**Guia Completo: Deploy de Aplicação React/Vite para GitHub Pages**

Este guia detalha o processo de preparar seu ambiente de desenvolvimento React/Vite, configurar o Git, criar um repositório no GitHub e implantar seu aplicativo usando GitHub Pages e GitHub Actions.

**Pré-requisitos**

Antes de começar, certifique-se de ter o seguinte instalado em seu sistema:

* **Node.js e npm (ou Yarn):** Para rodar e construir seu aplicativo React.
* **Git:** Para controle de versão e interação com o GitHub.
* **Conta no GitHub:** Para hospedar seu repositório.
* **Editor de Código:** Visual Studio Code (VS Code) é altamente recomendado.

**1. Configuração Inicial do Projeto Local**

Se você já tem seu projeto React/Vite configurado, pule para a seção 2.

Para criar um novo projeto Vite React:

1. **Abra seu terminal** (CMD, PowerShell ou Git Bash).
2. **Crie o projeto Vite:**

Bash

npm create vite@latest meu-app-react -- --template react

* + Substitua meu-app-react pelo nome desejado para a pasta do seu projeto (ex: trade-app).

1. **Navegue até a pasta do projeto:**

Bash

cd meu-app-react

1. **Instale as dependências:**

Bash

npm install

# OU, se você usa Yarn:

# yarn

1. **Teste seu aplicativo localmente:**

Bash

npm run dev

# OU

yarn dev

Abra http://localhost:5173 (ou a porta indicada) no seu navegador.

**2. Configuração do Git Local**

Dentro da pasta do seu projeto (C:\trade-app ou o nome que você escolheu):

1. **Inicialize o repositório Git:**

Bash

git init

1. **Configure seu nome de usuário e e-mail no Git:**

Bash

git config user.name "Seu Nome Completo"

git config user.email "seu.email@example.com"

* + \*\*Importante:\*\* Use o mesmo nome e e-mail que você usa na sua conta do GitHub.

1. **Adicione todos os arquivos ao estágio (staging area):**

Bash

git add .

1. **Faça o primeiro commit:**

Bash

git commit -m "Primeiro commit do projeto React"

**3. Preparando o Repositório no GitHub**

1. **Crie um novo repositório no GitHub:**
   * Acesse o GitHub (github.com) e faça login.
   * Clique no botão "**New**" (Novo) no canto superior esquerdo ou no "+" na barra de navegação e selecione "New repository".
   * **Nome do repositório:** Use o mesmo nome da pasta do seu projeto local (ex: trade-management-app).
   * **Descrição:** (Opcional) Adicione uma breve descrição.
   * **Público ou Privado:** Escolha conforme sua preferência.
   * **MUITO IMPORTANTE:** **NÃO MARQUE** as opções "Add a README file", "Add .gitignore" ou "Choose a license". Queremos um repositório completamente vazio.
   * Clique em "**Create repository**" (Criar repositório).
2. **Conecte seu repositório local ao GitHub:** Após criar o repositório vazio, o GitHub mostrará instruções. Copie a linha que adiciona o "remote". Geralmente é assim:

Bash

git remote add origin https://github.com/SeuUsuario/NomeDoRepositorio.git

* + Certifique-se de que NomeDoRepositorio está **exatamente** igual ao que você criou (ex: trade-management-app).

**Solução para "fatal: repository '**[**https://github.com/**](https://github.com/)**...' not found":**

* + Verifique se a URL em git remote -v está correta. Se não, remova com git remote remove origin e adicione com a URL certa.
  + Confirme que o repositório existe e o nome está correto no GitHub.

1. **Envie seu código para o GitHub:**

Bash

git push -u origin main

* + A primeira vez pode pedir autenticação no navegador.

**4. Configuração do Projeto Vite para GitHub Pages (Base Path)**

Esta é uma etapa CRÍTICA para que seu aplicativo carregue corretamente no GitHub Pages.

