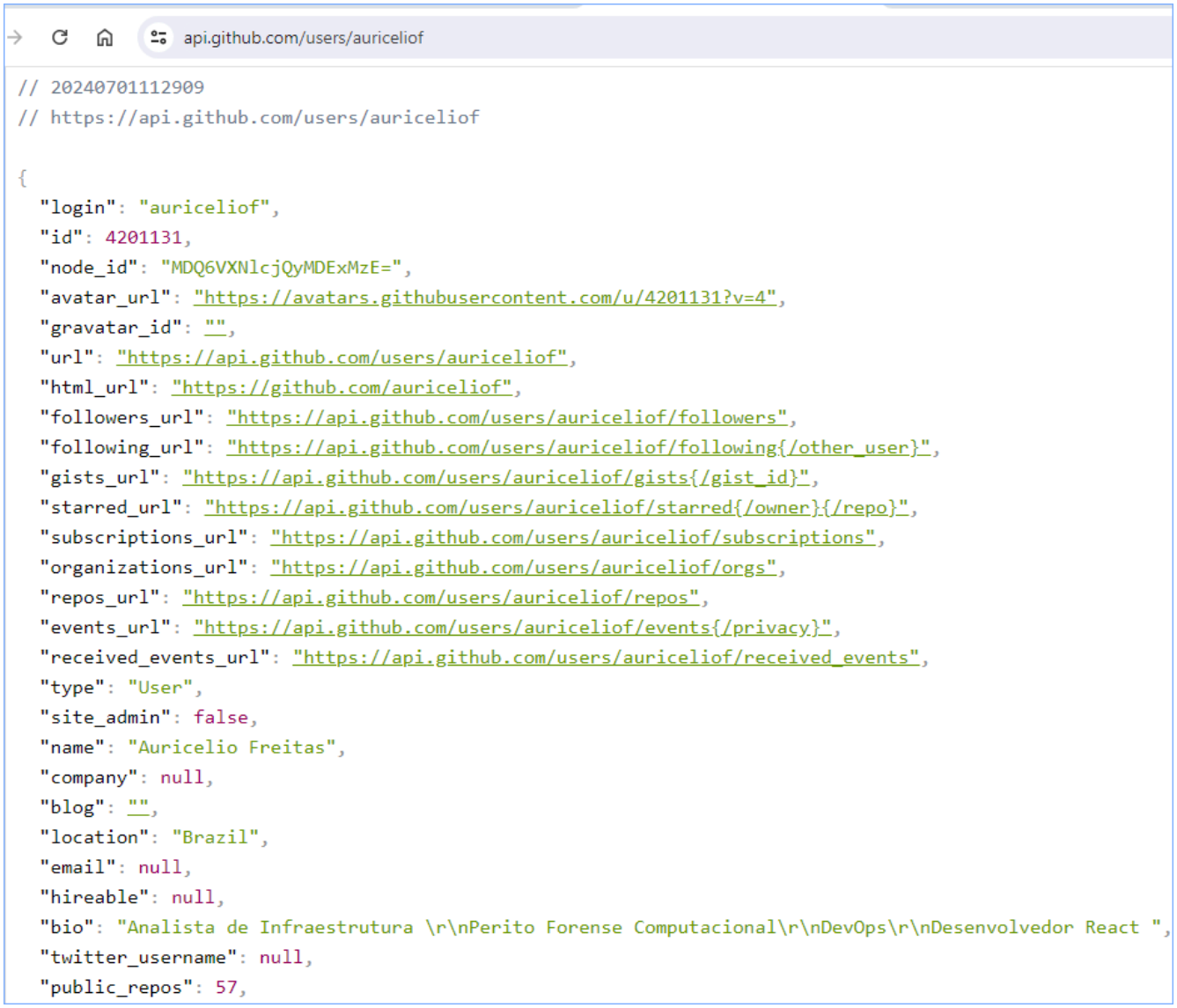
GitHub-10

- TRATAR EVENTOS DE FORMULÁRIO



12 - INTEGRANDO COM A API GITHUB

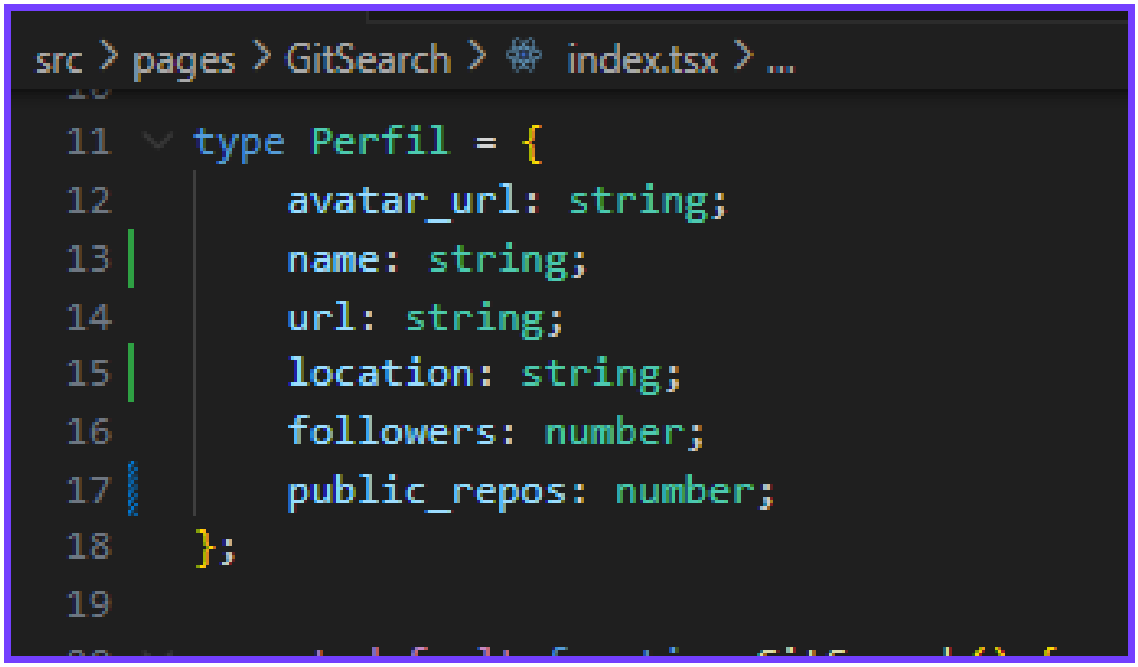
- GitHub: https://api.github.com/users/auriceliof



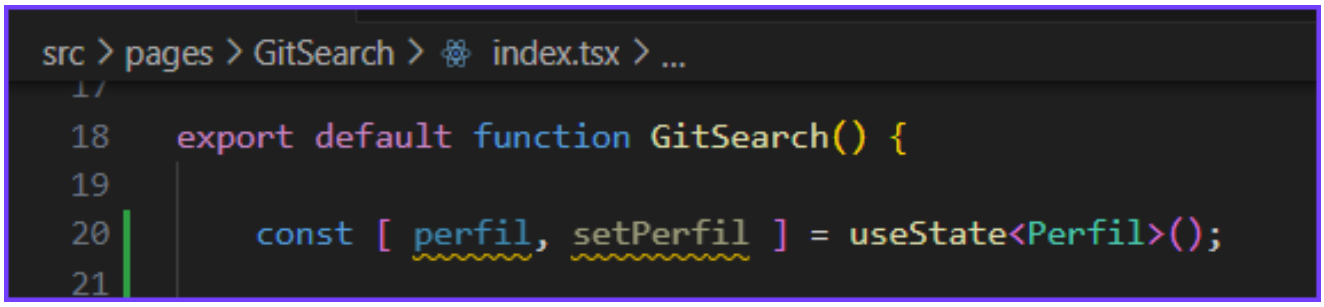
***NOTA**: Ao integrar com uma API, devemos atentar para o tipo de dados que a mesma retorna.*

