

BLOG

# Git para el Desarrollo Web: Conoce el Flujo de Trabajo Típico de un Proyecto

Daniel Díaz, septiembre 19, 2022



Compartir en

Despliega tu aplicación rápidamente y escálala según crezcas con nuestro Nivel Hobby



El desarrollo web está intrínsecamente relacionado con la colaboración. La mayoría de las veces trabajarás con otros desarrolladores, e incluso si no lo haces, Git puede ayudarte de muchas otras maneras.



abierto del mundo.



**¿Tienes prisa? Guarda este artículo como PDF.**

Descargar

Como [desarrollador web](#), es extremadamente importante saber cómo utilizar Git para el desarrollo web correctamente. No estamos hablando sólo de «git add», «git commit» y «git push». Debes conocer todo el flujo de trabajo para crear un proyecto web con Git.

¿Aún no estás convencido? ¡Empecemos!

## Tabla de contenidos

- [¿Por qué usar Git?](#)
- [Flujo de trabajo básico de Git para el desarrollo web](#)
- [Cómo usar Git en Kinsta](#)



Estas son sólo algunas de las razones para empezar a utilizar Git:

- **Organización:** En lugar de gestionar tu proyecto en carpetas como v1, v2, v3, etc, tienes un solo proyecto con una base de datos especial que almacena todas las versiones de los archivos
- **Colaboración:** Git te permite a ti y a otras personas trabajar en el mismo proyecto al mismo tiempo sin crear conflictos.
- **Código abierto:** Git es de código abierto, pero también es la herramienta que utilizamos para colaborar y crear un gran software de código abierto. Cualquiera puede hacer pull requests a proyectos de código abierto en plataformas como [GitHub](#) o [Bitbucket](#).
- **Flexibilidad de la plataforma:** Hoy en día, tienes muchos servicios de alojamiento de Git entre los que elegir, como [Gitlab](#), GitHub, Bitbucket y SourceForge. Incluso puedes utilizar una solución autoalojada para todos tus proyectos.
- **Copias de seguridad fáciles:** Deshaz los errores con facilidad, y nunca pierdas la base de código de tu proyecto.

“ Es hora de aprender algo más que 'git add', 'git commit' y 'git push' 🤖  
Sumérgete en todo lo que necesitas saber sobre un flujo de trabajo típico de Git en esta guía 📌

CLIC PARA TUITEAR

Ya hemos mencionado el término GitHub una o dos veces, así que ¿cuál es la diferencia entre Git y GitHub?

### ¿Tu sitio WordPress es lento?

Descubre los cuellos de botella en el rendimiento para ofrecer una mejor experiencia de usuario y alcanzar los objetivos de ingresos de tu empresa.

Auditoría Gratuita

Git es el sistema de control de versiones (VCS) que utilizamos para mantener el control de los cambios de nuestros archivos, mientras que GitHub es un servicio que utilizamos para almacenar los archivos de nuestro proyecto y su historial Git en línea (ubicado en la carpeta **.git/** de tu proyecto).

Git se instala localmente, en tu máquina, y sin servicios de alojamiento como GitHub o GitLab, sería muy difícil colaborar con otros [desarrolladores](#).

GitHub potencia a Git añadiendo otras funciones que mejoran la colaboración, como la clonación, la bifurcación y la fusión. Juntas, estas dos herramientas se combinan para ofrecerte un ecosistema relativamente amigable para desarrollar, gestionar y mostrar tu proyecto a otras personas.

## Flujo de trabajo básico de Git para el desarrollo web

En las siguientes secciones, aprenderás más sobre el flujo de trabajo de Git para el desarrollo web a través de la práctica.

### Requisitos de instalación

Si aún no has instalado Git, este es el momento perfecto. Es fácil de instalar y está disponible en la mayoría de los sistemas operativos.

Descárgalo desde la [página oficial de descargas](#), o instálalo con un gestor de paquetes si usas Linux o macOS:

**About****Documentation****Downloads**

GUI Clients  
Logos

**Community**

The entire **Pro Git book** written by Scott Chacon and Ben Straub is available to [read online for free](#). Dead tree versions are available on [Amazon.com](#).

## Downloads



Older releases are available and the [Git source repository](#) is on GitHub.

**GUI Clients**

Git comes with built-in GUI tools (**git-gui**, **gitk**), but there are several third-party tools for users looking for a platform-specific experience.

[View GUI Clients →](#)

**Logos**

Various Git logos in PNG (bitmap) and EPS (vector) formats are available for use in online and print projects.

[View Logos →](#)

**Git via Git**

If you already have Git installed, you can get the latest development version via Git itself:

— Página de descargas de Git.

Para comprobar que todo ha ido bien con la instalación, enciende un terminal en Linux o macOS buscando «Terminal» en tu menú de aplicaciones, o abriendo Git bash en Windows (que viene instalado con Git por defecto).

A continuación, escribe:

