

Introducción a git y GitHub

PSP 2025-2026

Qué es Git



GIT es un sistema de control de versiones:

- Guarda un historial de los cambios
- Permite abordar desarrollos colaborativos
- Permite saber qué cambios se han hecho, cuándo, y quién los realizó
- Permite revertir los cambios y volver a un estado previo
- Permite el control de versiones distribuido
- Los usuarios pueden trabajar en local, y mantener toda la historia de cambios.

¿Qué es GitHub?



GitHub es el mayor servicio de hosting de repositorios git basado en web

- Es decir se utiliza como servidor “remote” en muchos proyectos, y permite la colaboración online


Añade funcionalidad encima de git

- UI, documentación, bug tracking, feature requests, pull requests, etc.


Además github se ha convertido en uno de los lugares en los que los desarrolladores muestran su trabajo.


- Github permite construir un portfolio online


Conceptos básicos de git


 **Repositorio:** Es el lugar donde se guarda todo el historial de cambios de un proyecto. Puede estar en tu ordenador(local) o en una plataforma como GitHub (remoto). Piensa en él como una carpeta mágica que recuerda cada modificación que haces.


 **Stage** (o "Staging Area"): Es una zona intermedia donde colocas los archivos que quieres incluir en el próximo commit. Es como preparar los ingredientes antes de cocinar: seleccionas lo que vas a usar, pero aún no lo cocinas.

 **Commit:** Es el momento en que "cocinas" los cambios: guardas una versión del proyecto con un mensaje que explica qué hiciste. Cada commit es como una foto del estado del proyecto en ese instante.

 **Rama** (Branch): Una rama es una línea de desarrollo independiente. Te permite trabajar en nuevas funcionalidades sin afectar la versión principal (generalmente llamada main o master). Ideal para experimentar sin romper nada.

 **Pull:** Es la acción de traer los últimos cambios desde el repositorio remoto a tu repositorio local. Es como sincronizar tu proyecto con lo que otros han hecho.

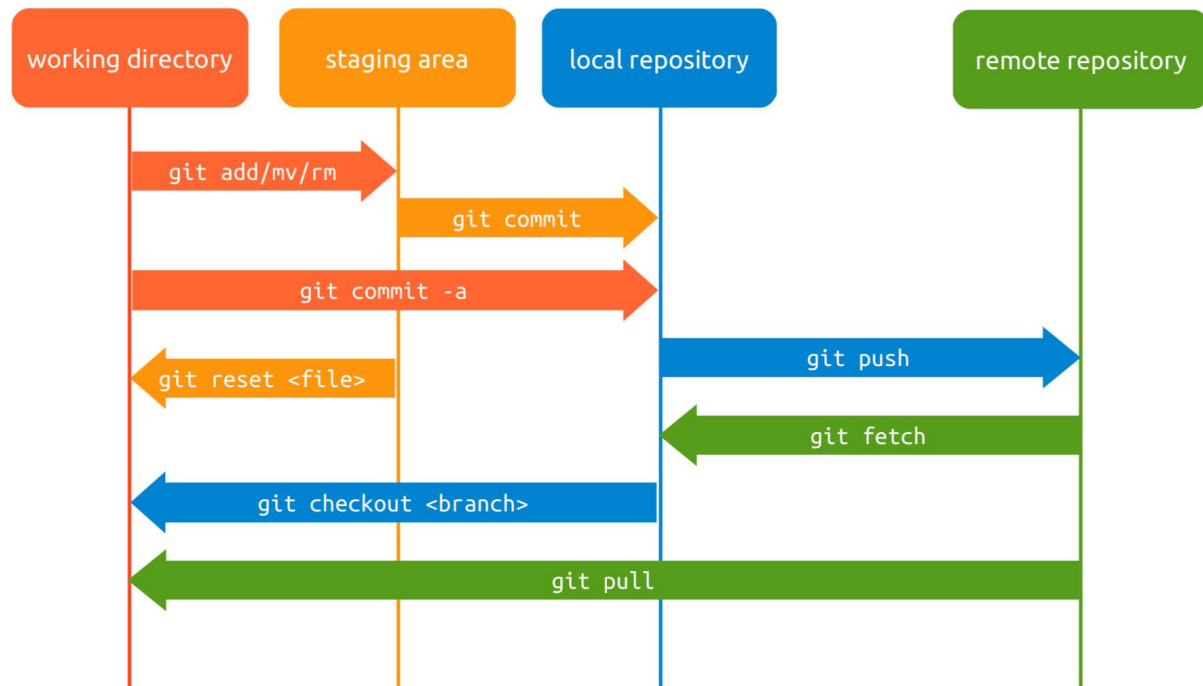
 **Push:** Es lo contrario de pull: envías tus cambios desde tu repositorio local al remoto. Así otros pueden ver y usar lo que tú has trabajado.

 **Merge:** Es el proceso de combinar los cambios de una rama con otra. Por ejemplo, cuando terminas una nueva funcionalidad en una rama, haces un merge con la rama principal para integrar tu trabajo.

Última diapositiva de teoría 🙌

Repositorio	Carpeta que guarda todo el historial del proyecto	Una caja fuerte con todas las versiones del trabajo
Stage	Zona donde preparas los archivos para guardar	Una bandeja donde colocas los documentos antes de archivarlos
Commit	Guardar una versión con un mensaje	Tomar una foto del proyecto en ese momento
Rama	Línea de desarrollo independiente	Una carretera secundaria para probar ideas sin afectar la principal
Pull	Traer cambios del repositorio remoto	Descargar la versión más reciente del proyecto
Push	Enviar tus cambios al repositorio remoto	Subir tu trabajo para que otros lo vean
Merge	Combinar dos ramas en una sola	Unir dos carreteras en una sola vía principal

Manos a la obra



Git se inventó como una herramienta de consola

Pero hoy en día existen miles de GUIs para usar git de forma más cómoda, nosotros vamos a usar VS Code