

GitLab / GitHub

DigitalHouse>



**Certified Tech
Developer**

The Ultimate Degree

Índice

1. [Git basics](#)
2. [GitLab](#)
3. [GitHub](#)
4. [Comparación entre GitLab y GitHub](#)
5. [Conclusión](#)

1 | Git basics

Git basics

¡Hagamos un repaso por Git! Este es un **sistema de control de versiones** —o VCS, por sus siglas en inglés de *version control system*— que nos ayuda a ordenar nuestro código.

Nos permite mantener instantáneas de nuestro código a través del tiempo (un historial) y en multiplicidad de versiones (ramas) que luego pueden ser fusionadas (integradas) para reunir todo el código en uno, completando la aplicación que se esté desarrollado en distintas partes.

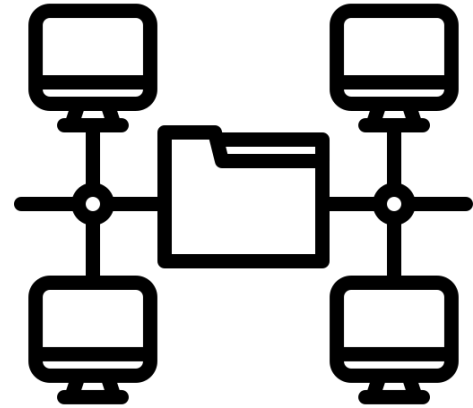


Repositorios locales y remotos en Git

Con Git tenemos repositorios locales y remotos. Cada desarrollador trabaja en su repositorio local hasta que tiene que fusionar sus cambios con los de otro. Esto último puede hacerse compartiendo el directorio de uno de los repositorios locales para que un compañero pueda subir sus cambios a un remoto mediante una conexión máquina a máquina (usando protocolos como samba, SSH HTTP, entre otros).

Git nació siendo una solución sin nube de por medio.

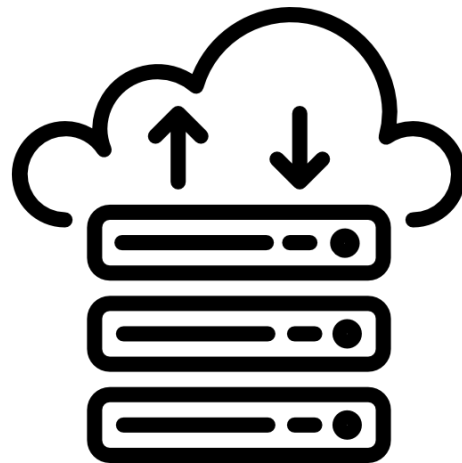
Tal así que para realizar una fusión de código los dev coordinaban por correo electrónico y luego compartían un repositorio.



Organización de los repositorios

Con el paso del tiempo aparecieron herramientas en la nube y con interface web, que nos facilitaron la organización de los repositorios y el seguimiento de funciones de código.

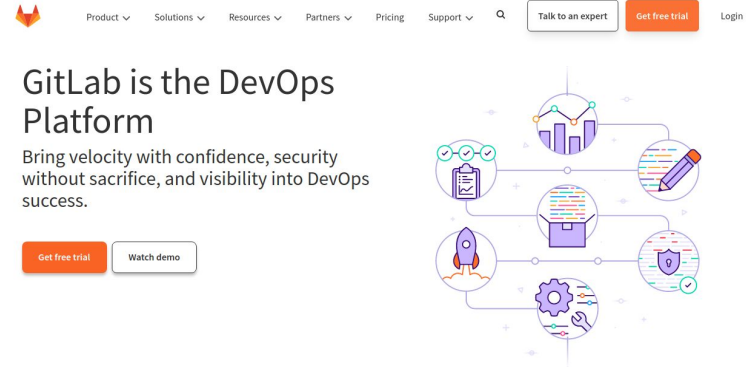
Las coordinaciones de devs por correo se integra en los sistemas y toman nombres **pull** o **merge request**. Se agregan además una serie de características que facilitan la organización de los repositorios de empresas u organizaciones.



2 | GitLab

GitLab

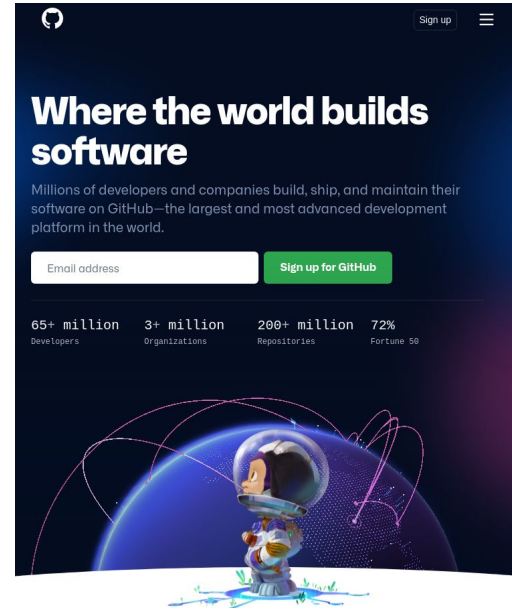
GitLab es una plataforma que utiliza el VSC Git y está **fuertemente centrada en la cultura DevOps**. En un principio era un gestor de repositorios, pero luego evolucionó agregando herramientas de CI/CD además de otras como wiki y gestión de proyectos. Suele ser conocida como una plataforma “todo en uno”, aunque también puede integrarse con otras tecnologías.



3 | GitHub

GitHub

GitHub es una plataforma de almacenamiento de repositorios Git muy conocida por contener grandes proyectos open source públicos para todos los devs. Es un lugar donde muchos desarrolladores publican el trabajo que quieren compartir con el resto del mundo. Al igual que GitLab, en sus comienzos no contaba con herramientas propias de CI/CD, pero se agregaron recientemente con el nombre de Github Actions.



4 | Comparación entre GitLab y GitHub

GitLab / GitHub

Si bien ambas herramientas comparten algunas características, ¡vamos a compararlas!

Criterio	GitLab	GitHub
Open source	Únicamente para la community edition	No lo es
Repositorios públicos	Se pueden hacer públicos	Sí
Repositorios privados	Sí, por defecto	Sí, limitados
Navegación del repositorio	Sí	Sí
CI/CD	Sí: GitLab CI/CD, GitLab Runners	Sí: GitHub Actions
Análisis de proyectos	Sí	No

Ventajas y desventajas

Criterio	GitLab	GitHub
Ventajas	<ul style="list-style-type: none">» Está disponible libremente, es open source en la community edition.» Es cloud native y altamente segura.» Permisos altamente ajustables.» No hay límite de colaboradores	<ul style="list-style-type: none">» Es muy usado para compartir el trabajo frente al público.» Menores precios empresariales o particulares.
Desventajas	<ul style="list-style-type: none">» Es robusto, por lo tanto, a veces tarda en cargar recursos.» Suele tener muchas mejoras pero también muchos bugs.	<ul style="list-style-type: none">» No es open source. Si queremos instalarlo hay que pagar.» Límite de colaboradores gratuitos.

5 | Conclusión

Conclusión

En esta presentación exploramos la plataforma de gestión de código **GitHub** —basada en el sistema de control de versiones Git— y realizamos una breve comparación con **GitLab**. ¡Ambas son soluciones muy potentes para CI/DC!

GitLab y GitHub tienen una influencia importante en la cultura DevOps y en los procesos de desarrollo de software e infraestructura como código. ¡Pasamos de tener herramientas separadas para usos específicos a plataformas que las integran, ayudando a derribar los silos en los equipos que las usan y las mantienen!



DigitalHouse>