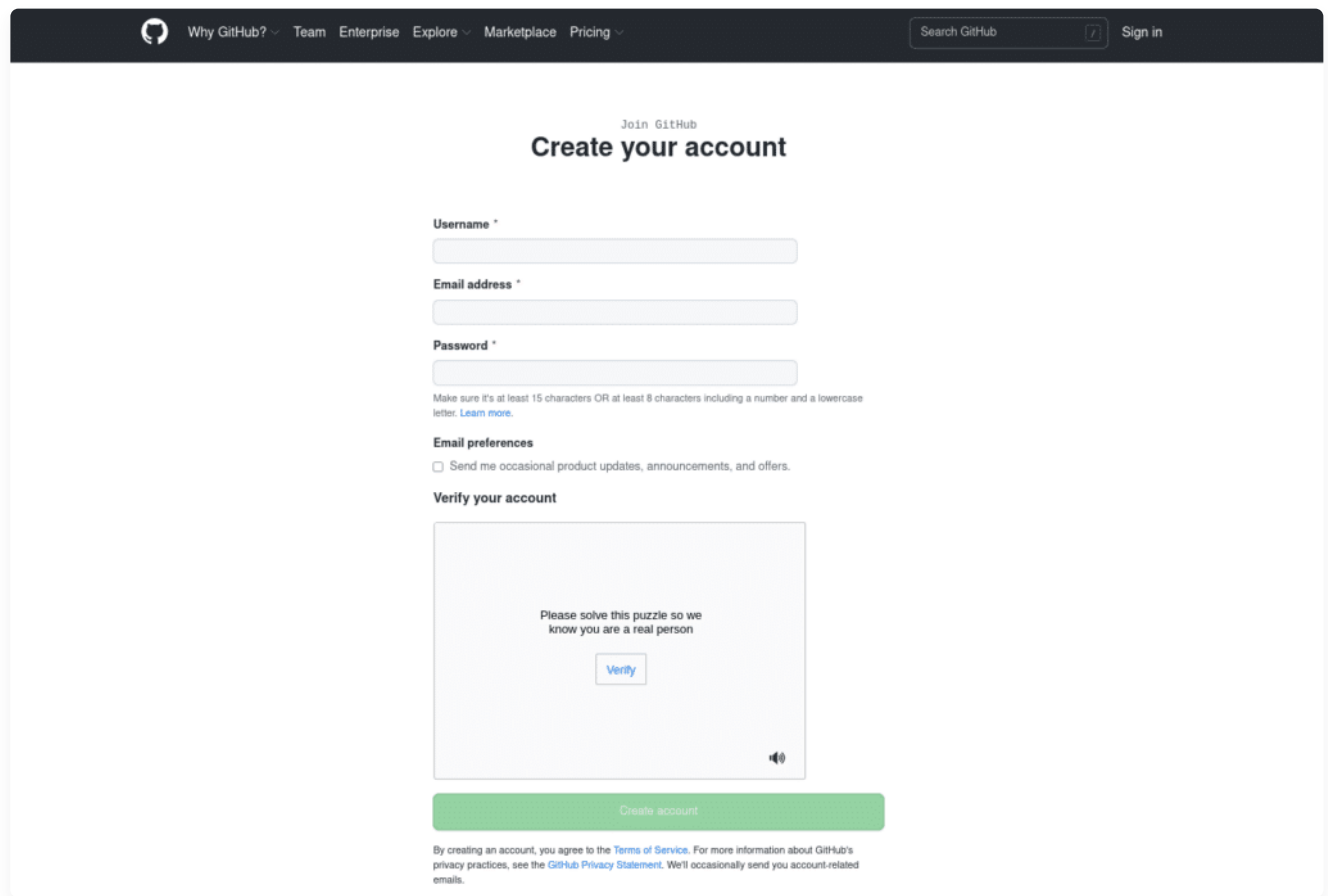
```
git --version
```

```
~  
> git --version  
git version 2.33.0  
~  
> █
```

— Versión Git.

Si obtienes una versión Git como respuesta, ya estás listo.

También necesitaremos una cuenta de GitHub, así que asegúrate de [registrarte](#) o iniciar sesión en GitHub:



Join GitHub

## Create your account

Username \*

Email address \*

Password \*

Make sure it's at least 15 characters OR at least 8 characters including a number and a lowercase letter. [Learn more.](#)

Email preferences

☐ Send me occasional product updates, announcements, and offers.

Verify your account

Please solve this puzzle so we know you are a real person

Verify

Create account

By creating an account, you agree to the [Terms of Service](#). For more information about GitHub's privacy practices, see the [GitHub Privacy Statement](#). We'll occasionally send you account-related emails.

— Página de registro en GitHub

Una vez que tengas instalado Git y hayas iniciado sesión en tu cuenta de GitHub, puedes pasar a la siguiente sección.

## Flujo de trabajo básico de Git para proyectos colaborativos

Como se hemos mencionado anteriormente, la mayor parte del tiempo no desarrollarás proyectos en solitario. Colaborar es una habilidad clave, y Git y GitHub nos ayudan a convertirlo en un proceso sencillo pero eficaz.

El flujo de trabajo típico de un proyecto Git es el siguiente

2. Crea una rama con un nombre representativo de la función en la que vas a trabajar.
3. Edita el proyecto.
4. Confirma los cambios en tu máquina local.
5. Empuja los cambios al repositorio remoto.
6. Crea un pull request al repositorio original.
7. Fusiona y resuelve los conflictos en la rama principal del repositorio original.

## Tutorial

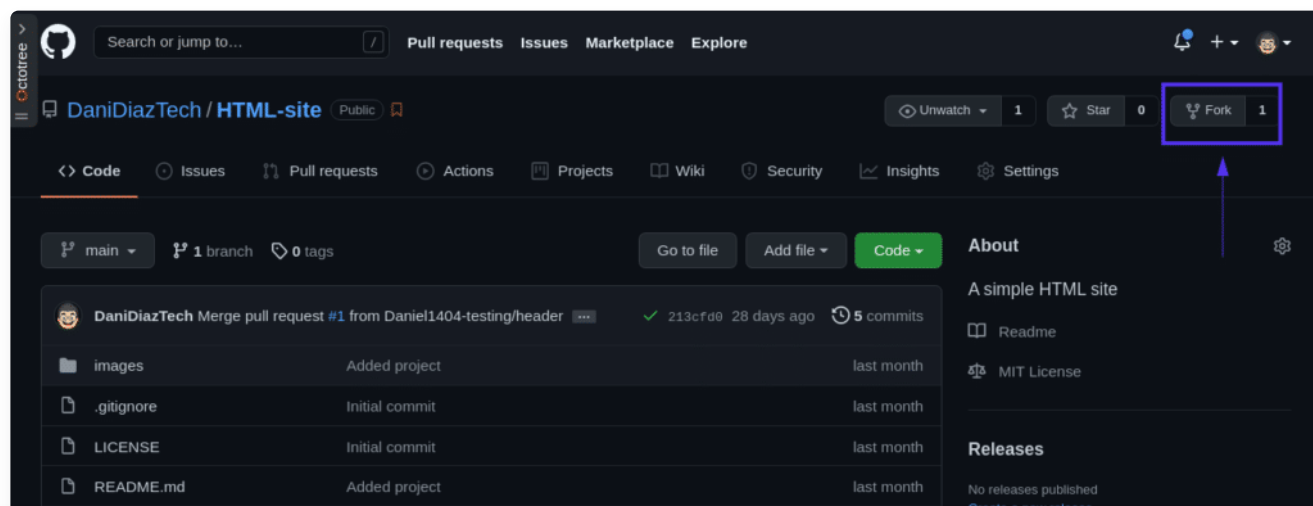
¡Ahora es el momento de ensuciarse las manos!

En esta guía, vas a crear un sitio web HTML sencillo. A efectos prácticos, bifurcarás el proyecto base del [repositorio del sitio HTML](#) a tu cuenta de GitHub. Esto se puede hacer para todos los repositorios públicos disponibles.