INTEGRAR COM A API

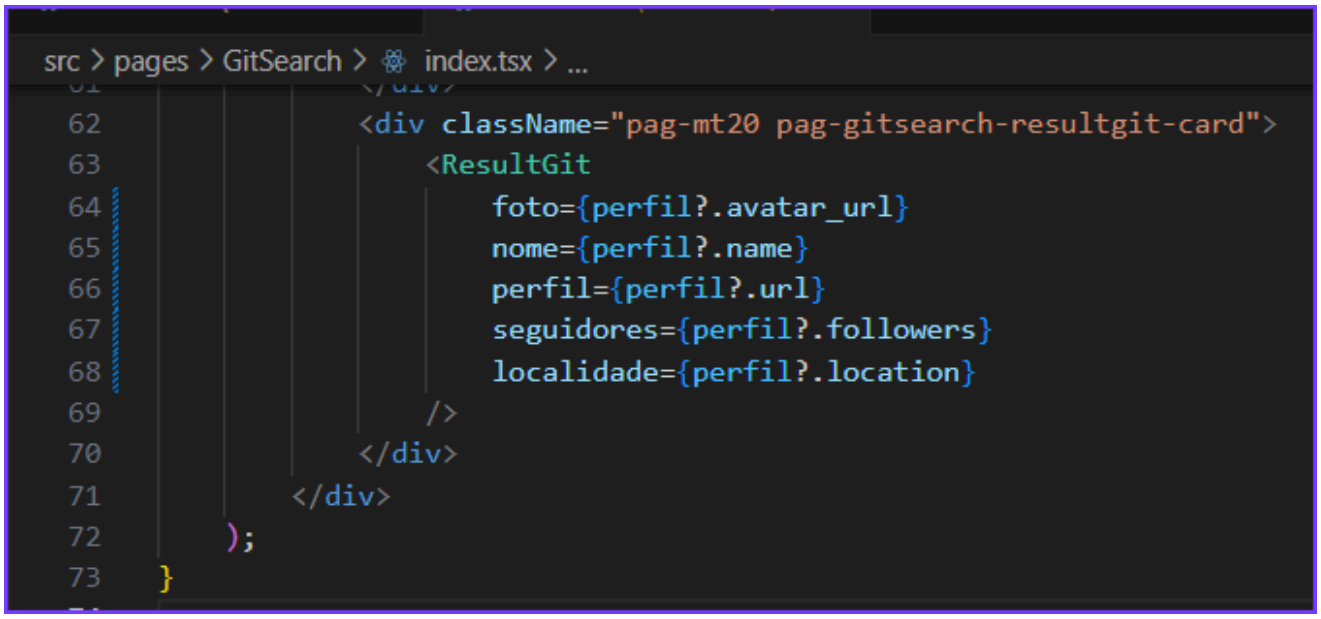
Implementar o tipo



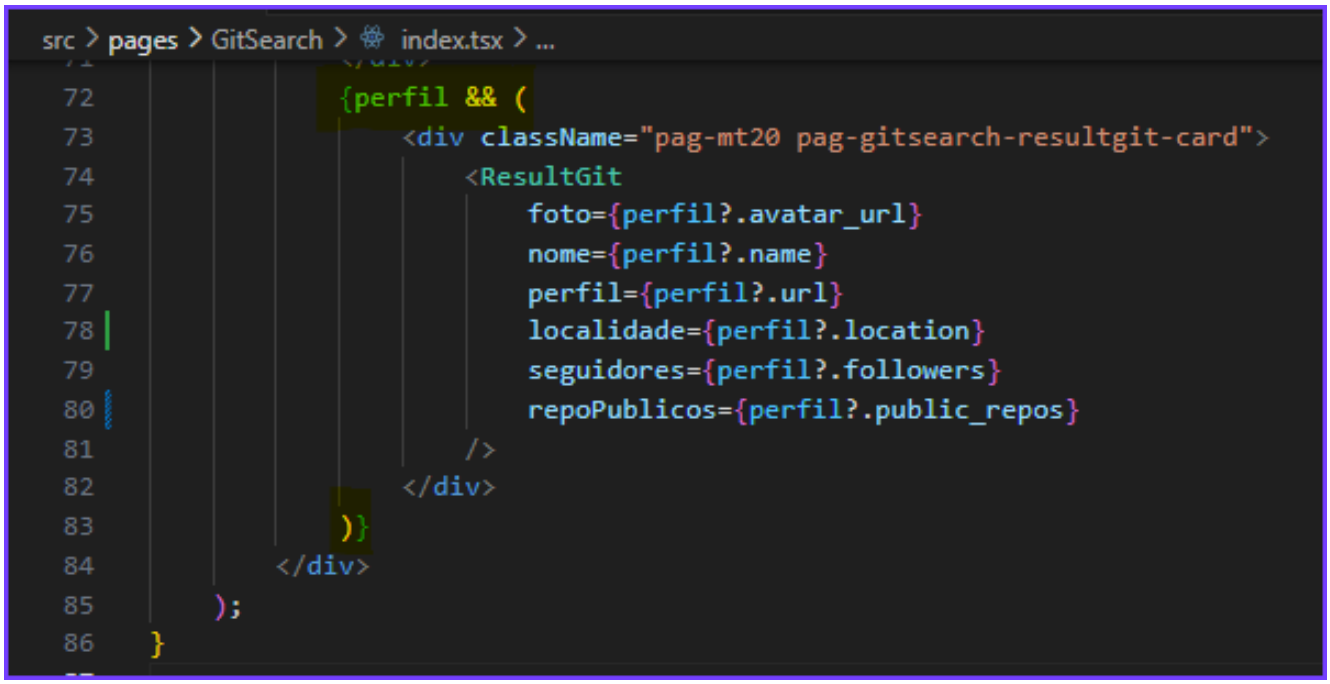
Implementar o useState



Implementar o ResultGti, no GitSearch



Implementar um condicional para renderizar o card



AXIOS

Instalar o Axios

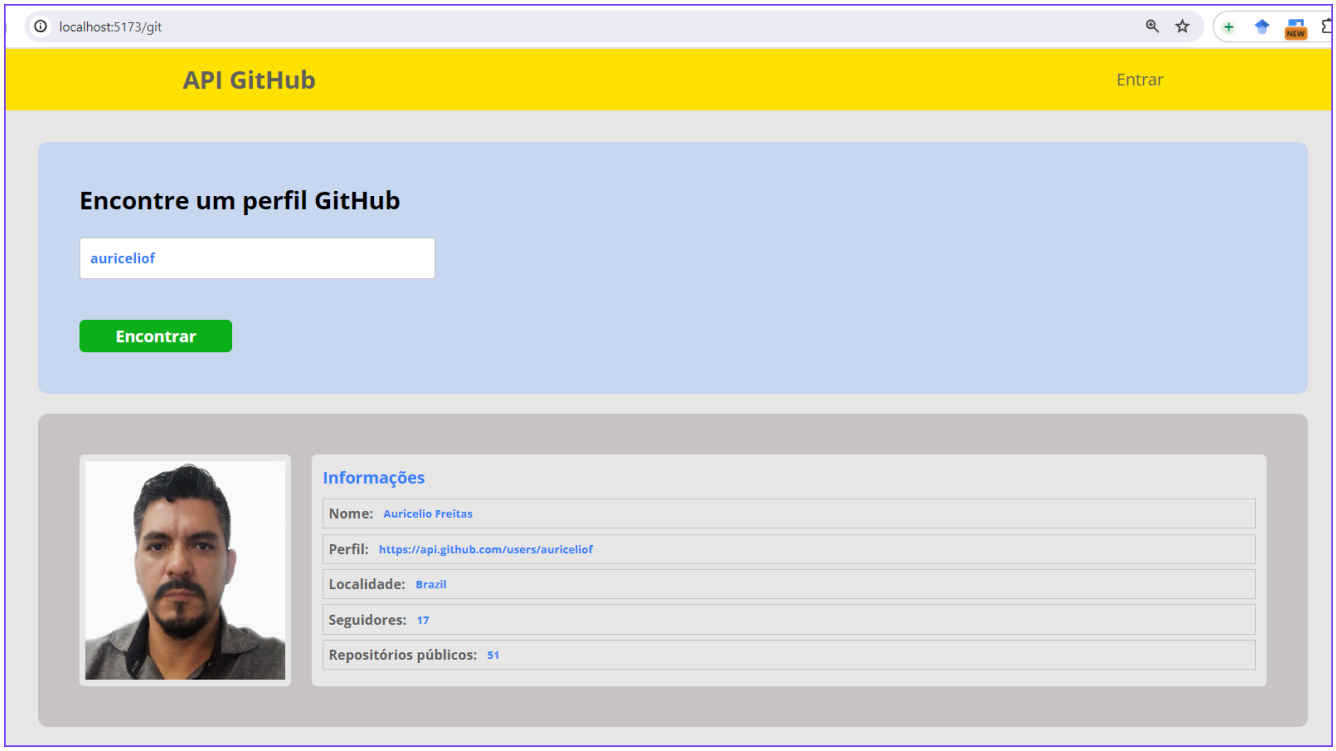
```
MINGW64:/c/PROJETOS/PESSOAL/ProjAPIGitHub/frontend

auric@NOTE-SAMSUNG MINGW64 /c/PROJETOS/PESSOAL/ProjAPIGitHub/frontend (main)
$ yarn add axios
yarn add v1.22.22
[1/4] Resolving packages...
[2/4] Fetching packages...
[3/4] Linking dependencies...
[4/4] Building fresh packages...
success Saved lockfile.
success Saved 9 new dependencies.
info Direct dependencies
└─ axios@1.6.8
