

Universidad de San Carlos de Guatemala

Facultad de Ingeniería

Escuela de Postgrado

Maestría en Ingeniería para la Industria con especialidad en Telecomunicaciones

Fundamentos de programación y Scripting

Erwin David Navarro Fuentes 999012461

¿Qué es GIT?

Git es un Sistema de Control de Versiones Distribuido (DVCS) utilizado para guardar diferentes versiones de un archivo (o conjunto de archivos) para que cualquier versión sea recuperable cuando lo desee.

Git también facilita el registro y comparación de diferentes versiones de un archivo. Esto significa que los detalles sobre qué cambió, quién cambió qué, o quién ha iniciado una propuesta, se pueden revisar en cualquier momento.

Sistema de Control de Versiones Distribuido

El término "distribuido" significa que cuando le instruyes a Git que comparta el directorio de un proyecto, Git no sólo comparte la última versión del archivo. En cambio, distribuye cada versión que ha registrado para ese proyecto. Este sistema "distribuido" tiene un marcado contraste con otros sistemas de control de versiones. Ellos sólo comparten cualquier versión individual que un usuario haya explícitamente extraído desde la base de datos central/local.

Un Sistema de Control de Versiones (VCS) se refiere al método utilizado para guardar las versiones de un archivo para referencia futura.

Estados de los archivos en Git

En Git hay tres etapas primarias (condiciones) en las cuales un archivo puede estar: estado modificado, estado preparado, o estado confirmado.

- Estado modificado

Un archivo en el estado modificado es un archivo revisado – pero no acometido (sin registrar).

En otras palabras, archivos en el estado modificado son archivos que has modificado pero no le has instruido explícitamente a Git que controle.

- Estado preparado

Archivos en la etapa preparado son archivos modificados que han sido seleccionados – en su estado (versión) actual – y están siendo preparados para ser guardados (acometidos) al repositorio .git durante la próxima instantánea de confirmación.

Una vez que el archivo está preparado implica que has explícitamente autorizado a Git que controle la versión de ese archivo.

- Estado confirmado

Archivos en el estado confirmado son archivos que se guardaron en el repositorio .git exitosamente.

Por lo tanto, un archivo confirmado es un archivo en el cual has registrado su versión preparada en el directorio (carpeta) Git.

El estado de un archivo determina la ubicación donde Git lo colocará.

GitHub es una plataforma basada en la web donde los usuarios pueden alojar repositorios Git. Facilita compartir y colaborar fácilmente en proyectos con cualquier persona en cualquier momento.

GitHub también fomenta una participación más amplia en proyectos Código Abierto al proporcionar una manera segura de editar archivos en repositorios de otros usuarios.

Para alojar (o compartir) un repositorio Git en GitHub, siga los siguientes pasos:

Paso 1: Registrar una cuenta de GitHub

El primer paso para comenzar con el alojamiento en GitHub es crear una cuenta personal. Visite la página de registro oficial para registrarse.

Paso 2: Crear un repositorio remoto en GitHub

Después de registrarse, crear un repo (un repositorio) en GitHub para el repositorio Git que quieres compartir.

Paso 3: Conectar el directorio Git del proyecto con el repositorio remoto

Una vez que hayas creado un repositorio remoto para tu proyecto, tienes que vincular el directorio .git del proyecto – ubicado localmente en tu sistema – con el repositorio remoto en GitHub.

Para conectar con el repositorio remoto, debes ubicarte en el directorio raíz del proyecto que quieras compartir, utilizando tu terminal local, y ejecuta este comando:

```
git remote add origin https://github.com/tuUsuario/tuRepositorio.git
```

Nota:

Reemplaza tuUsuario en el código de arriba con tu nombre de usuario de GitHub.

También reemplaza tuRepositorio con el nombre del repositorio remoto al que te quieres conectar.

El comando de arriba implica que git debe agregar al proyecto la URL especificada como una referencia remota con la cual el directorio local .git puede interactuar.

La opción origin en el comando de arriba es el nombre predeterminado (un nombre corto) que Git le otorga al servidor que aloja tu repositorio remoto.

Es decir, Git utiliza el nombre corto origin en vez de la URL del servidor.

No es obligatorio quedarse con el nombre predeterminado del servidor. Si prefieres otro nombre que origin, simplemente sustituya el nombre origin en el comando git remote add de arriba por cualquier nombre que prefieras.

Siempre recuerda que un nombre corto del servidor (por ejemplo, origin) no es nada especial! Sólo existe – localmente – para ayudarte a referenciar fácilmente la URL del servidor. Por lo tanto, siéntete libre de cambiarlo a un nombre corto que puedas referenciar fácilmente.

Para renombrar cualquier URL remota que exista, utilice el comando git remote rename de esta manera:

```
git remote rename nombreURLactual nuevoNombreURL
```

Cada vez que clones (descargues) cualquier repo remota, Git automáticamente le pone nombre origin a la URL de la repo. Sin embargo, le puedes especificar un nombre distinto con el comando git clone -o tuNombrePreferido.

Para ver la URL exacta guardada para los nombres como origin, ejecuta el comando git remote -v.

Paso 4: Confirmar la conexión

Una vez que hayas conectado tu directorio Git con el repositorio remoto, verifica si la conexión fue exitosa ejecutando el comando git remote -v.

Después verifica la impresión para confirmar que la URL mostrada sea la misma que la URL remota que intentas conectarte.

Sin embargo, si no estás seguro de la URL remota que vas a utilizar, echa un vistazo al artículo “¿Qué URL remota debería utilizar?”.

¿Quieres cambiar tu URL remota? Cambiar la URL de un remoto es una excelente guía.

Paso 5: Subir un repo Git local a un repo remoto

Después de conectar exitosamente tu directorio local con el repositorio remoto puedes comenzar a subir (cargar) tu proyecto local upstream.

Cuando estés listo para compartir tu proyecto en otra parte, en cualquier repo remoto, simplemente le dices a Git que suba todos tus confirmaciones, ramas y archivos en tu directorio .git local al repositorio remoto.

La sintaxis del código utilizado para subir (push) un directorio local Git a un repositorio remoto es `git push -u nombreRemoto nombreRama`.

Esto es, para subir tu directorio local .git y suponiendo que el nombre corto de la URL remota es "origin", ejecuta:

```
git push -u origin master
```

Nota:

El comando de arriba implica que git debe subir tu rama master local a la rama master remota ubicada en la URL con nombre origin.

Técnicamente puedes sustituir la opción origin con la URL del repositorio remoto. Recuerda, la opción origin es solo un nombre de la URL que hayas registrado en tu directorio .git local.

El indicador -u (indicador de referencia upstream/seguimiento) automáticamente vincula la rama local del directorio .git con la rama remota. Esto te permite usar git pull sin ningún argumento.

Paso 6: Confirmar la subida

Por último, vuelve a tu página de repositorio GitHub para confirmar que Git haya subido exitosamente tu directorio Git local al repositorio remoto.

GitHub también tiene un servicio gratuito opcional para convertir tu repositorio remoto en una página web funcional. Abajo vamos a ver cómo hacerlo

En resumen

GitHub es una plataforma en línea de alojamiento (o para compartir) repositorios de Git. Nos ayuda crear una avenida para colaborar fácilmente con cualquier persona, en cualquier lugar, en cualquier momento.

<https://github.com/ErwinNavarroGT/Homeworks-R>