### Info

Un fork es una copia separada de un repositorio que puedes gestionar y modificar sin afectar al proyecto original. Clonar un repositorio, por otro lado, simplemente crea una copia local de los archivos

Para hacer un fork del sitio HTML, ve a [este repositorio de GitHub](#) y haz clic en el botón **Fork** situado en la parte superior derecha de la página:



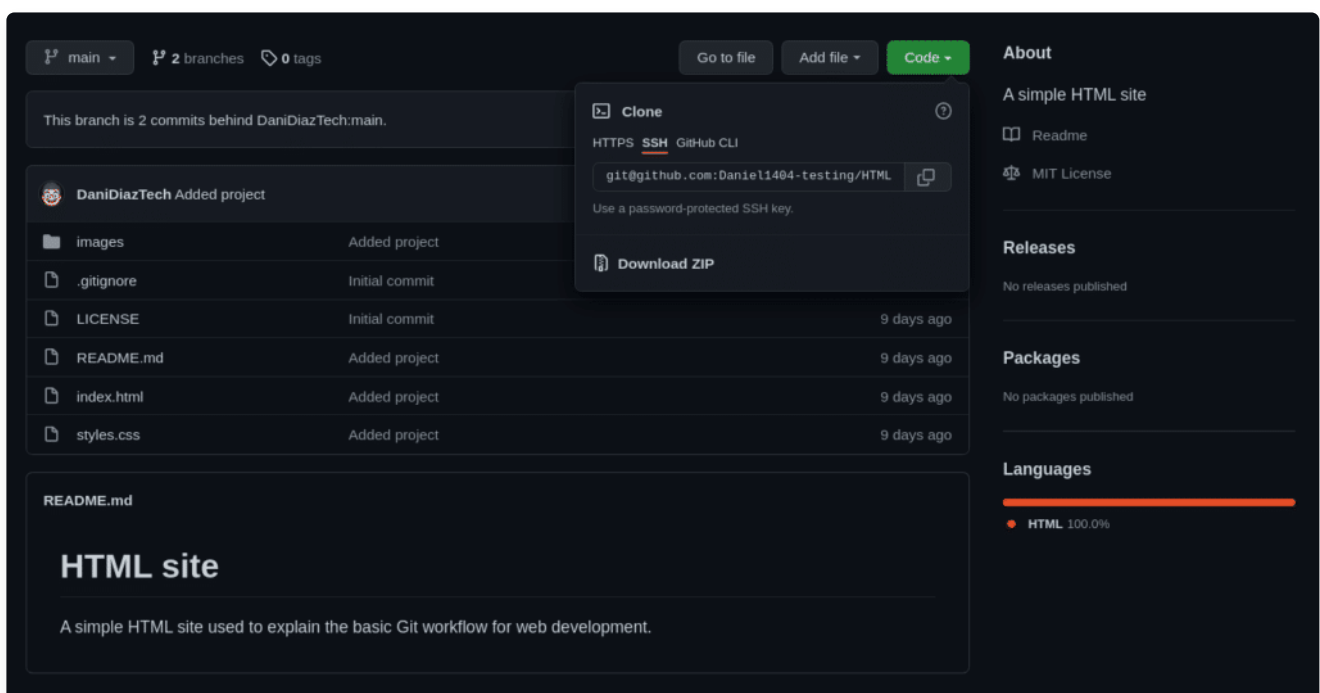
Ahora tienes un fork del repo original que sólo está disponible en tu cuenta de GitHub. Es exactamente el mismo repositorio, hasta que empiezas a hacer cambios.

Como puedes ver, bifurcar un repositorio público sólo lleva un par de segundos. Esto es genial para los proyectos de código abierto, pero ten en cuenta que si tu organización tiene un repositorio privado, tendrás que incluirte como colaborador antes de intentar bifurcarlo.

Es el momento de llevar tu fork a tu máquina local. Para ello, tienes que clonarlo con el comando `git clone` , que recupera el repositorio Git del servidor remoto:

```
git clone remote_url
```

Tienes que sustituir `remote_url` por la URL de tu fork. Para obtener la URL exacta de un repositorio de GitHub, ve a su página y haz clic en **Código**. Luego elige **SSH**, y copia el enlace que te da:





El comando que debes ejecutar para clonar el repo bifurcado es

```
git clone git@github.com:yourusername/HTML-site.git
```

Cuando clonas un repo, obtienes una carpeta con su nombre. Dentro de esa carpeta está el código fuente del proyecto (en este caso, el sitio HTML) y el repositorio Git, que se encuentra dentro de una carpeta llamada **.git**.

Puedes ver la lista de archivos dentro del nuevo directorio abriendo la nueva carpeta en un gestor gráfico de archivos, o listándolos directamente desde el terminal con los comandos

```
ls o dir :
```

```
# Linux/macOS
ls HTML-site/
# Windows
dir HTML-site\
.git images .gitignore index.html LICENSE README.md styles.css
```

Este sitio HTML es muy sencillo. Utiliza Bootstrap para fines prácticos y algunas imágenes de [Unsplash](#), donde puedes [descargar imágenes gratuitas para tu sitio](#).

Si abres el archivo **index.**html en tu navegador, verás una página sencilla con algunas imágenes:





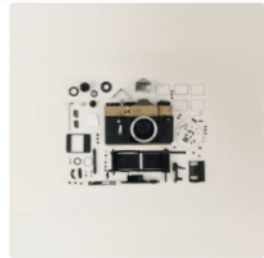
Super computer



Cool computer



Normal laptop



Old camera

— La sencilla página web que estamos creando

Es hora de jugar con este proyecto. Parece muy vacía, y quizá una cabecera con el nombre del sitio podría mejorar la experiencia del usuario.

Para ello, entra en el directorio **HTML-site** y crea una rama llamada `header`. En esta nueva rama, podemos editar todos los archivos e implementar todo el código que queramos porque no afectará a la rama principal (original).

Ejecuta el siguiente comando:

```
git checkout -b header
```

Esto creará una rama llamada «cabecera» y te cambiará a ella justo después de esto. Equivale a

```
git branch header  
git checkout header
```

Para confirmar que todo ha ido bien, ejecuta

```
git status  
  
# On branch header  
  
# nothing to commit, working tree clean
```

Verás que ha sido desplazado de la rama «principal» a la rama «cabecera», pero el árbol de trabajo sigue limpio ya que no hemos editado ningún archivo.

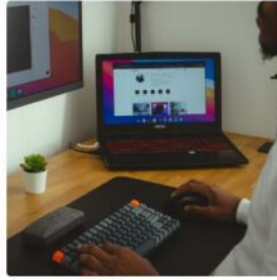
En tu [editor de código favorito](#), abre el archivo **index.html** de tu proyecto ramificado. Este archivo incluye algunos enlaces a [Bootstrap 5](#), para que podamos aprovechar los componentes listos para usar del framework.

