

Git

Git es un sistema de control de versiones distribuido de código abierto desarrollado por Linus Torvalds, el creador de Linux. El control de versiones se refiere al proceso de guardar diferentes archivos, o “versiones”, a lo largo de las diferentes etapas de un proyecto. Esto permite a los desarrolladores hacer un seguimiento de lo que se ha hecho y volver a una fase anterior si deciden revertir algunos de los cambios hechos.

La principal diferencia entre Git y cualquier otro sistema de control de versiones (VCS) es la forma en la que manejan sus datos. Conceptualmente, la mayoría de los VCS almacenan la información como una lista de cambios en los archivos. Estos sistemas manejan la información que almacenan como un conjunto de archivos y las modificaciones hechas a cada uno de ellos a través del tiempo.

GitHub Desktop

GitHub Desktop es, de hecho, una Interfaz Gráfica de Usuario (GUI, por sus siglas en inglés) diseñada para facilitar el uso de Git. Es posible utilizar GitHub Desktop para completar la mayoría de los comandos de Git desde una computadora de escritorio con confirmaciones visuales para los cambios. Permite subir, extraer y clonar repositorios remotos y utilizar herramientas colaborativas tales como atribuir confirmaciones y crear solicitudes de extracción. GitHub Desktop es un proyecto de código abierto, por lo tanto, es posible ver el itinerario, contribuir con el proyecto, o abrir un informe de problemas para proporcionar retroalimentación o solicitudes de características

Ventajas

Git

- Sistema distribuido, sin un punto central de fallo, que permite el trabajo incluso sin conexión.
- Rápido y ligero, optimizado para hacer operaciones de control muy rápidas.

- Crear ramas y mezclarlas es rápido y poco propenso a problemas, al contrario que en otros sistemas tradicionales.
- La integridad de la información está asegurada gracias a su modelo de almacenamiento, que permite predecir este tipo de problemas.
- Permite flujos de trabajo muy flexibles.
- El concepto de área de preparación o staging permite versionar los cambios de manera personalizada, no todo o nada.
- Es gratis y de código abierto.

GitHub Desktop

- La GUI le permite al usuario interactuar con el programa a través de un dispositivo visual que reemplaza la línea de comandos.
- Se reduce la curva de aprendizaje de Git.
- Búsqueda muy rápida en la estructura de los repositorios.
- Amplia comunidad y facilidad para encontrar ayuda.
- Ofrece herramientas de cooperación prácticas y una buena integración con Git.
- Fácil integración con otros servicios de terceros.
- Trabaja también con TFS, HG y SVN.
- Servicio gratuito.

Desventajas

Git

- Es más complejo que los sistemas centralizados tradicionales, ya que entran en juego más repositorios, más operaciones y más posibilidades para trabajar en equipo.
- La curva de aprendizaje es pronunciada y la documentación es tan compleja que muchas veces no resulta de ayuda.

- Los comandos y algunos conceptos que usa pueden llegar a ser confusos, al igual que algunos mensajes que muestra.
- Por defecto, se lleva mal con archivos binarios muy grandes, como vídeos o documentos gráficos muy pesados.

GitHub Desktop

- Algunos servicios son de pago.
- Tiene limitaciones de espacio, ya que no se puede exceder de 100MB en un solo archivo, mientras que los repositorios están limitados a 1GB en la versión gratis.

Archivos compatibles:

Git

- GITIGNORE: Git Ignore File.
- GITATTRIBUTES: Git Attributes File.
- GITKEEP: Git Keep File.
- SNAG: Snagit Capture File.
- SNAGPROJ: Snagit Project File.
- SNAGSTYLES: Snagit Style Archive.
- SNAGITSTAMPS: Snagit Stamp Archive.
- ZGT: Oren Scientific Word Processor.
- GITCONFIG: Git Configuration.
- GITEXT: Git Extension.
- GITMODULES: Git Module.
- NPMIGNORE: Git Data.
- SNAGACC: Snagit File.
- SNAGPROF: Snagit File.
- SNAGUNDO: Snagit Data.
- SSF: Snagit File.

