

[Comparte](#)
[en Twitter](#)
[Envía](#)
[a LinkedIn](#)
[Email](#)
[Programación](#)
[Comentario](#)

GitHub vs GitLab: ventajas y desventajas de estas plataformas



VS



Aunque ambas tienen similitudes, incluso en el propio nombre que comienza por Git debido a que ambas se basa en la famosa herramienta de control de versiones escrita por Linus Torvalds, pero **ni una ni otra son exactamente iguales**. Por ello, el ganador de la batalla GitHub vs GitLab no está tan claro, tienen algunas diferencias que hacen que tengan sus ventajas y desventajas para los usuarios y desarrolladores que las suelen usar.

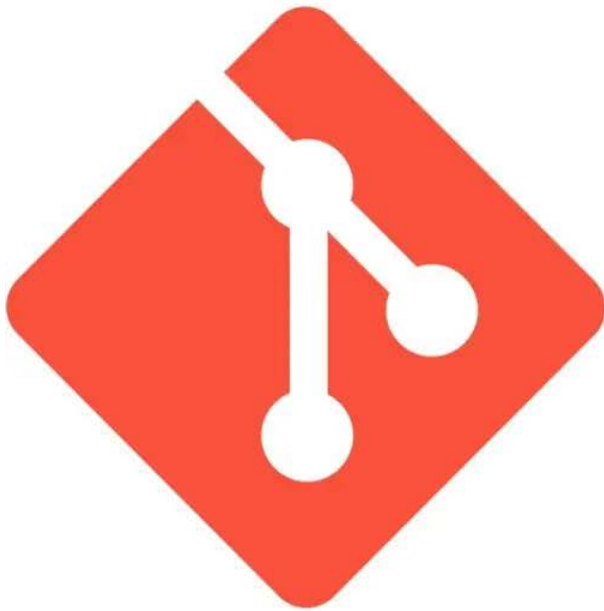
Por otro lado, algunos desarrolladores se han movido en los últimos tiempos hacia GitLab, con sus consecuencias positivas y negativas que ahora conocerás. El motivo de ese evento fue la compra de la plataforma GitHub por parte de Microsoft, y **las dudas que eso generó**. Pero, para ser honestos, la plataforma continúa funcionando con normalidad por el momento...

Índice

- 1 ¿Qué es Git?
- 2 ¿Qué es GitHub?
- 3 ¿Qué es GitLab?
- 4 GitHub vs GitLab
 - 4.1 Diferencias GitHub vs GitLab
 - 4.2 Ventajas y desventajas de GitLab
 - 4.2.1 Ventajas
 - 4.2.2 Desventajas
 - 4.3 Ventajas y desventajas de GitHub
 - 4.3.1 Ventajas

[Comparte](#) [Desventajas](#)
[Twitea](#)
[5 Conclusion](#)
[Envia](#)
[Pinea](#)
[Email](#)
[Print](#)

¿Qué es Git?



git

Git es un software de control de versiones que ideó Linus Torvalds para el kernel Linux, ya que no le convencían otros programas similares existentes. A pesar de que se hizo específicamente para el proyecto Linux, actualmente se ha expandido a otros muchos proyectos de código abierto por sus ventajas.

En su origen, se escribió pensando en la **eficiencia, confiabilidad, y compatibilidad** para proyectos que tienen un gran número de ficheros de código fuente.

En cuanto a lo que significa un software de **control de versiones**, como también lo es VCS, Subversion, CVS, entre otros, simplemente es un software de gestión de cambios que se realizan sobre elementos de un código fuente o su configuración. De esa forma, el equipo de desarrolladores independiente que trabajan sobre él pueden tener un mejor control y no se pisarían el trabajo o generarían problemas mientras colaboran en estos proyectos...

¿Qué es GitHub?