Añade el siguiente código al archivo **index.html** dentro de la etiqueta `<body>` y encima del contenedor de imágenes:

```
<header>  
  <nav class="navbar navbar-light bg-light">  
    <div class="container-fluid">  
      <span class="navbar-brand mb-0 h1">HTML site</span>  
    </div>  
  </nav>  
</header>
```



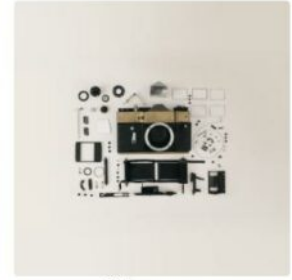
Super computer



Cool computer



Normal laptop



Old camera

— Nuestra página web con una nueva cabecera

¡Se ve mucho mejor! Siéntete libre de hacer otras ramificaciones y cambios que desees.

Una vez que hayas terminado de editar el proyecto, es el momento de confirmar todos los cambios en tu repositorio local. Dentro del directorio del proyecto, escribe lo siguiente en tu terminal:

```
git add --all  
git commit -m "Added simple header in index.html file"
```

### Important

Todos los mensajes de confirmación de Git deben ser claros y significativos. Cada vez que añadas una confirmación, tú y tus compañeros de equipo debéis ser capaces de identificar lo que ha ocurrido en esa confirmación, de modo que si se produce un error, podáis solucionarlo fácilmente

mantener una buena práctica de nomenclatura.

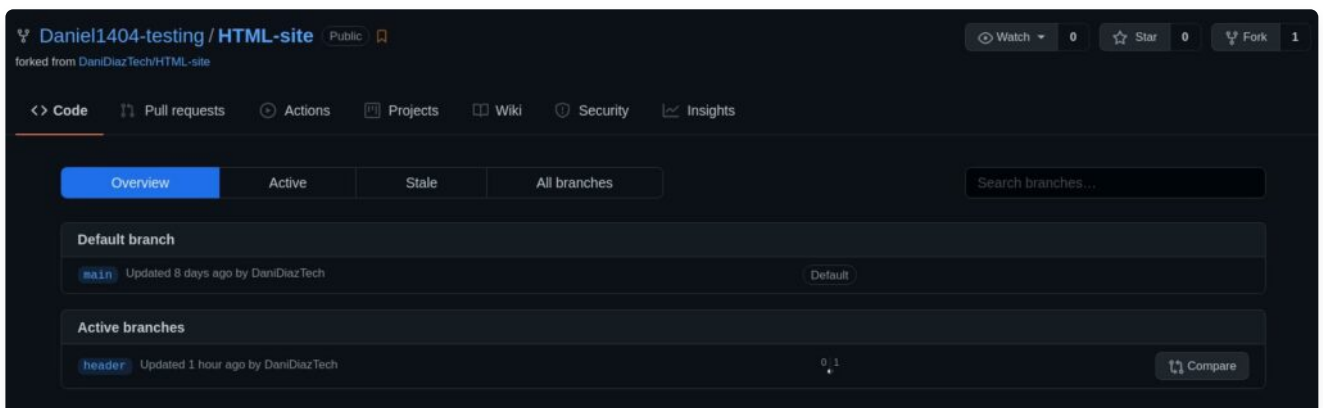
Ahora que has hecho un commit en tu repositorio local (que aún está disponible sólo en tu ordenador), es el momento de enviarlo al repositorio remoto.

Si intentas enviar la confirmación de forma normal, no funcionará porque estás trabajando en la rama `header`. Tienes que establecer la rama de subida para `header`:

```
git push --set-upstream origin header
```

A partir del 13 de agosto de 2021, GitHub requiere el uso de la autenticación SSH, así que asegúrate de que tienes tus claves correctamente configuradas.

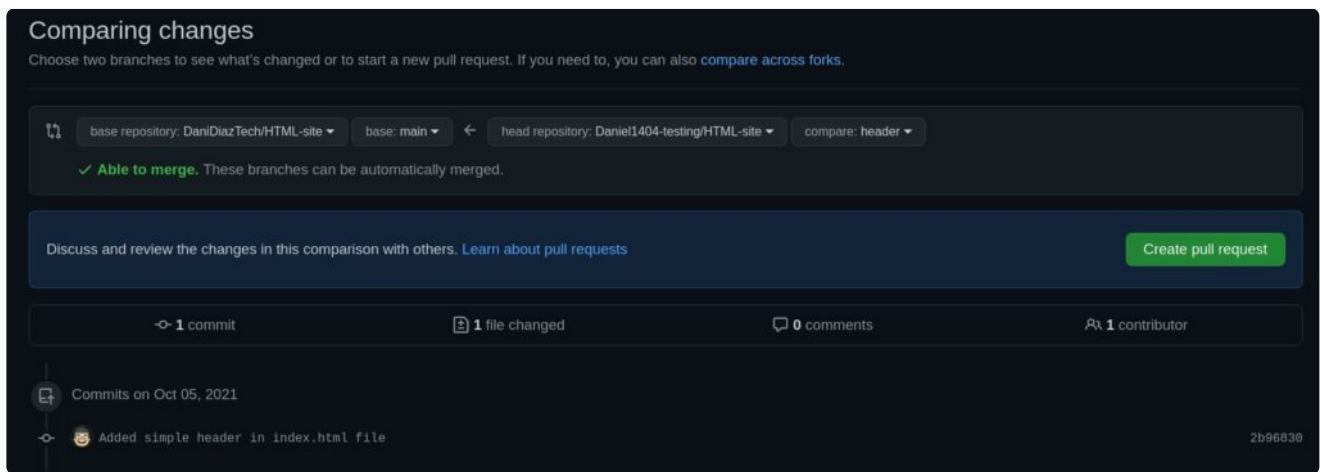
Después de esto, deberías poder ver una nueva rama llamada `header` en tu repositorio bifurcado (por ejemplo, <https://github.com/yourusername/HTML-site/branches>):



— La rama «cabecera»