GitHub Desktop

- PNG (.png)
- GIF (.gif)
- JPEG (.jpg)
- SVG (.svg)
- Archivos de registro (.log)
- Documentos de Microsoft Word (.docx), PowerPoint (.pptx) y Excel (.xlsx)
- Archivos de texto (.txt)
- PDF (.pdf)
- ZIP (.zip, .gz)
- Video (.mp4, .mov)

El tamaño máximo de archivo es:

- 10MB de imágenes y gifs.
- 10MB para videos que se suban a un repositorio que pertenezca a un usuario u organización en un plan gratuito de GitHub.
- 100MB para videos que se suban a los repositorios que pertenezcan a un usuario u organización de un plan de pago de GitHub.
- 25MB para el resto de los archivos.

Instalacion

Git

Instalar GIT en Windows

En Windows, sólo se tiene que descargar el instalador y ejecutarlo. Sigue estos sencillos pasos parahacerlo:

1. Descargar el instalador a través del siguiente enlace: <http://git-scm.com/download/win>.
2. Una vez que descargado el instalador, hacer doble clic sobre el ejecutable para que comience el proceso de instalación y siga las instrucciones que aparecerán en pantalla. Al igual que cualquier otro programa, se tendrá que dar “Next” (siguiente) en varias ocasiones hasta que aparezca la opción “Finish” (terminar)

para completar la instalación.

3. Ahora es necesario abrir el símbolo de sistema y escribir los siguientes comandos en la terminal:
 - a. `git config --global user.name "nombre".`
 - b. `git config --global user.email "ejemplo@email.com".`

Instalar GIT en MacOS

Es necesario seguir los siguientes paso:

1. Descargar el instalador a través del siguientes enlace: <http://git-scm.com/download/mac>.
2. Seguir las instrucciones que aparecerán en el programa de instalación.
3. Ahora ejecute los siguientes comandos en la terminal para configurar su correo y el nombre de usuario que están asociados a la cuenta GIT:
 - a. `git config --global user.name "nombre".`
 - b. `git config --global user.email "ejemplo@email.com".`

Instalar GIT en Linux (Ubuntu/Debian)

1. Abrir la terminal y ejecutar los siguientes comandos:
 - a. `sudo apt-get update`
 - b. `sudo apt-get install git`
2. Verificar que la instalación se haya logrado correctamente usando el comando: `git --version`.
3. A continuación, ejecutar los siguientes comandos en la terminal para poder configurar el correo y el nombre de usuario que están asociados a la cuenta GIT:
 - a. `git config --global user.name "nombre".`
 - b. `git config --global user.email "ejemplo@email.com".`

GitHub Desktop

1. Para instalar GitHub Desktop, visite la página de descargas:
<https://desktop.github.com/>.
2. Seguir las instrucciones que aparecerán en el programa de instalación.
3. Después de que haya instalado GitHub Desktop, puede autenticar la aplicación con su cuenta en GitHub o en GitHub Enterprise. Esta autenticación le permite conectarse remotamente a los repositorios en GitHub o en GitHub Enterprise.
 - a. En el menú desplegable de archivo, da clic en Opciones. En la ventana de opciones, da clic en Cuentas y sigue los pasos para iniciar sesión.

Compatibilidad de lenguajes

Los lenguajes centrales para las características de GitHub incluyen a C, C++, C#, Go, Java, JavaScript, PHP, Python, Ruby, Scala y TypeScript. Para las características que son compatibles con los administradores de paquetes, los administradores de paquete que son actualmente compatibles se incluyen en la tabla con sus lenguajes relevantes.

- Golang / Go
- Swift
- TypeScript
- Rust
- Kotlin
- Python
- PHP
- Julia
- Ruby
- Coffeescript
- Elixir
- Crystal
- PowerShell
- Scala
- Roslyn
- Clojure
- MicroPython
- Erlang
- Nim
- AssemblyScript
- PureScript
- Elm
- C#
- Red
- PonyLang

- Frege
- Goby
- Racket
- Idris
- OCaml
- Eta
- Nu
- LiveScript
- Io
- DMD
- Terra
- Skip
- Rakudo
- Chapel
- Lucee
- Golo
- Gosu
- Zig
- Haxe
- LiveCode
- Coq
- V
- Dart SDK
- Pharo
- Ring
- Senegal
- EoLang
- ChavaScript
- Jelly
- Beef
- Cue

- JDK
- TinyCC
- PC-Basic
- QB64
- Gleam
- Gwion