[Comparte](#)
[Twitea](#)
[Envía](#)
[Pinea](#)
[✉ Email](#)
[🖨 Print](#)



GitHub

GitHub es una plataforma de desarrollo colaborativo, también llamado forja. Es decir, una plataforma enfocada hacia la cooperación entre desarrolladores para la difusión y soporte de su software (aunque poco a poco se ha ido usando para otros proyectos más allá del software).

Como su propio nombre indica, se apoya sobre el **sistema de control de versiones Git**. Así, se puede operar sobre el código fuente de los programas y llevar un desarrollo ordenado. Además, esta plataforma está escrita en Ruby on Rails.

Tiene una enorme cantidad de proyectos de código abierto almacenada en su plataforma y accesibles de forma pública. Tal es su valor que **Microsoft optó por comprar esta plataforma** en 2018, aportando una cifra de nada menos que de 7500 millones de dólares.

Pese a las dudas sobre esa compra, la plataforma continuó operando como de costumbre, y continúa siendo **una de las más usadas**. En ella se alojan proyectos tan importantes como el propio kernel Linux...

[Más información](#)

¿Qué es GitLab?

[Comparte](#)
[Twitea](#)
[Envía](#)
[Pinea](#)
[✉ Email](#)
[🖨 Print](#)



GitLab es otra alternativa a GitHub, otro sitio de forja con un servicio web y un sistema de control de versiones también basado en Git. Por supuesto, se ideó para el alojamiento de proyectos de código abierto y para facilitar la vida de los desarrolladores, pero existen algunas diferencias con el anterior.

Esta web, además de la **gestión de repositorios** y control de versiones, también ofrece alojamiento para wikis, y sistema de seguimiento de errores. Una completa suite para crear y gestionar los proyectos de todo tipo, ya que, al igual que GitHub, actualmente se alojan proyectos que van más allá del código fuente.

Fue escrito por unos desarrolladores ucranianos, Dmitry Zaporozhets y Valery Sizov, usando lenguaje de programación Ruby y algunas partes en Go. Después se mejoró su arquitectura con Go, Vue.js, y **Ruby on Rails**, como en el caso de GitHub.

A pesar de ser muy conocida y ser la gran alternativa a GitHub, no cuenta con tantos proyectos. Eso no quiere decir que la cantidad de código alojado sea muy grande, con organizaciones que confían en ella **de la talla del CERN, la NASA, IBM, Sony**, etc.

[Más información](#)

GitHub vs GitLab



Personalmente, te diría que no existe un claro ganador en la **batalla GitHub vs GitLab**. No es tan sencillo elegir una plataforma que sea infinitamente superior a la otra, de hecho, cada una tiene sus puntos fuertes y sus puntos débiles. Y todo dependerá de lo que realmente busques para que te tengas que decantar por una u otra.

Diferencias GitHub vs GitLab

A pesar de todas las similitudes, una de las claves a la hora de decantarte en la comparativa GitHub vs GitLab pueden ser **las diferencias** entre ambas:

- **Niveles de autenticación:** GitLab puede establecer y modificar permisos a los diferentes colaboradores según su función. En el caso de GitHub, puede decidir quién tiene derechos de lectura y escritura en un repositorio, pero es más limitado en ese sentido.
- **Alojamiento:** aunque ambas plataformas permiten alojar el contenido de los proyectos en las propias plataformas, en el caso de GitLab también te puede permitir autoalojar tus repos, lo que puede ser una ventaja en algunos casos. GitHub ha agregado esa característica también, pero solo con ciertos planes de pago.
- **Importación y exportación:** GitLab contiene información muy detallada de cómo poder importar proyectos para moverlos de una plataforma a otra, como GitHub, Bitbucket, o traerlos a GitLab. Además, a la hora de exportar, GitLab ofrece un trabajo muy sólido. En el caso de GitHub no se ofrece documentación detallada, aunque se puede usar GitHub Importer como herramienta, aunque puede ser algo más restrictivo cuando se trata de exportar.

