

# React Build: Construyendo y Desplegando Aplicaciones

---

Diego Muñoz

09 de noviembre de 2025

## Del entorno local a producción

- Todo proyecto React requiere un proceso de **compilación (build)** antes de desplegar.
- El objetivo del build es **optimizar**: minificar, agrupar y preparar para servidores estáticos.
- Herramientas comunes: **Vite**, **esbuild**, **Rollup**, **Webpack**.

### Comando base

`yarn build`

- Genera la carpeta `dist/` con archivos listos para producción.
- Usa `vite.config.js` y variables de entorno que comienzan con `VITE_`.
- Revisa tamaños finales de JS/CSS y elimina dependencias innecesarias.

Carpeta `dist/`

`dist/`

```
├─ index.html
├─ assets/
│   ├─ main-8bd73c.js
│   └─ style-11f2.css
```

- `index.html` es el punto de entrada.
- Los archivos en `assets/` están **hashados** para caché eficiente.
- Se debe servir **todo el contenido estáticamente** (sin backend).

### Comando de previsualización

`yarn preview`

- Inicia un servidor local (por defecto en `http://localhost:4173`).
- Permite revisar rutas, assets y variables antes del despliegue.
- Ideal para simular el entorno productivo.

### Variables en tiempo de build

- Solo las variables que comienzan con `VITE_` se inyectan al bundle.
- Ejemplo:

`VITE_API_BASE_URL=https://api.example.com`

- No incluir secretos ni claves privadas.
- Los valores sensibles deben manejarse desde el backend o desde la plataforma (Netlify env vars).

### Introducción

- Netlify permite publicar sitios estáticos **directamente desde GitHub**.
- Cada push al repositorio puede **reconstruir y desplegar automáticamente**.
- Ideal para proyectos React con Vite, sin necesidad de servidores manuales.  
*La ventaja de Netlify es su integración con CI/CD y facilidad de uso, y es gratis para proyectos pequeños y medianos.*

# Paso 1: Conectar el repositorio

Desde el panel de Netlify

1. Inicia sesión en [netlify.com](https://netlify.com).
2. Selecciona “Add new site → Import an existing project”.
3. Conecta tu cuenta de **GitHub** (o GitLab/Bitbucket).
4. Elige el repositorio del proyecto React.

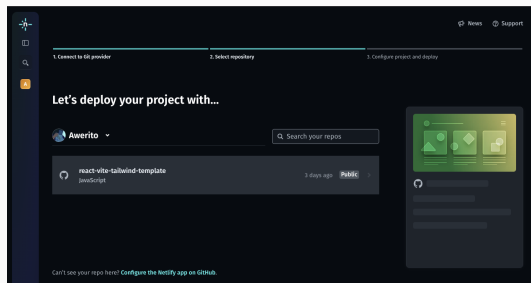


Figure 1: Conexión con GitHub



## Paso 2: Configurar el build

### Ajustes iniciales

- Build command: `yarn build`
- Publish directory: `dist`
- Netlify detecta automáticamente que es un proyecto Vite/React.
- Puedes añadir variables en **Site settings** → **Environment variables**.

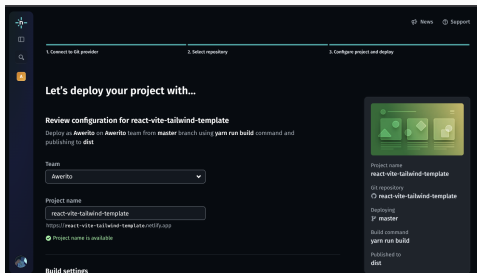


Figure 2: Configuración de build

### Primer despliegue

- Al hacer clic en **Deploy site**, Netlify ejecutará el build y publicará el resultado.
- URL temporal generada automáticamente:  
`https://<nombre-configurado>.netlify.app`
- Puedes cambiar el nombre del dominio en **Site settings** → **Domain management**.
- El nombre configurado debe ser único en Netlify como si fuera un nombre de usuario.

## Paso 4: Actualizaciones automáticas

### CI/CD integrado

- Cada vez que haces **push a main** o **merge a PR**, Netlify:
  1. Ejecuta **yarn build**
  2. Publica los nuevos archivos en el dominio
- Puedes revertir a versiones anteriores desde el panel (“Deploys → Rollback”).

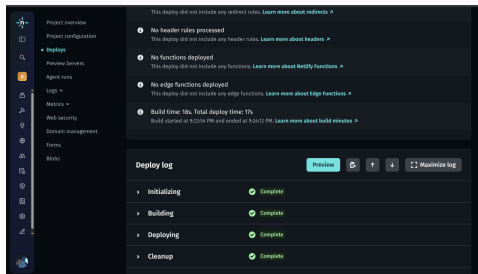


Figure 3: Deploys en Netlify

Etapa	Herramienta	Resultado
Desarrollo	<code>yarn dev</code>	Hot Reload, entorno local
Build	<code>yarn build</code>	Archivos minificados en <code>dist/</code>
Preview	<code>yarn preview</code>	Simula entorno productivo
Deploy	Servicio elegido	Sitio público online

1. Mantén `vite.config.js` actualizado con el `base` correcto (si el sitio no está en `/`).
2. Usa `.env` local para desarrollo y `env vars` para producción.
3. Valida que las rutas SPA (React Router) funcionen con `redirects`.

1. El build convierte tu código React en un sitio optimizado.
2. Servicios como Netlify automatizan el deploy y CI/CD con un click.
3. Revisión previa con `yarn preview` evita errores comunes.
4. Las variables `VITE_` son la única forma de pasar configuración al cliente.

- [Guía oficial de Deploy en Netlify](#)
- [Netlify Docs](#)
- [Vite Build Command](#)
- [Environment Variables en Netlify](#)