

El enfoque de Vercel es diferente: en lugar de "empujar" (push) tu código desde la terminal con un comando CLI (como firebase deploy), Vercel "tira" (pull) de tu código directamente desde un repositorio de Git (como GitHub).

Esto habilita un flujo de trabajo increíblemente poderoso llamado **CI/CD** (Integración Continua / Despliegue Continuo). En resumen: **cada vez que haces git push a tu repositorio, Vercel lo despliega automáticamente.**

---

### Tutorial: Despliega tu App de React en Vercel (con Git)

**Objetivo:** Publicar tu aplicación de React (con React Router) en una URL pública usando Vercel y un repositorio de GitHub.

#### Prerrequisitos:

- Una cuenta de Vercel (puedes registrarte gratis con tu cuenta de GitHub).
  - Una cuenta de GitHub.
  - Tu proyecto de React listo (el que hemos construido).
  - Tener git instalado en tu computadora.
- 

#### Paso 1: Poner tu Proyecto en GitHub

Vercel necesita un lugar de donde leer tu código. GitHub es la opción más común.

**Si tu proyecto ya está en GitHub, puedes saltar este paso.**

Si no lo está:

1. Ve a [GitHub.com](https://github.com) y crea un **nuevo repositorio** (ej: mi-app-react-demo). Puede ser público o privado, Vercel funciona con ambos.
2. Una vez creado, GitHub te dará la URL del repositorio (ej: <https://github.com/tu-usuario/mi-app-react-demo.git>).
3. Abre tu terminal en la **raíz de tu proyecto de React** (mi-app-modular).
4. Ejecuta los siguientes comandos para conectar y subir tu código (asegúrate de que tu proyecto ya tenga un .gitignore, que Create React App incluye por defecto):

Bash

```
# Inicializa git si no lo has hecho
```

```
git init
```

```
# Añade todos tus archivos para el primer commit
```

```
git add .

# Crea el primer commit
git commit -m "Versión inicial para Vercel"

# Conecta tu repositorio local con el de GitHub
git remote add origin https://github.com/tu-usuario/mi-app-react-demo.git

# Sube tu código a la rama 'main' (o 'master')
git push -u origin main

¡Listo! Tu código ahora vive en GitHub.
```

---

**Paso 2: Conectar Vercel y Desplegar**

Este es el proceso central de Vercel.

1. Ve a [vercel.com](#) e inicia sesión. Lo más fácil es hacer clic en "**Continue with GitHub**".
2. Una vez en tu "Dashboard", haz clic en "**Add New...**" y selecciona "**Project**".
3. Vercel te mostrará la sección "**Import Git Repository**". Como iniciaste sesión con GitHub, deberías ver una lista de tus repositorios.
4. Busca el repositorio que acabas de subir (ej: mi-app-react-demo) y haz clic en el botón "**Import**" que está a su lado.
5. Vercel te pedirá permiso para instalar su app en tu cuenta de GitHub. Apruébalo (puedes darle acceso a todos tus repositorios o solo a ese).

---

### **Paso 3: Configurar el Proyecto (La "Magia" de Vercel)** ✨

Aquí es donde Vercel brilla. Te llevará a una pantalla de "Configure Project".

1. **Framework Preset:** Vercel escaneará tu código y **automáticamente detectará que es una "Create React App"**.
2. **Build and Output Settings:** Gracias a que detectó el framework, Vercel **configura todo por ti**:
  - **Build Command:** npm run build
  - **Output Directory:** build

- **Install Command:** npm install

¡No necesitas tocar nada! Vercel es lo suficientemente inteligente como para saber cómo construir una app de Create React App y, lo que es más importante, **automáticamente configura los "rewrites" para que React Router funcione** (el mismo paso que hicimos manualmente en Firebase).

3. Haz clic en el botón "Deploy".
- 

#### Paso 4: ¡Tu App está en Vivo!

Vercel te mostrará un registro (log) de lo que está haciendo: clonando tu repositorio, instalando las dependencias (npm install), construyendo el proyecto (npm run build) y desplegándolo.

En uno o dos minutos, verás una pantalla de felicitaciones con una captura de pantalla de tu app.

¡Haz clic en el dominio (ej: mi-app-react-demo.vercel.app) y listo! Tu aplicación está en línea, con HTTPS, en una CDN global.

---

#### Paso 5: El Flujo Mágico (CI/CD)

Ahora, probemos el verdadero poder de Vercel.

1. En tu computadora, abre tu proyecto en VS Code.
2. Ve a src/components/Home/Home.js y haz un pequeño cambio de texto. Por ejemplo:
  - <h2>Bienvenido a la Aplicación de Demostración en Vercel!</h2>
3. Guarda el archivo.
4. Abre tu terminal y simplemente haz un git push:

Bash

```
git add .
```

```
git commit -m "Actualizando el título de Home"
```

```
git push
```

5. Vuelve a tu dashboard de Vercel. Verás que un **nuevo despliegue ha comenzado automáticamente**. Vercel detectó tu push y está reconstruyendo la app.
6. Cuando termine (en menos de un minuto), recarga tu URL de Vercel. Verás tu cambio en vivo.

#### Resumen: Vercel vs. Firebase Hosting

Característica	Firebase Hosting	Vercel
<b>Flujo de Trabajo</b>	Basado en CLI (local)	Basado en Git (en la nube)
<b>Comando de Despliegue</b>	npm run build luego firebase deploy	git push
<b>Configuración SPA (React Router)</b>	Manual (en firebase init)	<b>Automática</b>
<b>Configuración de Build</b>	Manual (en firebase init)	<b>Automática</b>
<b>CI/CD</b>	Manual o con GitHub Actions	<b>Incorporado y gratuito</b>

Ambos son excelentes, pero Vercel está diseñado desde cero para este flujo de trabajo moderno de git push, lo que lo hace increíblemente rápido y conveniente para los desarrolladores de frontend.