info All dependencies
├─ asynckit@0.4.0
├─ axios@1.6.8
├─ combined-stream@1.0.8
├─ delayed-stream@1.0.0
├─ follow-redirects@1.15.6
├─ form-data@4.0.0
├─ mime-db@1.52.0
├─ mime-types@2.1.35
└─ proxy-from-env@1.1.0
Done in 3.00s.
```

Implementar o axios, no gitSearch

src > pages > GitSearch > index.tsx > ...

```
35
36      function handleSubmit(event: any) {
37          event.preventDefault();
38
39          axios.get(`https://api.github.com/users/${formData.git}`)
40              .then((response) => {
41                  setPerfil(response.data);
42              })
43      }
```

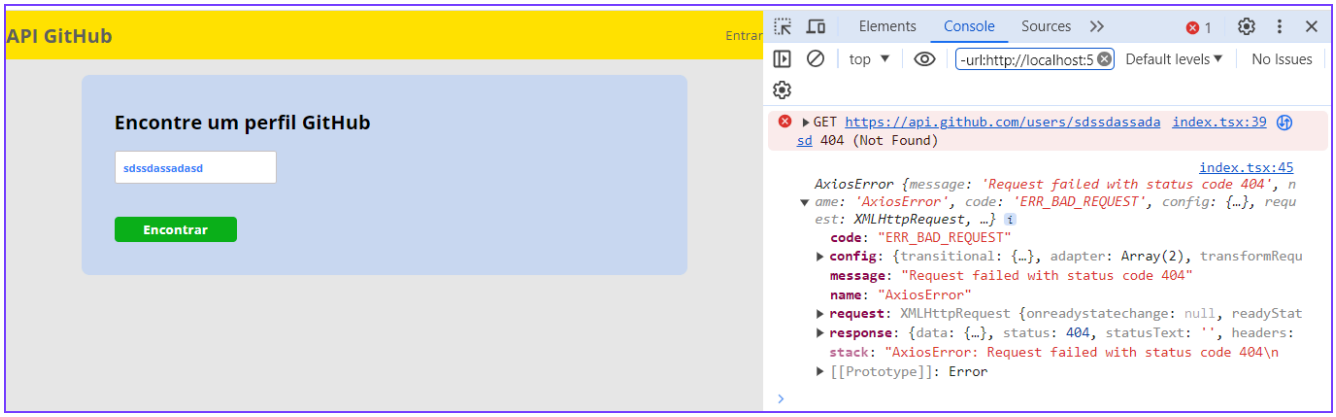
Visualização web



TRATANDO ERRO (PERFIL INVÁLIDO)