- **Comunidad:** ambos tienen una buena comunidad tras de sí, aunque GitHub parece haber ganado la batalla en popularidad. Actualmente aglutina millones de desarrolladores. Por eso, será más sencillo encontrar ayuda al respecto.
- **Versiones Enterprise:** ambos las ofrecen si pagas la cuota, por lo que se podría pensar que la comparativa GitHub vs GitLab no tiene sentido en este punto, pero lo cierto es que GitLab ofrece unas prestaciones muy interesantes, y se ha hecho popular entre los equipos de desarrollo muy grandes.

En resumen, **las diferencias** GitHub vs GitLab resumidas las tienes en esta tabla:

Características	GitLab Septiembre de 2011	GitHub Abril de 2008
Inicio		
Plan gratuito	Repositorios públicos y privados ilimitados	Gratis solo para repositorios públicos
Planes pagados	Desde 19\$ por usuario por año para el Plan Premium. O 99\$ por usuario y año para el Ultimate.	Desde \$4 por usuario y año para el Team, 21\$ para el Enterprise, o más para el One.
Funciones de revisión de código	sí	sí
Wiki	sí	sí
Seguimiento de errores y problemas	sí	sí
Rama privada	sí	sí
Sistema de construcción	sí	sí (con servicio de terceros)
Importar proyectos	sí	No
Exportar proyectos	sí	No
Seguimiento del tiempo	sí	No
Alojamiento web	sí	sí
Autohospedaje	sí	sí (con plan empresarial)
Popularidad	546.000+ proyectos	69.000.000+ proyectos

Ventajas y desventajas de GitLab

Una vez conocidas las diferencias y semejanzas entre GitHub vs GitLab, las ventajas y desventajas de estas plataformas **te pueden ayudar a decidirte**.

Ventajas

- Plan gratuito y sin limitaciones, aunque tiene planes de pago.
- Es de licencia de código abierto.
- Permite el autohospedaje en cualquier plan.
- Está muy bien integrado con Git.

Desventajas

- Su interfaz puede ser algo más lenta con respecto a la competencia.

- [Comparte](#) algunos problemas habituales con los repositorios.



Ventajas y desventajas de GitHub

Por otro lado, GitHub también tiene sus **pros y contras**, entre los que destacan los siguientes:

Ventajas

- Servicio gratuito, aunque también tiene servicios de pago.
- Búsqueda muy rápida en la estructura de los repos.
- Amplia comunidad y fácil encontrar ayuda.
- Ofrece prácticas herramientas de cooperación y buena integración con Git.
- Fácil integrar con otros servicios de terceros.
- Trabaja también con TFS, HG y SVN.

Desventajas

- No es absolutamente abierto.
- Tiene limitaciones de espacio, ya que no puedes exceder de 100MB en un solo archivo, mientras que los repositorios están limitados a 1GB en la versión gratis.

Conclusión

Como ves, **no existe un claro ganador**. La elección no es fácil y, como comenté, deberías vigilar muy bien las ventajas, desventajas y diferencias de cada uno para poder identificar cuál se adapta mejor a tus necesidades.

Personalmente te diría que si quieres disponer de un entorno totalmente abierto, mejor usa GitLab. En cambio, si prefieres más facilidades y usar el servicio web con más presencia, entonces ve a por GitHub. Incluso incluiría **un tercero en discordia** y te diría que si buscas trabajar con servicios Atlassian deberías mirar del lado de [Bitbucket](#)...

El contenido del artículo se adhiere a nuestros principios de [ética editorial](#). Para notificar un error pincha [aquí](#).

Ruta completa hacia el artículo: [Desde Linux](#) » [Programación](#) » **GitHub vs GitLab: ventajas y desventajas de estas plataformas**

Te puede interesar

Un comentario, deja el tuyo

<https://blog.desdelinux.net/github-vs-gitlab/>