

Guia Completo: Criação e Deploy da Área de Downloads no Portfólio

Este guia detalhado aborda a criação da página de downloads, a estrutura de código (baseada na análise do seu projeto React/JavaScript) e o processo de deploy no GitHub Pages.

1. Estrutura de Arquivos e Pastas

Para que os links de download funcionem corretamente, é essencial que os arquivos estejam no local esperado pelo seu código.

1.1. Criação da Pasta de Arquivos

No diretório raiz do seu projeto (ou na pasta `public/` se você estiver usando um *framework* como React/Vite), crie a pasta que contém os arquivos de download:

Plain Text

```
/seu-repositorio-portifolio
├── ...
├── /arqs_nn <-- Crie esta pasta
│   ├── Form_Monit_Cht_V5.0.1.xlsx
│   └── Form_Monit_Lgc_V5.0.1.xlsx
└── ...
```

Caminho de Acesso: O código espera que os arquivos sejam acessíveis via `./arqs_nn/nome_do_arquivo.xlsx`.

2. Criação do Componente da Página de Downloads

Com base na análise do seu código JavaScript empacotado, o componente responsável pela página de downloads (`X3e` no código original) segue a lógica abaixo. Você deve replicar essa lógica em um novo arquivo, por exemplo, `DownloadPage.jsx` ou `DownloadPage.js`.

2.1. Definição dos Dados de Download

Crie um array de objetos que contenha o nome a ser exibido e o link para o arquivo.

JavaScript

```
// Exemplo de código para o seu componente de página de downloads
```

```
const downloadItems = [  
  {  
    nome: "📊 Form_Monit_Cht_V5.0.1",  
    link: "./arqs_nn/Form_Monit_Cht_V5.0.1.xlsx"  
  },  
  {  
    nome: "📊 Form_Monit_Lgc_V5.0.1",  
    link: "./arqs_nn/Form_Monit_Lgc_V5.0.1.xlsx"  
  }  
];
```

2.2. Estrutura do Componente (React/JSX)

O componente deve mapear o array `downloadItems` para criar os links de download. O uso do atributo `download` no elemento `<a>` é crucial para forçar o navegador a baixar o arquivo em vez de tentar abri-lo.

JavaScript

```
import React from 'react';  
  
// Importe outros componentes de layout (X0, Y0, wV) se necessário  
  
function DownloadPage() {  
  const downloadItems = [  
    {  
      nome: "📊 Form_Monit_Cht_V5.0.1",  
      link: "./arqs_nn/Form_Monit_Cht_V5.0.1.xlsx"  
    },  
    {  
      nome: "📊 Form_Monit_Lgc_V5.0.1",  
      link: "./arqs_nn/Form_Monit_Lgc_V5.0.1.xlsx"  
    }  
  ];  
  
  return (  
    <div className="DownloadPage relative aurora-background min-h-screen  
flex items-center justify-center">  
      { /* Componentes de Layout (se houver) */ }  
      { /* <X0 /> */ }  
      { /* <Y0 /> */ }  
      { /* <wV /> */ }  
  
      <main className="relative z-10 bg-white rounded-lg shadow-lg max-  
w-2xl w-full p-8 text-center">
```

```

        <h1 className="text-4xl font-bold mb-6 text-gray-800">Área
de Downloads</h1>
        <p className="mb-10 text-lg text-gray-600">Baixe aqui os
arquivos e materiais disponíveis no site.</p>

        <div className="grid grid-cols-1 sm:grid-cols-2 gap-6">
          {downloadItems.map((item, index) => (
            <a
              key={index}
              href={item.link}
              download={true} // Força o download do arquivo
              className={`bg-gradient-to-r from-blue-600 to-
blue-800
                                hover:from-blue-700 hover:to-blue-
900
                                text-white py-4 px-6 rounded-lg
                                shadow-lg
                                transition btn-shine subtle-float`}
            >
              {item.nome}
            </a>
          ))}
        </div>
      </main>
    </div>
  );
}

export default DownloadPage;

```

3. Configuração de Roteamento (Acesso à Página)

Seu portfólio utiliza roteamento baseado em *hash* (`#/downloads`), o que é comum em aplicações de página única (SPA) hospedadas no GitHub Pages.

3.1. Adicionar a Rota

No seu arquivo principal de roteamento (provavelmente onde você configura o `react-router-dom`), adicione a nova rota para a página de downloads.

JavaScript

```

// Exemplo de configuração de rotas (adaptado do seu código)

import { BrowserRouter as Router, Routes, Route } from 'react-router-dom';
import HomePage from './HomePage'; // Seu componente da página inicial

```

```
import DownloadPage from './DownloadPage'; // O componente que você acabou
de criar

// ... outros imports

function App() {
  return (
    <Router>
      <Routes>
        { /* Rota para a página inicial */ }
        <Route path="/" element={<HomePage />} />

        { /* Rota para a nova página de downloads */ }
        <Route path="/downloads" element={<DownloadPage />} />

        { /* ... outras rotas */ }
      </Routes>
    </Router>
  );
}

// ...
```

4. Processo de Deploy no GitHub Pages

4.1. Build Local

Execute o comando de build do seu projeto para gerar os arquivos estáticos de produção.

Bash

```
npm run build
# ou
yarn build
```

4.2. Commit e Push

Certifique-se de que a nova pasta `arqs_nn/` e o novo código do componente de downloads estejam incluídos no seu *commit*.

1. **Adicione as alterações:**
2. **Faça o commit:**
3. **Envie para o GitHub:**

4.3. Verificação Final

Após o deploy do GitHub Pages ser concluído (pode levar alguns minutos):

1. Acesse a página de downloads: <https://jmc-codes.github.io/jmc.portfolio/#/downloads>
2. Verifique se a página carrega corretamente.
3. Clique nos botões para confirmar que o download dos arquivos `.xlsx` é iniciado.

Referências

- [1] [jmc-codes.github.io/jmc.portfolio/#/downloads](<https://jmc-codes.github.io/jmc.portfolio/#/downloads>) - URL da Área de Downloads.
- [2] [Documentação do React Router](<https://reactrouter.com/en/main/start/overview>) - Guia para configurar rotas em aplicações React.
- [3] [Documentação do GitHub Pages](<https://docs.github.com/pt/pages>) - Informações sobre como hospedar sites estáticos no GitHub.