GitHub

Luis Mateo Hincapié Martínez

Arquitectura de software
Facultad de Ingeniería
Universidad de Antioquia
2020-1

En un alto nivel, GitHub es un sitio web y un servicio basado en la nube que ayuda a los desarrolladores a almacenar y administrar su código, así como a rastrear y controlar los cambios en su código. Para comprender exactamente qué es GitHub, necesita conocer dos conceptos relacionados: Control de versiones y Git

Control de versiones

El control de versiones ayuda a los desarrolladores a rastrear y administrar los cambios en el código de un proyecto de software. A medida que crece un proyecto de software, el control de versiones se vuelve esencial.

El control de versiones permite a los desarrolladores trabajar de forma segura mediante la ramificación y la fusión. Con la ramificación, un desarrollador duplica parte del código fuente (llamado repositorio). Luego, el desarrollador puede realizar cambios de manera segura en esa parte del código sin afectar el resto del proyecto.

Luego, una vez que el desarrollador consigue que su parte del código funcione correctamente, puede fusionar ese código con el código fuente principal para hacerlo oficial.

Luego, se realiza un seguimiento de todos estos cambios y se pueden revertir si es necesario.

Git

Git es un sistema de control de versiones de código abierto específico creado por Linus Torvalds en 2005.

Específicamente, Git es un sistema de control de versiones distribuido , lo que significa que todo el código base y el historial están disponibles en la computadora de cada desarrollador, lo que permite una fácil ramificación y fusión.

Según una encuesta de desarrolladores de Stack Overflow , más del 87% de los desarrolladores utilizan Git.

Palabras claves

Línea de comandos

El programa de computadora que usamos para ingresar comandos de Git. En una Mac, se llama Terminal. En una PC, es un programa no nativo que se descarga cuando descarga Git por primera vez (lo haremos en la siguiente sección). En ambos casos, escribe comandos basados en texto, conocidos como indicaciones, en la pantalla, en lugar de usar un mouse.

Repositorio

Un directorio o espacio de almacenamiento donde pueden vivir sus proyectos. A veces, los usuarios de GitHub lo acortan a "repositorio". Puede ser local a una carpeta en su computadora, o puede ser un espacio de almacenamiento en GitHub u otro host en línea. Puede mantener archivos de código, archivos de texto, archivos de imagen, lo que sea, dentro de un repositorio.

Control de versiones

Básicamente, el propósito para el que Git fue diseñado. Cuando tiene un archivo de Microsoft Word, sobrescribe cada archivo guardado con un nuevo guardado o guarda varias versiones. Con Git, no es necesario. Mantiene “instantáneas” de cada punto en el tiempo del historial del proyecto, por lo que nunca podrá perderlo ni sobreescribirlo.

Confirmar (Commit)

Este es el comando que le da a Git su poder. Cuando te comprometes, estás tomando una "instantánea" de tu repositorio en ese momento, lo que te da un punto de control al cual puedes reevaluar o restaurar tu proyecto a cualquier estado anterior.

Branch

¿Cómo trabajan varias personas en un proyecto al mismo tiempo sin que Git las confunda? Por lo general, se “bifurcan” del proyecto principal con sus propias versiones llenas de cambios que ellos mismos han realizado. Una vez que hayan terminado, es hora de "fusionar" esa rama con el "maestro", el directorio principal del proyecto.

Comandos específicos de Git

git init:

Inicializa un nuevo repositorio de Git. Hasta que ejecute este comando dentro de un repositorio o directorio, es solo una carpeta normal. Solo después de ingresar esto, acepta más comandos de Git.

git config:

Abreviatura de "configurar", esto es más útil cuando estás configurando Git por primera vez.

git help:

¿Olvidaste un comando? Escriba esto en la línea de comando para que aparezcan los 21 comandos de git más comunes. También puede ser más específico y escribir "*git help init"* u otro término para averiguar cómo usar y configurar un comando git específico.

git status:

Verifique el estado de su repositorio. Vea qué archivos están dentro, qué cambios aún deben confirmarse y en qué rama del repositorio está trabajando actualmente.

git add:

Esto no agrega archivos nuevos a su repositorio. En cambio, trae nuevos archivos a la atención de Git. Después de agregar archivos, se incluyen en las "instantáneas" del repositorio de Git.

git commit:

El comando más importante de Git. Después de realizar cualquier tipo de cambio, ingresa esto para tomar una "instantánea" del repositorio. Por lo general, va *git commit -m “Message here.”* El *-m* indica que la siguiente sección de la orden debe ser leído como un mensaje.

git branch:

