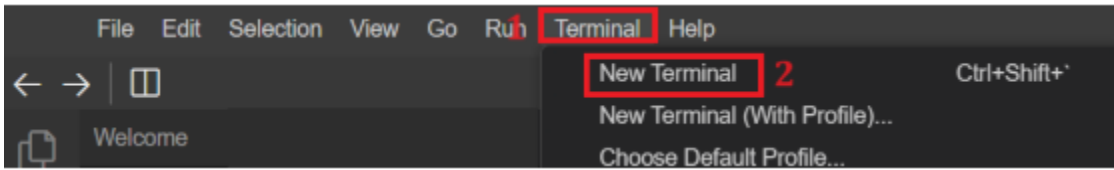


# Pasos para clonar mi repositorio e inicializar mi proyecto de REACT

- Comandos de git en la terminal



Asegúrate de que la ruta de tu terminal sea `/home/project/ cd <learning_react>`, donde debes reemplazar `<learning_react>` con el nombre de tu propio proyecto.

A continuación, inicializa la terminal para crear un repositorio Git de modo que puedas ejecutar todos los comandos Git en esta terminal. Usa el comando proporcionado y presiona Enter.

**git init**

**Presiona enter**

Luego debes configurar la terminal para ejecutar los comandos necesarios dentro del entorno de red de habilidades.

**git config --global --add safe.directory /home/Project**

**Presiona enter**

A continuación, debes configurar Git con tu dirección de correo electrónico. Reemplaza `you@example.com` por tu dirección de correo electrónico. Asegúrate de que tu dirección de correo electrónico esté entre comillas dobles.

**git config --global user.email "you@example.com"**

**Presiona enter**

A continuación, ejecute `git add` y `git commit` para guardar los cambios en el repositorio de GitHub. Ejecute los siguientes comandos uno tras otro:

**git add -a**

**git commit -m "initial commit"**

Ejecute comandos `git push` para enviar los archivos a su repositorio de GitHub. Debe configurar su rama principal en su repositorio de GitHub.

**git branch -M main**

Agrega la URL de tu repositorio de GitHub en una variable de origen. También reemplaza `<git-repo-url>` con la URL de tu repositorio de GitHub.

Por ejemplo:

**git remote add origin <https://github.com/<youraccountname>/<yourrepositoryname>>**

**git remote add origin <git-repo-url>**

**Presiona enter**

Luego, envía el contenido de tu archivo a tu repositorio de GitHub:

**git push -u origin main**

**Presiona enter**



Al enviar los archivos a GitHub mediante el comando `git push`, se le pedirá que ingrese su nombre de usuario y contraseña para su cuenta de GitHub en la terminal.

Ingresa tu nombre de usuario y luego presiona Enter.

Para tu contraseña, debes pegar el token de acceso personal generado en el paso 1.

Al pegar su token de acceso personal en la terminal, no se mostrará por razones de seguridad, pero estará allí. Simplemente presione Enter y se enviarán sus archivos y carpetas al repositorio de GitHub.

Todos sus archivos se han enviado a su directorio de GitHub.

**Si previamente iniciaste sesión no te pedirá tu nombre y el token, de no ser así se tendrá que generar el token.**

## Deploy usando GitHub Pages

Para implementar tu aplicación React en GitHub, debes instalar `gh-pages`. Esto te permite usarlo como una herramienta para implementar tu proyecto en GitHub Pages. Ejecuta el comando dado en la terminal.

**`npm install gh-pages --save-dev`**

Agregue las líneas dadas antes de "build": "vite build" en el archivo `package.json`.

**"predeploy": "npm run build",**

**"deploy": "gh-pages -d dist",**

```
"scripts": {
  "dev": "vite",
  "predeploy": "npm run build",
  "deploy": "gh-pages -d dist",
  "build": "vite build",
  "lint": "eslint . --ext js,jsx --report-unused-disable-directives --max-warr
```

Luego, en el archivo `vite.config.js`, agregue esta línea antes de los complementos:

`[react()]`

**base: "/YOUR\_REPOSITORY\_NAME",**

```
export default defineConfig({
  base: "/YOUR_REPOSITORY_NAME",
  plugins: [react()],
})
```

Nota: En lugar de `<YOUR_REPOSITORY_NAME>`, escribe el nombre de tu propio repositorio, por ejemplo, si el nombre de tu repositorio de GitHub es `learning_react`, debería verse así:

base: `"/learning_react"`



Ahora ejecuta el comando de implementación en la terminal para ejecutar el script "deploy" definido en el archivo package.json, implementando el proyecto en GitHub Pages usando la herramienta gh-pages.