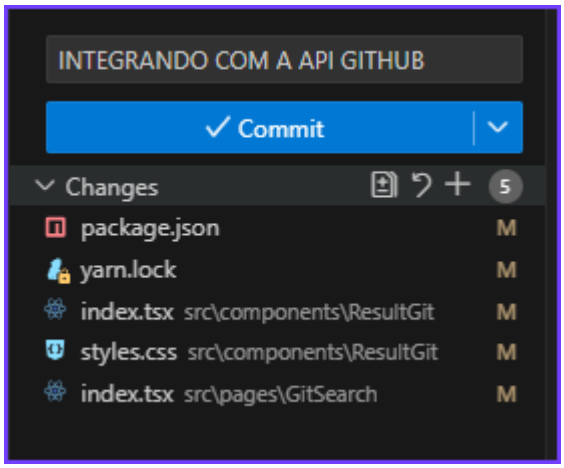
Implementar o catch

Visualização web



GitHub-11

- INTEGRANDO COM A API GITHUB



13 - REPOSITÓRIOS DO PERFIL, COMO COMPONENTE

COMPONENTE RESULT_GIT_REPO

Criar e implementar o componente ResultGitRepo

src > components > ResultGitRepo > index.tsx > ...

```tsx
import './styles.css';

type Props = {
    nomeRepo?: string;
    urlRepo?: string;
    descRepo?: string;
}

export default function ResultGitRepo({nomeRepo, urlRepo, descRepo}: Props) {

    return (
        <div className="pag-resultrepo-container">
            <div className="pag-resultrepo-card">
                <div className="pag-resultrepo-information">
                    <div className="pag-resultgit-content">
                        <h5>Nome: </h5>
                        <h6>{nomeRepo}</h6>
                    </div>
                    <div className="pag-resultgit-content">
                        <h5>Repositório: </h5>
                        <h6>{urlRepo}</h6>
                    </div>
                    <div className="pag-resultgit-content">
                        <h5>Descrição: </h5>
                        <h6>{descRepo}</h6>
                    </div>
                </div>
            </div>
        </div>
    );
}
```

Implementar o estilo do ResultGitRepo

src > components > ResultGitRepo > styles.css > ...

```css
.pag-resultrepo-card {
    display: flex;
}

.pag-resultrepo-information {
    width: 100%;
    padding: 10px;
    background: var(--pag-color-bg-primary);
    border: 1px solid var(--pag-color-bg-primary);
    border-radius: 5px;
}

.pag-resultrepo-information h4 {
    color: var(--pag-color-font-secondary);
    margin-bottom: 10px;
}

.pag-resultrepo-content {
    display: flex;
    align-items: center;
    border: 1px solid var(--pag-color-bg-tertiary);
    padding: 5px;
    margin-bottom: 5px;
}

.pag-resultrepo-content h5 {
    color: var(--pag-color-font-primary);
    margin-right: 10px;
}

.pag-resultrepo-content h6 {
    color: var(--pag-color-font-secondary);
}
```

Implementar o estilo do card, no GitSearch

src > pages > GitSearch > styles.css > ...

```
36
37  .pag-gitsearch-resultRepos-card {
38      width: 100%;
39      background: var(--pag-color-bg-quinternary);
40      border-radius: 10px;
41      padding: 20px 40px;
42  }
43
44  .pag-gitsearch-resultRepos-card h4{
45      display: flex;
46      justify-content: center;
47      color: var(--pag-color-font-primary);
48  }
```

CHAMADA NA PAGINA GIT_SEARCH

Implementar o tipo

src > pages > GitSearch > index.tsx > ...

```
20
21  type Repos = {
22      id: number;
23      name: string;
24      html_url: string;
25      description: string;
26  }
27
```

Implementar o useState para os repositórios

```
src > pages > GitSearch > index.tsx > ...
28    export default function GitSearch() {
29
30        const [ perfil, setPerfil ] = useState<Perfil>();
31
32        const [ repo, setRepo ] = useState<Repos[]>([]);
33
```

Implementar o handleSubmit para os repositórios

```
src > pages > GitSearch > index.tsx > ...
49        function handleSubmit(event: any) {
50            event.preventDefault();
51
52            axios.get(`https://api.github.com/users/${formData.git}`)
53                .then((response) => {
54                    setPerfil(response.data);
55                })
56                .catch((error) => {
57                    setPerfil(undefined);
58                    console.log(error);
59                })
60
61            axios.get(`https://api.github.com/users/${formData.git}/repos`)
62            .then((response) => {
63                console.log(response.data)
64                setFindRepo(response.data);
65                setRepo(response.data);
66            })
67        }
```

Implementar o map e a chamada do ResultGitRepo

src > pages > GitSearch > index.tsx > ...

```
                )}

                <div className="pag-mt20 pag-gitsearch-resultRepos-card">
                    <h4>Repositórios Público</h4>
                    {repo.map((product) =>  (
                        <div key={product.id} className="pag-mt20">
                            <ResultGitRepo
                                nomeRepo={product?.name}
                                urlRepo={product?.html_url}
                                descRepo={product?.description}
                            />
                        </div>
                    ))}
                </div>
```

Implementar o useState para o condicional do ResultGitRepo

src > pages > GitSearch > index.tsx > ...