¿Trabaja con varios colaboradores y desea realizar cambios por su cuenta? Este comando le permitirá construir una nueva rama, o línea de tiempo de confirmaciones, de cambios y adiciones

de archivos que son completamente suyos. Tu título va después del comando. Si quisiera una nueva rama llamada "gatos", escribiría git branch cats.

git checkout:

Literalmente le permite "verificar" un repositorio en el que no se encuentra actualmente. Este es un comando de navegación que le permite moverse al repositorio que desea verificar. Puede usar este comando git checkout master para ver la rama maestra o git checkout cart para ver otra rama.

git merge:

Cuando haya terminado de trabajar en una rama, puede fusionar sus cambios con la rama principal, que es visible para todos los colaboradores. *git merge cats* tomaría todos los cambios que hizo en la rama "gatos" y los agregaría al maestro.

git push:

Si está trabajando en su computadora local y desea que sus confirmaciones también sean visibles en línea en GitHub, “empuje” los cambios a GitHub con este comando.

git pull:

Si está trabajando en su computadora local y desea trabajar con la versión más actualizada de su repositorio, “extraiga” los cambios desde GitHub con este comando.

GitHub

GitHub es una empresa con fines de lucro que ofrece un servicio de alojamiento de repositorios Git basado en la nube. Esencialmente, hace que sea mucho más fácil para las personas y los equipos usar Git para el control de versiones y la colaboración.

La interfaz de GitHub es lo suficientemente fácil de usar para que incluso los programadores novatos puedan aprovechar Git. Sin GitHub, el uso de Git generalmente requiere un poco más de conocimientos técnicos y el uso de la línea de comandos.

Sin embargo, GitHub es tan fácil de usar que algunas personas incluso usan GitHub para administrar otros tipos de proyectos, como escribir libros .

Además, cualquiera puede registrarse y alojar un repositorio de código público de forma gratuita, lo que hace que GitHub sea especialmente popular entre los proyectos de código abierto.

Como empresa, GitHub gana dinero vendiendo repositorios de código privados alojados, así como otros planes centrados en el negocio que facilitan a las organizaciones la gestión de los miembros del equipo y la seguridad. Utilizamos Github ampliamente en Kinsta para administrar y desarrollar proyectos internos.

Repasemos algunas de las principales razones por las que a los geeks les gusta usar GitHub y aprendamos algo de terminología en el camino.

Repositorio

Un repositorio (generalmente abreviado como "repositorio") es una ubicación donde se almacenan todos los archivos de un proyecto en particular. Cada proyecto tiene su propio repositorio y puede acceder a él con una URL única.

Forking un repo

"Forking" es cuando creas un nuevo proyecto basado en otro proyecto que ya existe. Esta es una característica sorprendente que fomenta enormemente el desarrollo de programas y otros proyectos. Si encuentra un proyecto en GitHub en el que le gustaría contribuir, puede bifurcar el repositorio, realizar los cambios que desee y lanzar el proyecto revisado como un nuevo repositorio. Si el repositorio original que bifurcó para crear su nuevo proyecto se actualiza, puede agregar fácilmente esas actualizaciones a su bifurcación actual.

Solicitud de *Pull*

Ha hecho un *fork* a un repositorio, ha realizado una gran revisión del proyecto y desea que los desarrolladores originales lo reconozcan, tal vez incluso incluido en el proyecto / repositorio oficial. Puede hacerlo creando una solicitud de extracción. Los autores del repositorio original pueden ver su trabajo y luego elegir si lo aceptan o no en el proyecto oficial. Siempre que emite una solicitud de extracción, GitHub proporciona un medio perfecto para que usted y el responsable del mantenimiento del proyecto principal se comuniquen.

Redes Sociales

El aspecto de redes sociales de GitHub es probablemente su característica más poderosa, permitiendo que los proyectos crezcan más que cualquiera de las otras características ofrecidas. Cada usuario de GitHub tiene su propio perfil que actúa como una especie de currículum, mostrando su trabajo anterior y contribuciones a otros proyectos a través de solicitudes de extracción.

Registro de cambios

Cuando varias personas colaboran en un proyecto, es difícil realizar un seguimiento de las revisiones: quién cambió qué, cuándo y dónde se almacenan esos archivos. GitHub se encarga de este problema haciendo un seguimiento de todos los cambios que se han enviado al repositorio.

Referencias

What Is GitHub? A Beginner's Introduction to GitHub

What Is GitHub, and What Is It Used For?

The world's leading software development platform · GitHub

Git SCM

GitHub For Beginners: Don't Get Scared, Get Started