UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE CIENCIAS

Instalación de Git

Profesora

Verónica Esther Arriola Ríos

Autor

Arroyo Martínez Erick Daniel

ASIGNATURA

Introducción a Ciencias de la Computación

7 de febrero de 2024

Índice

1. ¿Qué es Git?[1]

`Git` ha alcanzado el estatus de estándar global para el control de versiones. Pero, ¿qué implica realmente? `Git` es un sistema de control de versiones distribuido, lo que implica que un clon local del proyecto constituye un repositorio de control de versiones completo. Estos repositorios locales plenamente operativos posibilitan trabajar de manera autónoma o de forma remota con facilidad. Los desarrolladores confirman sus cambios a nivel local y luego sincronizan sus repositorios locales con los del servidor. Este enfoque difiere del control de versiones centralizado, donde los clientes deben sincronizar el código con un servidor antes de crear nuevas versiones.

La versatilidad y popularidad de `Git` lo convierten en una elección sobresaliente para cualquier equipo. Muchos desarrolladores y graduados universitarios ya están familiarizados con el uso de `Git`. La comunidad de usuarios de `Git` ha generado recursos para capacitar a los desarrolladores, y la prevalencia de `Git` facilita obtener asistencia cuando se requiere. Prácticamente todos los entornos de desarrollo son compatibles con `Git`, y las herramientas de línea de comandos de `Git` están implementadas en todos los sistemas operativos principales.

1.1. Principios fundamentales de Git

En `Git`, cada vez que se guarda el trabajo, se crea una confirmación. Esta representación es una instantánea de todos los archivos en un momento específico. Si un archivo no ha experimentado cambios entre una confirmación y la siguiente, Git utiliza la versión previamente almacenada. Este enfoque difiere de otros sistemas que almacenan una versión inicial de un archivo y registran las diferencias a lo largo del tiempo.

Las confirmaciones establecen conexiones con otras confirmaciones, dando forma a un gráfico que representa el historial de desarrollo. Es factible retroceder el código a una confirmación anterior, examinar cómo evolucionan los archivos de una confirmación a otra y revisar detalles como la ubicación y el momento en que se llevaron a cabo los cambios. En `Git`, las confirmaciones se identifican mediante un **hash** criptográfico único del contenido de la confirmación. Dado que todo está **hashado**, resulta imposible realizar modificaciones, perder información o dañar archivos sin que `Git` lo detecte.

1.2. Ramas

Cada desarrollador guarda los cambios en su propio repositorio de código local. Como resultado, puede haber muchos cambios diferentes basados en la misma confirmación. `Git` proporciona herramientas para aislar los cambios y volver a combinarlos posteriormente. Las ramas, que son punteros ligeros para el trabajo en curso, administran esta separación. Una vez finalizado el trabajo creado en una rama, se puede combinar de nuevo en la rama principal (o troncal) del equipo.

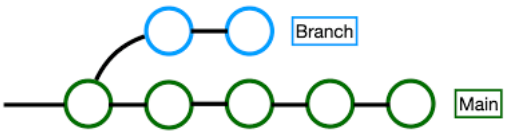

Figura 1: Diagrama de ramas.

1.3. Estados de los archivos y confirmaciones en Git

Los archivos en `Git` pueden encontrarse en uno de tres estados: modificados, almacenados provisionalmente o confirmados. Cuando se realiza la primera modificación en un archivo, esos cambios existen únicamente en el directorio de trabajo y aún no forman parte de ninguna confirmación ni del historial de desarrollo. El desarrollador debe almacenar provisionalmente los archivos modificados que se desean incluir en la próxima confirmación. El área de almacenamiento provisional contiene todos los cambios que se incorporarán en la siguiente confirmación. Una vez que el desarrollador está satisfecho con los archivos almacenados provisionalmente, estos se consolidan en una confirmación acompañada de un mensaje que describe las modificaciones realizadas. Esta confirmación se integra al historial de desarrollo.

La funcionalidad de almacenamiento provisional permite a los desarrolladores seleccionar qué cambios de archivos se incluirán en una confirmación, posibilitando dividir cambios extensos en una serie de confirmaciones más pequeñas. Al reducir el alcance de las confirmaciones, resulta más sencillo revisar el historial para identificar cambios específicos en los archivos.