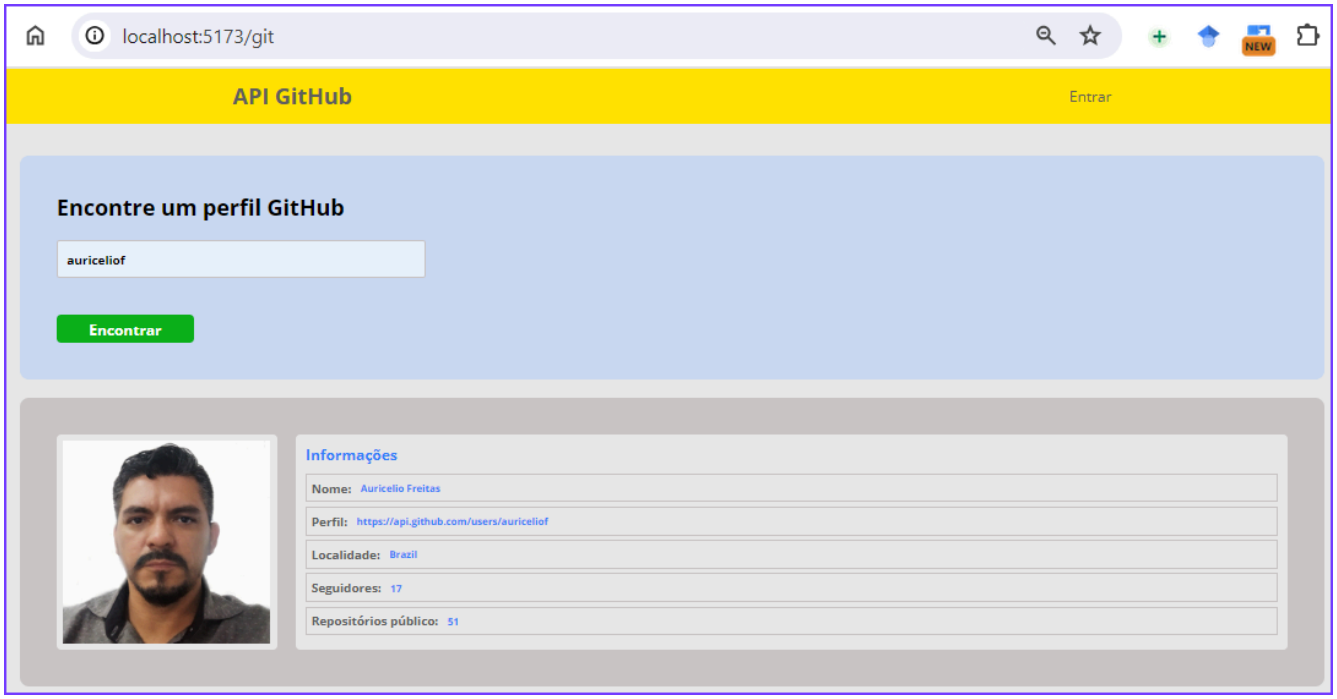
```
export default function GitSearch() {

    const [ perfil, setPerfil ] = useState<Perfil>();

    const [ repo, setRepo ] = useState<Repos[]>([]);

    const [ findRepo, setFindRepo ] = useState<Repos>();