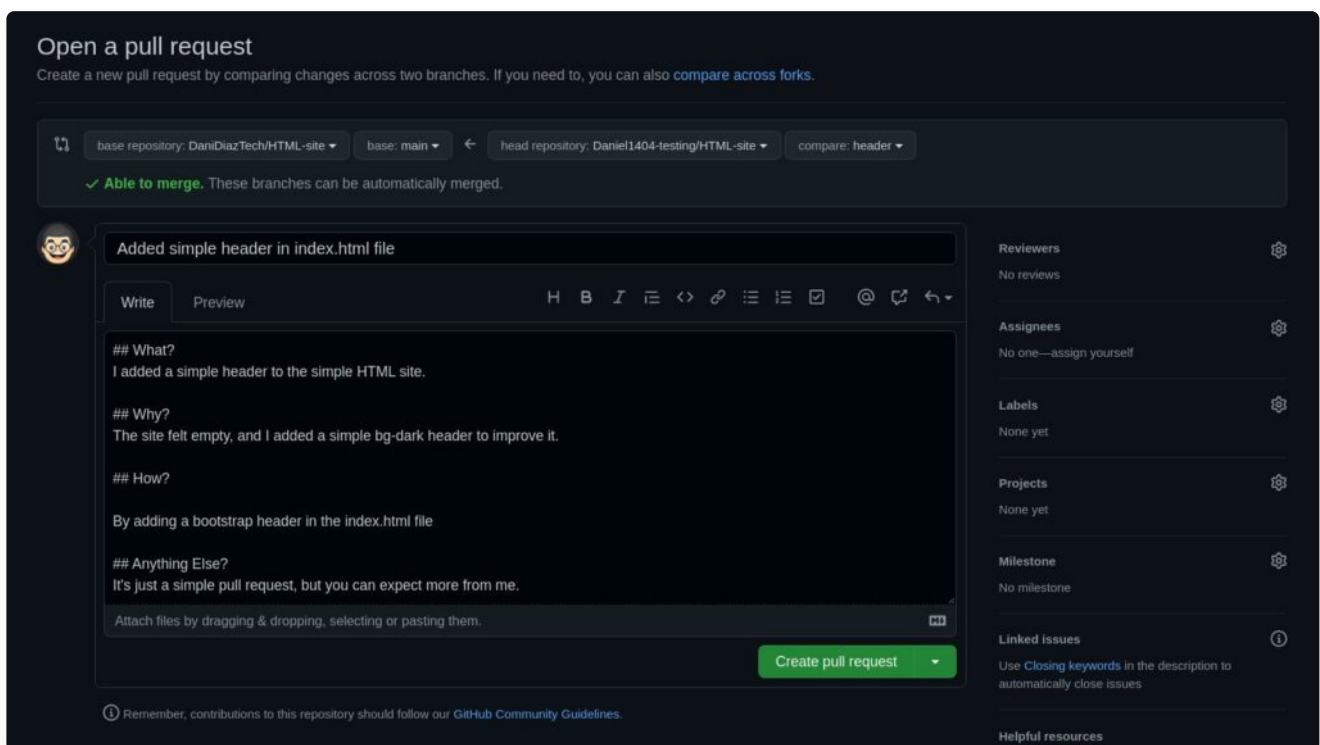
Para crear un pull request al repo original, haz clic en **Comparar**, abajo en la sección **Ramas activas**.

Esto te guiará a un pull request, donde tendrás que elegir con qué rama (la original o tu fork) quieres fusionar. Por defecto, aparece la opción de fusionar con el repositorio base:



— Creando pull requests en GitHub

Una vez que hagas clic en la opción de solicitud de extracción, tendrás que escribir una breve descripción de los cambios realizados, al igual que con tus confirmaciones anteriores. Una vez más, intenta ser conciso pero descriptivo:



— Escribiendo un mensaje de solicitud de extracción

Haz clic en el botón **Crear pull request** y espera a que el propietario del repositorio base acepte o te dé su opinión sobre tus cambios.

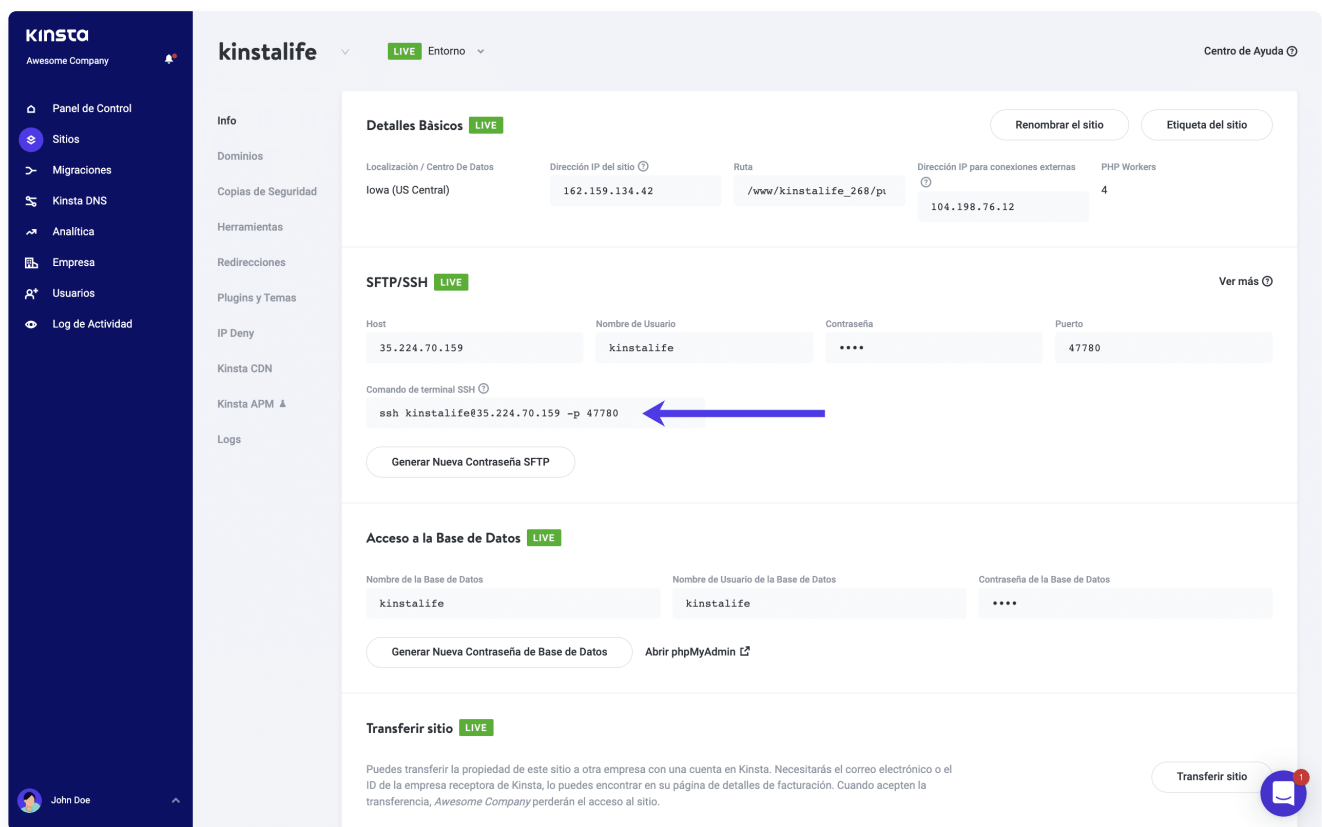
Este ha sido un ejemplo muy básico, pero la lógica se extiende a proyectos de todos los tamaños. Asegúrate de implementar este flujo de trabajo de cerca en proyectos de colaboración más grandes también.