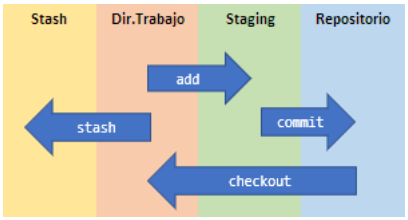

Figura 2: Estados de archivos.

1.4. Ventajas de Git

Git ofrece ventajas significativas en el desarrollo de software:

- **Desarrollo simultáneo**: Permite a todos los usuarios trabajar simultáneamente en sus propias copias locales de código, incluso sin conexión.
- **Versiones de lanzamiento más rápidas**: El uso de ramas facilita el desarrollo flexible y simultáneo, permitiendo la publicación de código estable de alta calidad de manera más rápida.
- **Integración incorporada**: Git se integra fácilmente con la mayoría de las herramientas y productos, simplificando el flujo de trabajo diario, incluyendo funciones como integración y despliegue continuos, pruebas automatizadas y seguimiento de elementos de trabajo.
- **Sólido soporte técnico de la comunidad**: Al ser de código abierto y un estándar de facto, Git cuenta con un amplio respaldo de la comunidad, facilitando el acceso a recursos y soporte técnico.
- **Compatible con cualquier equipo**: Utilizar Git con herramientas de administración de código fuente mejora la productividad del equipo al fomentar la colaboración, aplicar directivas y mejorar la visibilidad del trabajo.
- **Solicitudes de incorporación de cambios**: Facilita la revisión y discusión de cambios de código antes de la fusión con la rama principal, mejorando la calidad y el conocimiento del equipo.
- **Directivas de rama**: Permite configurar flujos de trabajo y procesos coherentes en todo el equipo, asegurando que las solicitudes de incorporación de cambios cumplan con los requisitos establecidos.

2. Instalación de Git en Windows[3]

- Accede al instalador más reciente de Git para Windows y descarga la última versión.
- Una vez que se haya iniciado el instalador, sigue las instrucciones proporcionadas en la pantalla del asistente de configuración de `Git` hasta que la instalación esté completa.
- Abre la ventana de comandos de Windows (cmd o `Git Bash` si seleccionaste no utilizar el `Prompt` de Comandos estándar de `Git` durante la instalación de `Git`).
- Escribe `git version` para verificar que `Git` se haya instalado correctamente.

3. Instalar Git para macOS[2]

La mayoría de las versiones de macOS ya incluyen Git instalado, y puedes activarlo a través del terminal con el comando GIT VERSION. Sin embargo, si por alguna razón no tienes Git instalado, puedes instalar la última versión siguiendo los siguientes pasos:

1. Accede al instalador más reciente de Git para macOS y descarga la última versión.
2. Una vez que se haya iniciado el instalador, sigue las instrucciones proporcionadas hasta que la instalación esté completa.
3. Abre el terminal y escribe `git version` para verificar que Git se haya instalado.

4. Instalar Git para Linux[2]

4.1. Debian/Ubuntu

1. Los paquetes de `Git` están disponibles mediante `apt`.
2. Es recomendable asegurarse de que estás utilizando la versión más reciente. Para hacerlo, navega a tu terminal de comandos y ejecuta el siguiente comando para garantizar que todo esté actualizado: `sudo apt-get update`.
3. Para instalar `Git`, ejecuta el siguiente comando: sudo `apt-get install git-all`.
4. Una vez que la salida del comando se haya completado, puedes verificar la instalación escribiendo: `git version`.

4.2. Fedora

1. Paquetes de Git están disponibles mediante `dnf`.
2. Para instalar Git, navega hasta tu terminal de comandos y ejecuta el siguiente comando: `sudo dnf install git-all`.
3. Una vez que la salida del comando se haya completado, puedes verificar la instalación escribiendo: `git version`.

Referencias bibliográficas

[1] Git. *Getting Started - What is Git?* URL: `https://git-scm.com/book/en/v2/Getting-Started-What-is-Git%3F`. (accessed: 08/02/2024).

[2] Git Guides. *Install Git.* URL: `https://github.com/git-guides/install-git#install-git-on-windows`. (accessed: 08/02/2024).

[3] Windows. *Instalación y configuración de Git.* URL: `https://learn.microsoft.com/es-es/devops/develop/git/install-and-set-up-git`. (accessed: 08/02/2024).