```

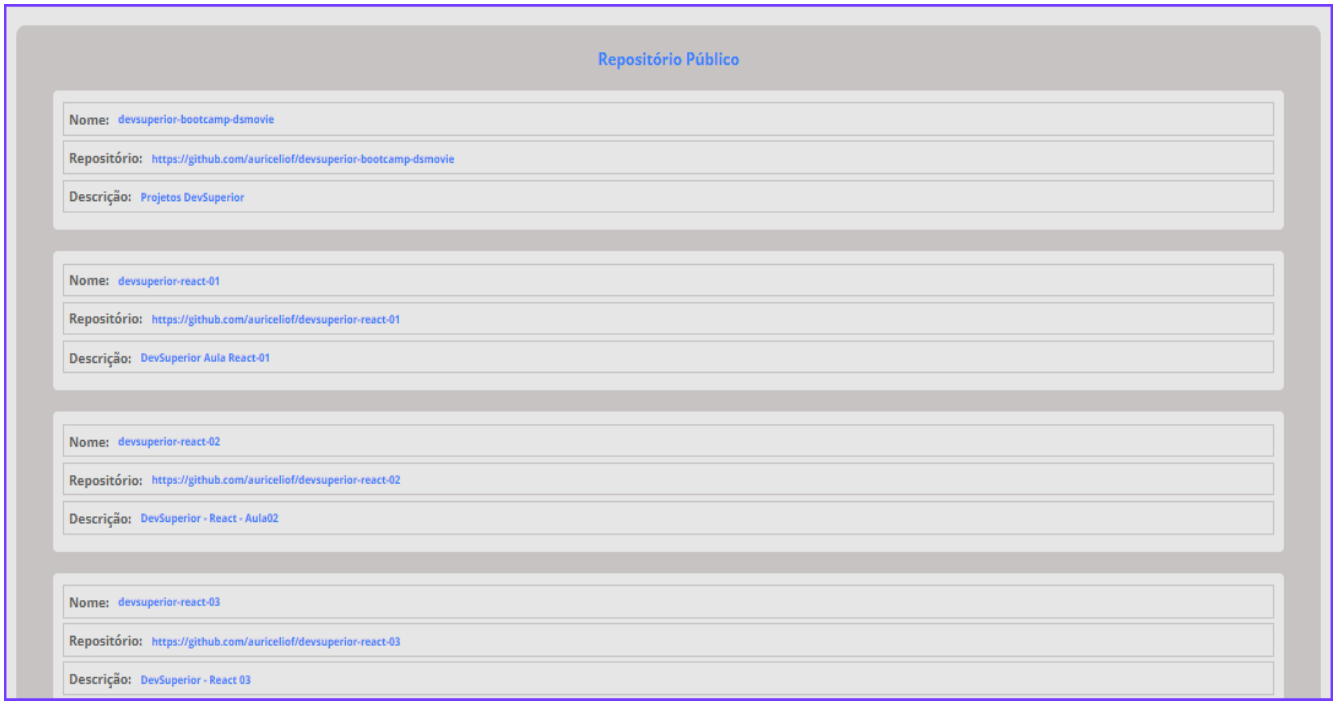
Implementar o condicional findRepo

src > pages > GitSearch > index.tsx > GitSearch

```
103                 {findRepo &&
104                     <div className="pag-mt20 pag-gitsearch-resultRepos-card">
105                         <h4>Repositório Público</h4>
106                         {repo.map((product) =>  (
107                             <div key={product.id} className="pag-mt20">
108                                 <ResultGitRepo
109                                     nomeRepo={product?.name}
110                                     urlRepo={product?.html_url}
111                                     descRepo={product?.description}
112                                 />
113                             </div>
114                         ))}
115                     </div>
116                 }
117
118             </div>
119         );
120     }
```