## Cómo usar Git en Kinsta

Si eres [usuario de Kinsta](#), ya tienes dos formas de utilizar Git y GitHub desde tu [portal MyKinsta](#).

Empecemos con la primera opción. Puedes [entrar fácilmente por SSH](#) y sacar un repo desde cualquier servicio de alojamiento de Git como GitHub, Gitlab o Bitbucket.

Para ello, ve a tu pestaña de **Sitios**, selecciona un sitio, y ve a [tu sección de detalles SSH](#), y copia el comando de terminal SSH.



The screenshot shows the Kinsta dashboard interface. On the left is a dark blue sidebar with the 'Kinsta' logo and a list of navigation items: Panel de Control, Sitios, Migraciones, Kinsta DNS, Analítica, Empresa, Usuarios, and Log de Actividad. The main content area is titled 'kinstalife' and shows details for a 'LIVE' environment. It includes sections for 'Detalles Básicos' (Basic Details), 'SFTP/SSH' (SFTP/SSH), 'Acceso a la Base de Datos' (Database Access), and 'Transferir sitio' (Transfer site). A blue arrow points to the 'Comando de terminal SSH' field, which contains the command: `ssh kinstalife@35.224.70.159 -p 47780`. The 'Transferir sitio' button at the bottom right has a red notification badge with the number '1'.

— Sección de detalles SSH.

Accede por SSH a tu sitio pegando el comando anterior en tu terminal y entrando en la



todos tus archivos de WordPress, para que puedas sacar un repositorio Git con un tema o plugin personalizado en el que hayas estado trabajando.

Así es como se extrae un repositorio Git con un simple comando:

```
ssh yourkinstasite@1.2.3.4 -p PORT "cd /www/my_site/public && git pull https://g
```

### Important

El SSH debe ser utilizado por usuarios avanzados. Si no sabes hacerlo, no dudes en ponerte en contacto primero con [el soporte de Kinsta](#).

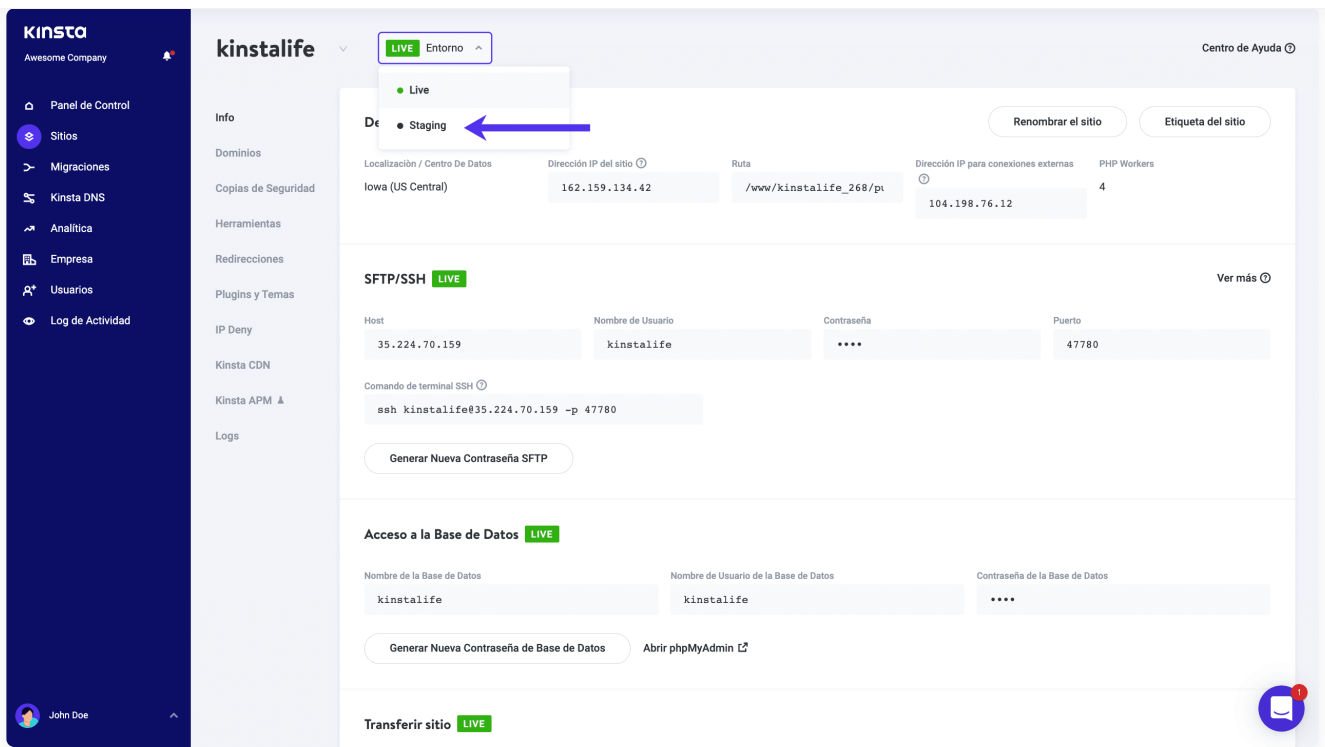
Ahora, introduciendo la nueva función de despliegue de GitHub en Kinsta, puedes desplegar un sitio completo de WordPress desde un repositorio de GitHub.

Tu repositorio de GitHub debe incluir una copia de los [archivos del núcleo de WordPress](#) y, por supuesto, el contenido de tu sitio dentro de la carpeta **wp-content**.

Echemos un vistazo a esta opción.