### npm run deploy

```
theia@theia-ritikaj:/home/project/react$ npm run deploy

> react@0.0.0 predeploy
> npm run build

> react@0.0.0 build
> vite build

vite v5.3.1 building for production...
✓ 34 modules transformed.
dist/index.html      0.49 kB | gzip: 0.30 kB
dist/assets/react-CHdo91hT.svg  4.13 kB | gzip: 2.14 kB
dist/assets/index-DIwrgTda.css  1.39 kB | gzip: 0.72 kB
dist/assets/index-DQ9y_QkT.js 143.38 kB | gzip: 46.06 kB
✓ built in 1.20s

> react@0.0.0 deploy
> gh-pages -d dist

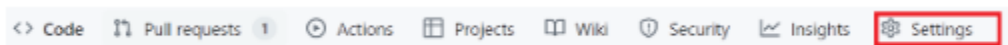
Username for 'https://github.com':
Password for 'https://github.com':
Published
theia@theia-ritikaj:/home/project/react$
```

**Nota:** Siempre que realices algún cambio en tu código, debes guardar todos tus archivos y ejecutar comandos git para ellos.

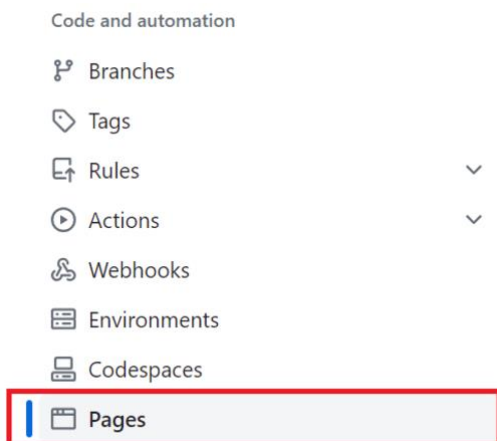
Ejecuta los comandos git add y git commit para actualizar los cambios en tu código. Luego, ejecuta el comando git push para actualizar tu repositorio de GitHub para una administración adecuada del código.

Ve a tu repositorio de GitHub. Luego, navega hasta el repositorio de tu sitio que creaste.

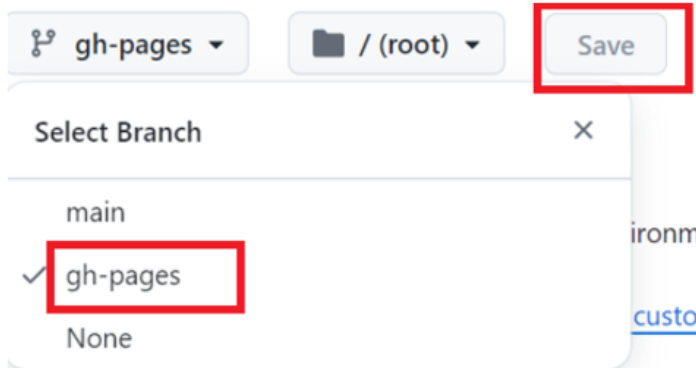
Debajo del nombre de tu repositorio, haz clic en Configuración.



Vaya a la barra de navegación del lado izquierdo. En la sección Código y automatización de la barra lateral, haga clic en Páginas.



Verá la página que se muestra a continuación. Haga clic en el menú desplegable donde aparece Ninguno, luego haga clic en gh-pages y luego haga clic en el botón Guardar.



Actualice su página nuevamente y verá el enlace, tal como se muestra a continuación. En lugar de shoppingreact, verá el nombre de su repositorio de Github.

Haga clic en el enlace generado arriba para ver su sitio web en vivo.



# Vite + React

count is 0

Edit `src/App.jsx` and save to test HMR

1. Si no has realizado ningún cambio en tu aplicación React predeterminada, deberías ver un resultado similar para la tarea 3, paso 6, como se muestra a continuación.
2. Después de la implementación en GitHub Pages, puede que pase un tiempo hasta que todos los contenidos e imágenes aparezcan correctamente. Espera unos minutos más para que la aplicación se cargue por completo.