1. **Abra o arquivo vite.config.js** na raiz do seu projeto local (C:\trade-app\vite.config.js).
2. **Edite o conteúdo** para incluir o base path correto:

JavaScript

import { defineConfig } from 'vite'

import react from '@vitejs/plugin-react'

// https://vitejs.dev/config/

export default defineConfig({

plugins: [react()],

// IMPORTANTE: O 'base' deve ser o nome do seu repositório no GitHub,

// SEMPRE com uma barra inicial e uma barra final.

base: '/trade-management-app/' // Mude 'trade-management-app' para o nome exato do seu repositório

})

* + **\*\*Problema "Página em branco" ou "404 para arquivos CSS/JS":** Isso acontece quando o base path está incorreto ou não foi aplicado. A correção acima resolve isso.

**5. Automatizando o Deploy com GitHub Actions**

O GitHub Actions automatiza o processo de build e deploy, garantindo que seu aplicativo esteja sempre atualizado no GitHub Pages quando você faz um git push.

1. **Crie a estrutura de pastas:** Dentro do seu projeto (C:\trade-app), crie as pastas: C:\trade-app\.github\workflows\
2. **Crie o arquivo de workflow deploy.yml:** Dentro de C:\trade-app\.github\workflows\, crie um arquivo chamado deploy.yml e cole o seguinte conteúdo:

YAML

# Nome do workflow que aparecerá na aba 'Actions' do seu repositório

name: Deploy React App to GitHub Pages

# Quando este workflow será executado

on:

# Executa o workflow em cada push para a branch 'main'

push:

branches:

- main

# Permite que você execute o workflow manualmente na aba 'Actions'

workflow\_dispatch:

# Permissões necessárias para o GitHub Pages e Actions

permissions:

contents: read

pages: write

id-token: write

# Tarefas (jobs) a serem executadas

jobs:

build-and-deploy:

runs-on: ubuntu-latest # O ambiente onde o código será executado

steps:

# 1. Checkout do código: Pega o seu código do repositório

- name: Checkout your repository

uses: actions/checkout@v4

# 2. Configura o Node.js: Instala a versão do Node.js

- name: Setup Node.js

uses: actions/setup-node@v4

with:

node-version: '20' # Use a versão do Node.js que você usa localmente

# 3. Instala as dependências do projeto

- name: Install dependencies

run: npm install # OU yarn install, se você usa Yarn

# \*\*Solução para "sh: 1: vite: Permission denied":\*\*

# Concede permissão de execução ao binário 'vite' (muitas vezes é resolvido por npm install, mas as vezes é necessário este passo)

- name: Add execute permission to Vite binary

run: chmod +x node\_modules/.bin/vite

# 4. Faz o build do seu aplicativo React com Vite

- name: Build React App

run: npx vite build # Usa npx para garantir que o vite seja encontrado

# 5. Configura o GitHub Pages para o deploy

- name: Setup Pages

uses: actions/configure-pages@v4

# 6. Faz o upload dos arquivos de build (da pasta 'dist') como um artefato

- name: Upload artifact

uses: actions/upload-pages-artifact@v3

with:

path: './dist' # Caminho para a pasta de build do Vite

# 7. Faz o deploy dos arquivos para o GitHub Pages

- name: Deploy to GitHub Pages

id: deployment

uses: actions/deploy-pages@v4

1. **Salve o arquivo deploy.yml.**
2. **Adicione, commite e envie o workflow para o GitHub:**

Bash

git add .

git commit -m "Adiciona workflow para deploy no GitHub Pages"

git push origin main

Isso iniciará a primeira execução do seu workflow.

**6. Ativando GitHub Pages via GitHub Actions**

Após enviar o arquivo deploy.yml:

1. **Vá para o seu repositório no GitHub.**
2. Clique na aba **Settings** (Configurações).
3. No menu lateral esquerdo, clique em **Pages**.
4. Em "Build and deployment", para "Source", selecione **GitHub Actions**.
5. Clique em **Save** (se houver um botão).