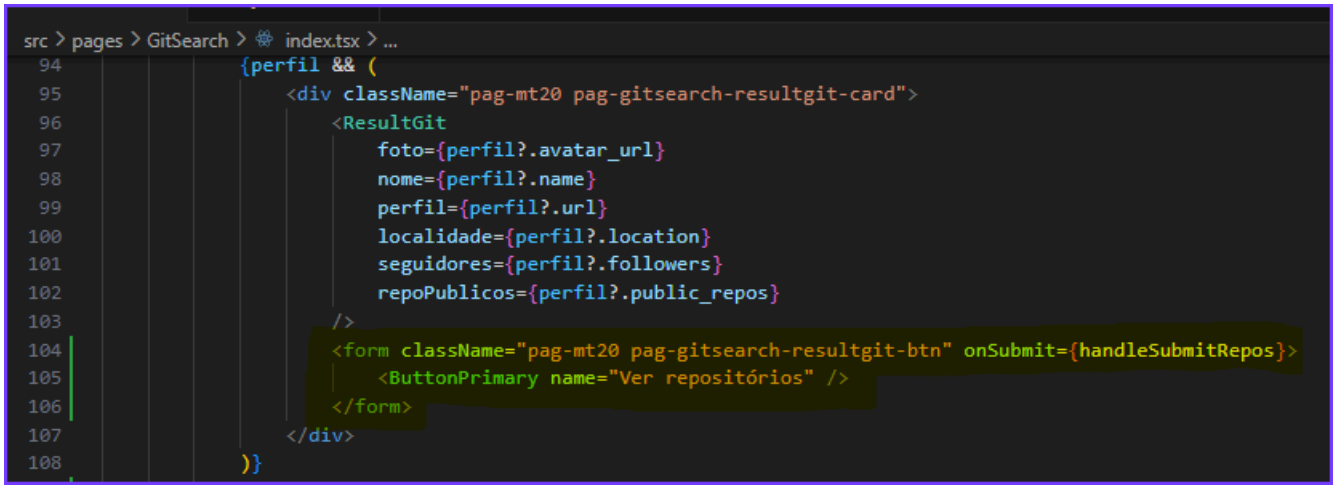
Visualização web



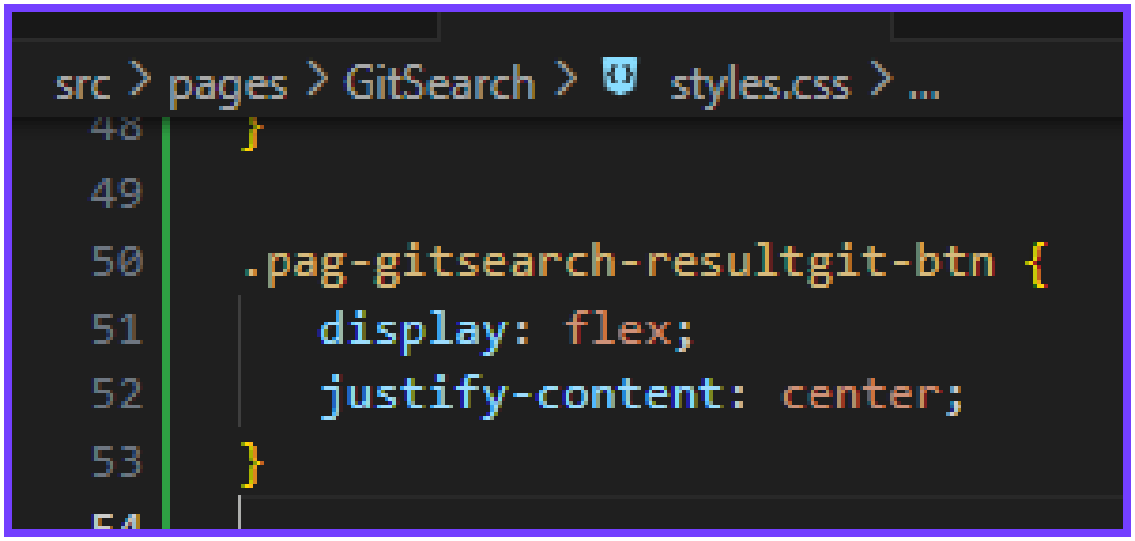


BOTÃO DO REPOSITORIO

Implementar o ButtonPrimary, no ResultSearch



Implementar estilo

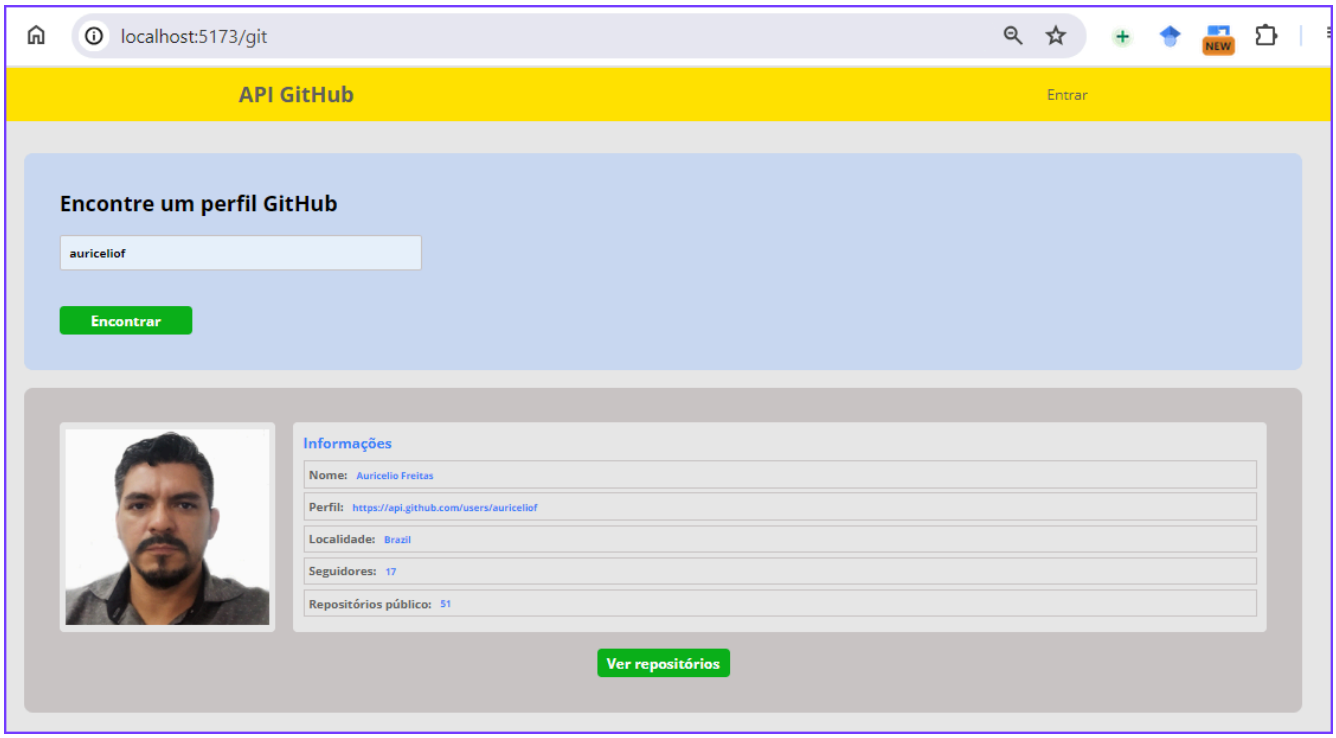


Implementar a função handleSubmitRepos

src > pages > GitSearch > index.tsx > ...

```
48
49      function handleSubmit(event: any) {
50          event.preventDefault();
51
52          axios.get(`https://api.github.com/users/${formData.git}`)
53              .then((response) => {
54                  setPerfil(response.data);
55              })
56              .catch((error) => {