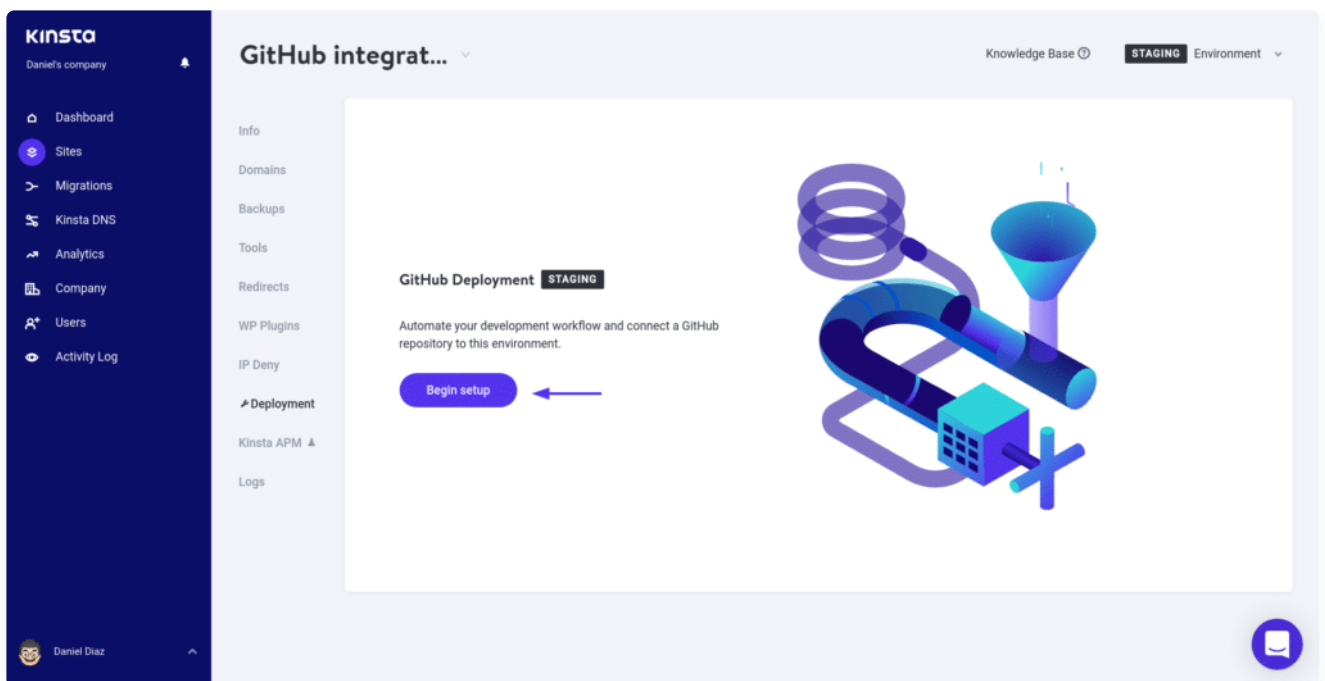
Ve a uno de los sitios de tu empresa y [crea un entorno staging](#). Esto no te llevará más de un par de minutos.





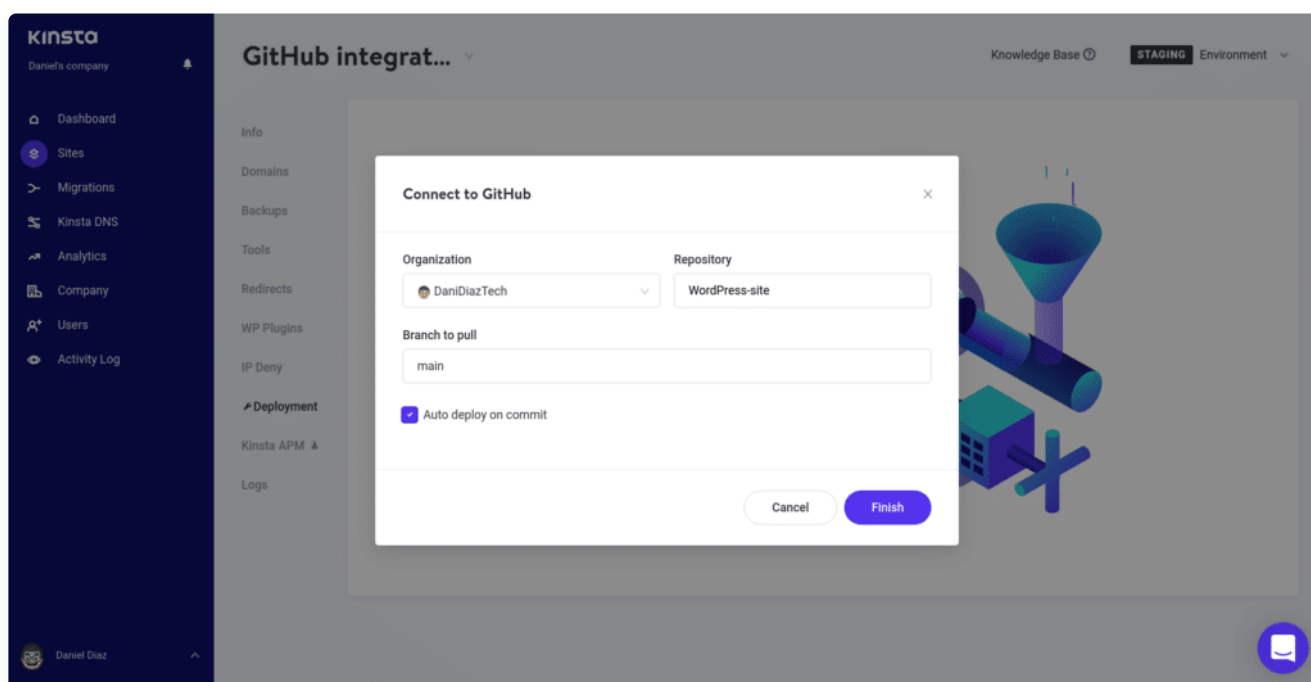
— Entorno staging.

Una vez que estés en tu sitio staging, ve a la pestaña **Despliegue** y haz clic en el botón **Comenzar configuración**. Verás un modal de GitHub que permitirá a Kinsta conectarse con tu cuenta de GitHub.



