

¿Qué es un repositorio?

Un repositorio es un espacio de almacenamiento donde se guarda el código fuente de un proyecto junto con su historial de cambios. Este puede ser local (en una computadora) o remoto (en un servidor en línea), y permite la colaboración entre desarrolladores al gestionar versiones y cambios del código.

Ejemplos de proveedores online de repositorios:

1. **GitHub:** Un servicio popular para alojar repositorios Git, con funcionalidades colaborativas como pull requests, issues y acciones automatizadas.
2. **GitLab:** Otra plataforma basada en Git que ofrece gestión de repositorios y herramientas de integración/entrega continua (CI/CD).

Principales componentes de un versionamiento en GIT:

1. **Repositorio:** Contiene todos los archivos, historial de cambios y configuraciones.
2. **Commit:** Un registro de cambios aplicados a los archivos en el repositorio, con un mensaje descriptivo.
3. **Branch:** Una rama del repositorio donde se trabaja en paralelo para desarrollar nuevas funcionalidades o arreglar errores.
4. **Stage/Index:** Zona intermedia donde los archivos se preparan antes de ser confirmados con un commit.
5. **HEAD:** Un puntero que indica en qué rama o commit te encuentras trabajando.

Ventajas de GIT frente a otros sistemas de control de versiones:

1. **Distribuido:** GIT permite que cada desarrollador tenga una copia completa del historial del proyecto, a diferencia de sistemas centralizados como SVN.
2. **Velocidad:** Las operaciones locales (como commits o branch switching) son más rápidas porque no dependen de un servidor remoto.
3. **Branching eficiente:** Crear, gestionar y fusionar ramas es rápido y consume pocos recursos.
4. **Integración con herramientas modernas:** GIT tiene soporte en múltiples plataformas y se integra fácilmente con CI/CD y sistemas de gestión de proyectos.
5. **Seguridad:** Utiliza SHA-1 para identificar commits, lo que lo hace confiable frente a manipulaciones.

¿Qué es un repositorio?

Un repositorio es un espacio de almacenamiento donde se guarda el código fuente de un proyecto junto con su historial de cambios. Este puede ser local (en una computadora) o remoto (en un servidor en línea), y permite la colaboración entre desarrolladores al gestionar versiones y cambios del código.

Ejemplos de proveedores online de repositorios:

1. **GitHub:** Un servicio popular para alojar repositorios Git, con funcionalidades colaborativas como pull requests, issues y acciones automatizadas.
2. **GitLab:** Otra plataforma basada en Git que ofrece gestión de repositorios y herramientas de integración/entrega continua (CI/CD).

Principales componentes de un versionamiento en GIT:

1. **Repositorio:** Contiene todos los archivos, historial de cambios y configuraciones.
2. **Commit:** Un registro de cambios aplicados a los archivos en el repositorio, con un mensaje descriptivo.
3. **Branch:** Una rama del repositorio donde se trabaja en paralelo para desarrollar nuevas funcionalidades o arreglar errores.
4. **Stage/Index:** Zona intermedia donde los archivos se preparan antes de ser confirmados con un commit.
5. **HEAD:** Un puntero que indica en qué rama o commit te encuentras trabajando.

Ventajas de GIT frente a otros sistemas de control de versiones:

1. **Distribuido:** GIT permite que cada desarrollador tenga una copia completa del historial del proyecto, a diferencia de sistemas centralizados como SVN.
2. **Velocidad:** Las operaciones locales (como commits o branch switching) son más rápidas porque no dependen de un servidor remoto.
3. **Branching eficiente:** Crear, gestionar y fusionar ramas es rápido y consume pocos recursos.
4. **Integración con herramientas modernas:** GIT tiene soporte en múltiples plataformas y se integra fácilmente con CI/CD y sistemas de gestión de proyectos.
5. **Seguridad:** Utiliza SHA-1 para identificar commits, lo que lo hace confiable frente a manipulaciones.

Ejemplos de comandos básicos en GIT:

1. `git init`: Crea un nuevo repositorio Git.
2. `git clone [URL]`: Descarga un repositorio remoto a tu máquina.
3. `git add [archivo]`: Añade un archivo al área de preparación (staging).
4. `git commit -m "[mensaje]"`: Crea un commit con los cambios preparados.
5. `git push`: Envía los commits locales al repositorio remoto.

¿Qué son los branch y cuáles son sus funciones?

Un **branch** (rama) es una bifurcación del proyecto donde se puede trabajar de forma aislada para desarrollar nuevas características, solucionar problemas o realizar experimentos, sin afectar el código principal.

Funciones principales de los branches:

- Facilitar el trabajo colaborativo, permitiendo a los desarrolladores trabajar en diferentes ramas.
- Proteger el código estable, ya que los cambios en una rama no afectan al resto hasta ser fusionados.
- Simplificar la integración y revisión de cambios mediante herramientas como pull requests.

¿Cuál es el nombre del branch principal?

El nombre predeterminado del branch principal es **main** (anteriormente conocido como **master**, pero ha sido reemplazado en muchas plataformas para promover un lenguaje más inclusivo).