57                  setPerfil(undefined);
58                  console.log(error);
59              })
60      }
61
62      function handleSubmitRepos(event: any) {
63          event.preventDefault();
64
65          axios.get(`https://api.github.com/users/${formData.git}/repos`)
66          .then((response) => {
67              setFindRepo(response.data);
68              setRepo(response.data);
69          })
70      }
```

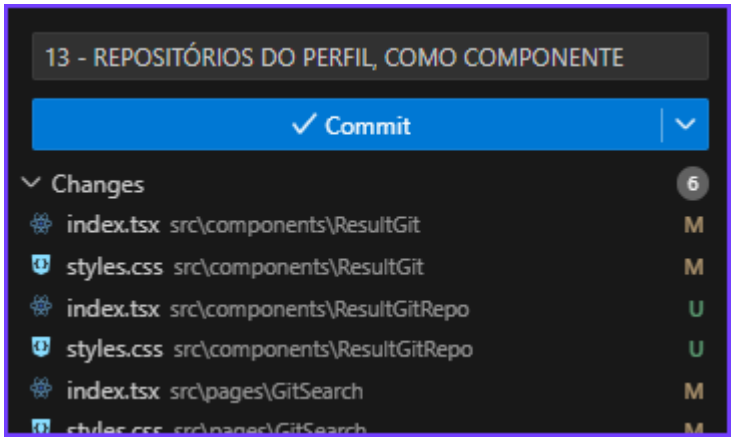
Visualização web



***NOTA**: Só aparecerá os repositórios caso clique no botão "Ver repositórios".*

GitHub-12

- 13 - REPOSITÓRIOS DO PERFIL, COMO COMPONENTE



FIM