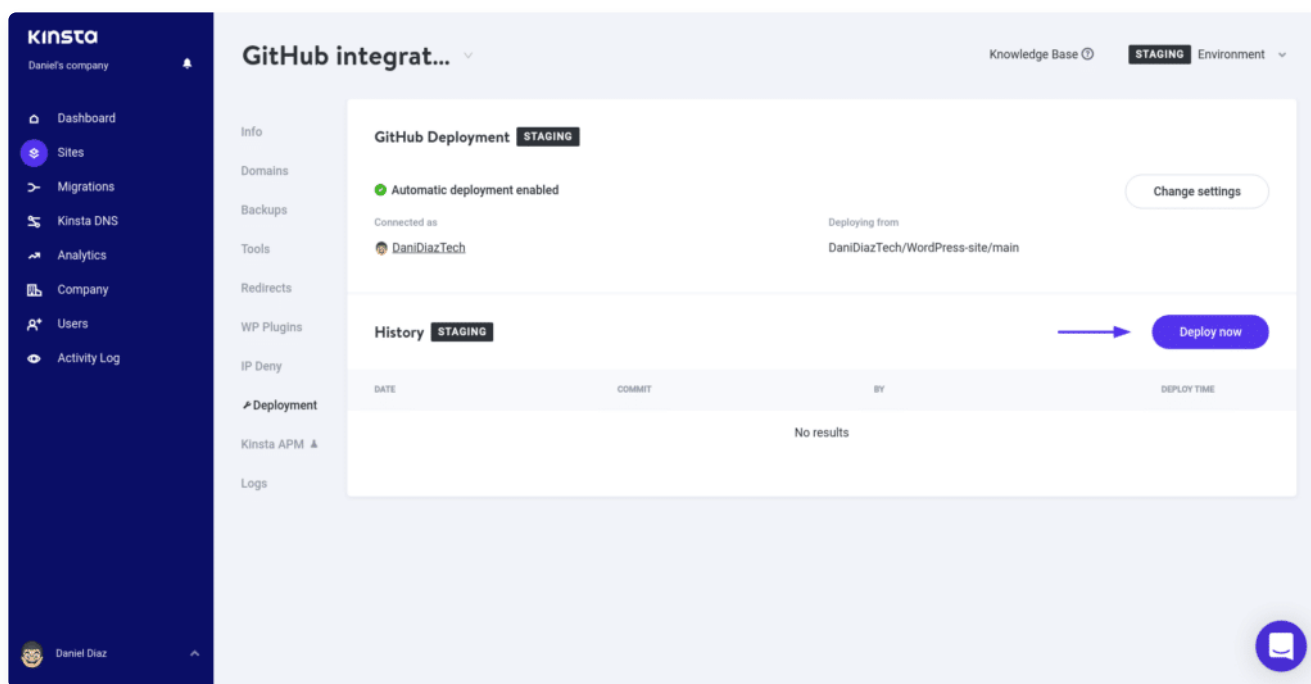
— Pestaña de despliegue de GitHub

Ahora, selecciona el repositorio del que vas a extraer tu sitio.



— Conecta Kinsta a GitHub

Finalmente, despliega tu sitio y visítalo a través de la URL de tu sitio staging.



— Botón «Desplegar ahora»

Esta función aún está en **Beta**, pero pronto todos los usuarios de Kinsta tendrán acceso a ella.

El uso de Git y Kinsta puede ser una poderosa combinación si sabes utilizarlos bien. Aunque nuestro tutorial aquí presenta sólo un ejemplo sencillo, puedes aprender mucho más en [nuestro artículo de la base de conocimientos sobre Git](#).

## “ Aumenta tus conocimientos sobre Git con esta guía sobre el flujo de trabajo de un proyecto típico ”

CLIC PARA TUITEAR

## Resumen

Hoy en día, Git es una herramienta que debes aprender para el desarrollo web, ya que la mayor parte del tiempo estarás colaborando con otros para crear el mejor proyecto posible.

En este artículo, hemos hablado de algunas razones importantes para utilizar Git en tus proyectos, y te hemos mostrado el flujo de trabajo básico para colaborar en un repo de Git.

Git es una herramienta tan potente que puedes extender su uso incluso al [alojamiento de WordPress](#), por lo que sólo puede beneficiarte aprenderlo e implementarlo como parte de tu arsenal de habilidades de desarrollo web.

*¿Tienes alguna otra sugerencia para mejorar este flujo de trabajo básico de Git para el desarrollo web? ¡Háznoslo saber en la sección de comentarios!*

Consigue todas tus [aplicaciones](#), [bases de datos](#), and [Sitios de WordPress](#) online y bajo un mismo techo. Nuestra plataforma en la nube de alto rendimiento y repleta de funciones incluye:

- Fácil configuración y gestión en el panel de MyKinsta
- Soporte experto 24/7
- El mejor hardware y red de Google Cloud Platform, con tecnología Kubernetes para una escalabilidad máxima
- Una integración de Cloudflare a nivel empresarial para mayor velocidad y seguridad
- Alcance de audiencia global con hasta 35 centros de datos y 275 PoPs en todo el mundo



Empieza con una prueba gratuita de nuestro [Alojamiento de Aplicaciones](#) o [Alojamiento de Bases de Datos](#). Explora nuestros planes o [habla con el departamento de ventas](#) para encontrar el que mejor se adapte a ti.

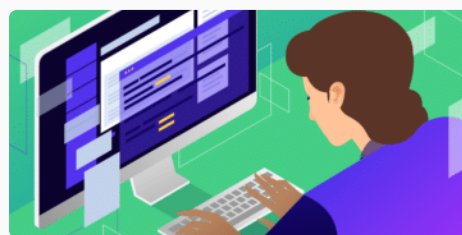
## Artículos relacionados

### BLOG

#### Guía para Convertirse en un Desarrollador Frontend: Competencias y Responsabilidades Laborales

Esta sencilla guía te guiará a través de todo lo necesario para convertirte en un desarrollador frontend de éxito.

16 MIN DE LECTURA • DICIEMBRE 30, 2021



### BLOG

#### 61 Impresionantes Herramientas de Desarrollo Web para Usar en 2023

Explora 61 increíbles herramientas de desarrollo web que puedes aprovechar para impulsar tus habilidades de desarrollo web y tu productividad.

85 MIN DE LECTURA • JUNIO 3, 2021



### BLOG

#### GitLab vs GitHub: Descubre Sus Principales Diferencias y Similitudes

Descubre las principales diferencias y similitudes entre GitLab y GitHub. Te mostraremos cómo elegir la solución Git más adecuada para tu equipo.

20 MIN DE LECTURA • ABRIL 15, 2021



# Deja una respuesta

**Política de comentarios:** Nos encantan los comentarios y agradecemos el tiempo que los lectores dedican para compartir ideas y para retroalimentación. Sin embargo, todos los comentarios son moderados manualmente y aquellos que consideramos como spam o comentario promocional se eliminarán.

**Comentario**

**Nombre**

**Email**

Al enviar este formulario: Usted acepta el procesamiento de los datos personales enviados de acuerdo con los [Política de Privacidad](#) de Kinsta, incluyendo la transferencia de datos a los Estados Unidos.

- ☒ También acepta recibir información de Kinsta relacionada con nuestros servicios, eventos y promociones. Puede darse de baja en cualquier momento siguiendo las instrucciones de las comunicaciones recibidas.

Publicar el comentario





## Suscríbete a nuestro Newsletter

Mantente al día de las últimas tendencias en desarrollo web, frameworks y lenguajes.

Suscribirse

- Kinsta Hosting** >
- Plataforma** >
- Recursos** >
- Empresa** >
- Comparar Kinsta** >

**KINSTA**



Español



© 2023 Kinsta Inc. Todos los derechos reservados. Kinsta® y WordPress® son marcas registradas.  
Aviso legal.