**7. Verificando o Deploy e Acessando seu Aplicativo**

1. **Monitore o Workflow:**
   * Vá para a aba **Actions** no seu repositório.
   * Clique no workflow "Deploy React App to GitHub Pages".
   * Aguarde até que todos os passos fiquem verdes (indicando sucesso). Isso pode levar alguns minutos.
2. **Acesse seu Aplicativo:** Uma vez que o workflow esteja completo, seu aplicativo estará disponível em: https://SeuUsuario.github.io/NomeDoRepositorio/ No seu caso, é: https://RafaelDavidTeixeira.github.io/trade-management-app/

**8. O que fazer se mudar o local ou nome da pasta/repositório?**

**A. Mudar a Pasta de Local no seu Computador (sem Renomear)**

Se você apenas moveu a pasta do projeto (ex: de C:\trade-app para D:\MeusProjetos\trade-app), a única coisa que você precisa fazer é:

* **Navegar até a nova localização** no seu terminal (CMD, PowerShell, Git Bash) antes de executar qualquer comando Git (git push, git pull) ou comandos do npm (npm run dev, npm run build, npm install).

Bash

cd D:\MeusProjetos\trade-app

* **Nada mais muda.** O Git ainda sabe qual é o repositório remoto, e a configuração do GitHub Pages permanece a mesma.

**B. Renomear a Pasta do Projeto Localmente (sem Renomear o Repositório no GitHub)**

Se você apenas renomeou a pasta do projeto no seu computador (ex: de trade-app para meu-novo-trade-app), as ações são as mesmas do item A:

* **Navegue até a pasta com o novo nome** no seu terminal.
* **Nada mais muda.** O Git e o GitHub Pages não são afetados, pois o nome do repositório remoto permanece o mesmo.

**C. Renomear o Repositório no GitHub**

Este é o cenário mais importante, pois afeta a URL do seu aplicativo e a configuração do seu projeto.

1. **Renomeie o repositório no GitHub:**
   * Vá para o seu repositório no GitHub (ex: https://github.com/RafaelDavidTeixeira/trade-management-app).
   * Clique na aba **Settings** (Configurações).
   * No menu lateral esquerdo, clique em **General** (Geral).
   * Role para baixo até a seção "Repository name" e clique em **Rename** (Renomear).
   * Digite o **Novo Nome** do repositório (ex: trade-manager-app-v2) e clique em "Rename".
   * \*\*\*\*Consequência:\*\* A URL do seu aplicativo no GitHub Pages mudará automaticamente. Se era[https://SeuUsuario.github.io/trade-management-app/](https://www.google.com/search?q=https://SeuUsuario.github.io/trade-management-app/), agora será[https://SeuUsuario.github.io/trade-manager-app-v2/](https://www.google.com/search?q=https://SeuUsuario.github.io/trade-manager-app-v2/).
2. **Atualize seu Projeto Local:**
   * **Atualize a URL do remote no Git local:** Abra seu terminal na pasta do seu projeto local e execute:

Bash

git remote set-url origin https://github.com/SeuUsuario/NovoNomeDoRepositorio.git

(Ex: git remote set-url origin https://github.com/RafaelDavidTeixeira/trade-manager-app-v2.git) Você pode verificar com git remote -v.

* + **Atualize o base path no vite.config.js:** Abra vite.config.js e altere a linha base para o novo nome do repositório:

JavaScript

// vite.config.js

export default defineConfig({

// ...

base: '/NovoNomeDoRepositorio/' // EX: '/trade-manager-app-v2/'

})

* + **Refaça o Build Localmente:**

Bash

npm run build

# ou

yarn build

* + **Adicione, Commite e Envie as Alterações para o GitHub:**

Bash

git add .

git commit -m "Atualiza configuracao para novo nome do repositorio"

git push origin main

1. **Verifique o GitHub Actions e o GitHub Pages:**
   * Após o git push, o workflow do GitHub Actions será acionado. Verifique se ele é concluído com sucesso.
   * Acesse as configurações de **GitHub Pages** (Settings > Pages). Verifique se a "Source" ainda está definida como "GitHub Actions". O URL do seu aplicativo já deve estar atualizado para o novo nome do repositório.