**Despliegue en GitHub Pages, de proyecto en Vite v3 + React 18**

3 votos3

[Juan Carlos](https://www.udemy.com/user/juan-carlos-caycho-paucar/) ·

Clase 215

 · hace 19 días

El proyecto fue desarrollado con Vite, el cual se generó con la versión 3 al momento de inicial izarlo.

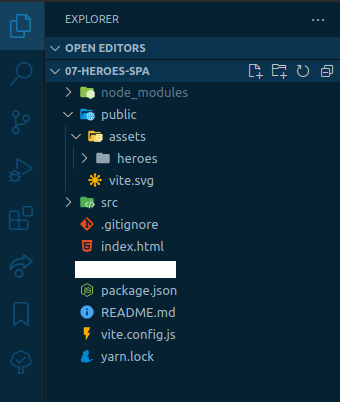
1. "vite": "^3.0.7"

Por lo que se creó por defecto la carpeta public.

Antes de poder realizar la subida de archivos a GitHub, debemos de realizar algunos cambios en la estructura del proyecto. Esto es, para que no tengamos problemas al momento de mostrar las imágenes.

**PASOS PREVIOS**

1. Mover la carpeta assets, hacia adentro de la carpeta public.



2. Abrimos el archivo src/heroes/components/HeroCard.jsx.

3. Modificamos heroImageUrl y guardamos.

1. const heroImageUrl = `/assets/heroes/${id}.jpg`;
3. por
5. const heroImageUrl = `./assets/heroes/${id}.jpg`;

4. Abrimos el archivo src/heroes/pages/HeroPage.jsx.

5. Nos ubicamos sobre el if() y agregamos la siguiente linea de codigo:

1. const heroImageUrl = `./assets/heroes/${id}.jpg`; // usando github pages
3. if (!hero) {
4. return <Navigate to="/marvel" />;
5. }

6. Luego, en nuestra etiqueta <img> vamos a modificar el contenido del src.

7. Debe de quedar de la siguiente manera:

1. <div className="col-4">
2. <img
3. src={heroImageUrl}
4. alt={hero.superhero}
5. className="img-thumbnail animate\_\_animated animate\_\_fadeInLeft"
6. />
7. </div>

8. Guardamos el archivo.

**Habiendo realizado estas modificaciones, ya podemos preparar el proyecto para ser subido a GitHub Pages.**

**REACT  18 + VITE + GITHUB PAGES DEPLOYMENT**

- Cuando alojamos nuestra pagina en Github Pages y utilizamos react router dom 6 + react 18.

**PASOS PARA UN NUEVO REPOSITORIO (si aún no lo hemos hecho)**

1 .Inicializamos un nuevo repositorio en la linea de comandos:

1. git init
2. git add .
3. git commit -m "first commit"

2. Creamos un repositorio público en GitHub.

3. Para el ejemplo, lo nombramos como: 07-heroes-spa-2

4. Realizamos los pasos que nos indica GitHub: (donde xxxxxx es nuestro usuario), por ejemplo:

1. git remote add origin https://github.com/xxxxxx/07-heroes-spa-2.git
2. git branch -M main
3. git push -u origin main

**MODIFICACIONES LUEGO DE CREAR NUESTRO REPOSITORIO**

1. En nuestro componente que tenga el BrowserRouter, cambiarlo por HashRouter.

2. En este caso, el componente es el main.jsx.

3. Modificamos (**no olvidar realizar la importacion** de react-router-dom) y guardamos:

1. ReactDOM.createRoot(document.getElementById('root')).render(
2. <React.StrictMode>
3. <BrowserRouter>
4. <HeroesApp />
5. </BrowserRouter>
6. </React.StrictMode>
7. );

10. por

13. ReactDOM.createRoot(document.getElementById('root')).render(
14. <React.StrictMode>
15. <HashRouter>
16. <HeroesApp />
17. </HashRouter>
18. </React.StrictMode>
19. );

4. En el archivo package.json, agregar la propiedad homepage, puede ser al inicio:

1. "homepage": "https://myuser.github.io/xxxxxxx/",
3. donde 'xxxxxxx' será el nombre de nuestro repositorio que creamos antes en GitHub
4. donde 'myuser' es nuestro usuario de github

**PASOS PARA REALIZAR EL BUILD**

1. Habiendo realizado los pasos anteriores, ya estamos listos para realizar el build.

2. En la consola, colocamos:

1. yarn build

3. Se crea una carpeta llamada dist, la cual renombraremos por docs.

4. Dentro de la carpeta docs, abrimos el archivo index.html.

5. Dentro de la etiqueta <head>, ubicamos las etiquetas <script> y <link> correspondientes a los archivos de js y css.

6. Les agregamos un . al inicio de la ruta, por ejemplo:

1. <script type="module" crossorigin src="./assets/index.e61fb4c0.js"></script>
2. <link rel="stylesheet" href="./assets/index.4d801bf2.css">

7. Subimos los cambios a nuestro repositorio:

1. git add .
2. git commit -m "build project"
3. git push

**PASOS PARA CREAR LA PAGINA EN GITHUB PAGES**

- Pasos para crear la pagina en github

1. Abrimos nuestro repositorio.

2. Damos clic en Settings.

3. Damos clic en Pages.

4. En Branch, seleccionamos main.

5. Donde dice /root lo cambiamos por /docs.

6. Clic en Save.

7. Esperamos 1 o 2 minutos hasta que se despliegue.

8. Recargamos la pagina para que nos muestre la url generada.

9. Abrimos nuestra url.

**IMPORTANTE:**

Una vez que hemos probado que nuestra pagina funciona correctamente en GitHub Pages, debemos de realizar las siguientes modificaciones, **para que nuestro proyecto funcione en local (modo desarrollo).**

En **PASOS PREVIOS,** en el **punto 5**, debemos de modificar la ruta. Debe de quedar así:

5. Nos ubicamos sobre el if() y agregamos la siguiente linea de código:

1. const heroImageUrl = `./../assets/heroes/${id}.jpg`; // en modo desarrollo
3. if (!hero) {
4. return <Navigate to="/marvel" />;
5. }

También, modificar el archivo main.jsx, debe de quedar así:

1. ReactDOM.createRoot(document.getElementById('root')).render(
2. <React.StrictMode>
3. <BrowserRouter>
4. <HeroesApp />
5. </BrowserRouter>
6. </React.StrictMode>
7. );

Luego, en el archivo el package.json, borrar la linea de **homepage**.

Por ultimo, probar con **yarn dev** que todo esté funcionando correctamente en modo desarrollo.

**NOTA:** Este es el enlace a mi pagina desplegada [udemy-heroes-app-2](https://jccaychop.github.io/07-heroes-spa-2/#/)

Espero les sirva y perdón por extenderme, ya que traté de ser lo mas detallado posible.

<https://www.udemy.com/course/react-cero-experto/learn/lecture/32129514#questions/18155686>

<https://www.udemy.com/course/react-cero-experto/learn/lecture/19968822#questions/18152638>