

Jorge David Top Raxon

Fundamentos de programación y scripting

Maestría en Ingeniería para la Industria con especialidad en Ciencias de la Computación

04 de junio de 2021

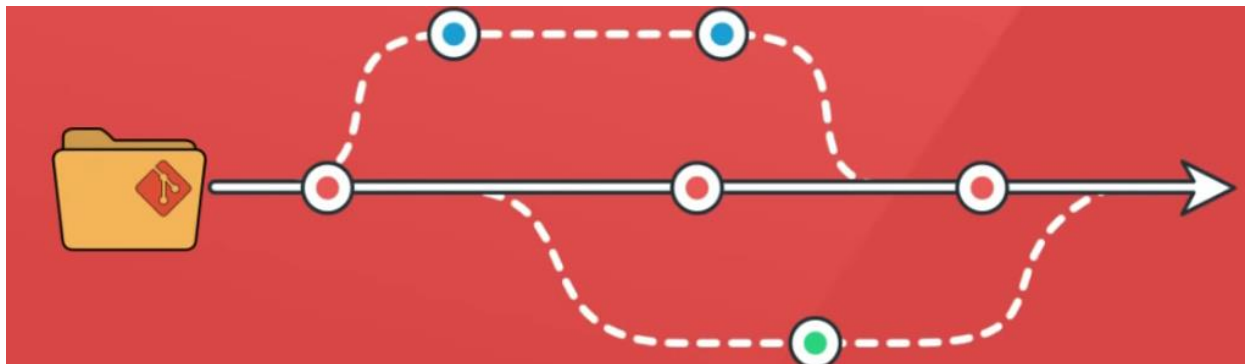
Tarea 1

Que es GIT

Git es un Sistema de control de versiones (VCS, por sus siglas en ingles) que ayuda a manejar y facilitar el control de versiones de software, al registrar cada cambio, configuración, actualización etc. Al estilo de un diario de una persona, es un registro de cada suceso que le ha pasado al proyecto a medida que se va desarrollando y lo mantiene para que en cualquier momento pueda ser consultado.

Terminología

Un repositorio es todo proyecto que está siendo registrado por un GIT, un Commit es cada cambio registrados en GIT enviado por cada uno de los desarrolladores, las ramas son los nuevos caminos o bifurcaciones que toma el proyecto donde se pueden hacer cambios a un proyecto sin afectar al proyecto principal. Un clon, es una copia exacta de un repositorio, un fork es un proyecto diferente pero basado en un proyecto principal. Fusiones o Merge, son los cambios aceptados que son finalmente integrados al repositorio principal.

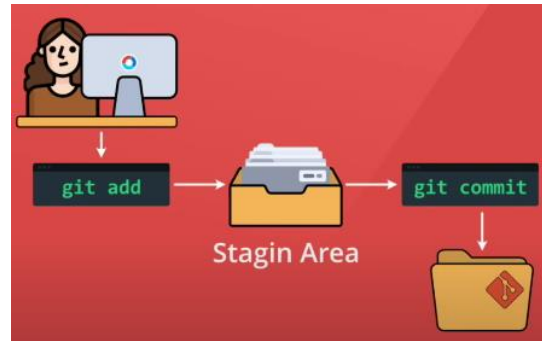


Control de versiones con GIT

El control de versiones con Git no es más que la gestión de los cambios sobre un producto a través de sus herramientas disponibles distribuidas que están disponibles para cada desarrollador independientemente y que pueden fusionarse posteriormente a la rama principal. Git ofrece además seguridad en términos de integridad al agregar un Checksum a cada archivo y commits para evitar corrupción en los datos.

Git permite llevar de manera ordenada y eficiente dicho control, principalmente cuando se trata de una gran cantidad de archivos de código. El flujo de trabajo se puede describir como:

- Inicialización o clonación de un repositorio para generar una copia en el equipo local.
- Mandar los cambios al repositorio a través de Commits.
- Los cambios son enviados al Staging Area, que es una etapa intermedia o de preparación.



Cuando hay varias personas en un equipo, el archivo master que es el principal y más protegido de las versiones, de ella sale el archivo “dev” de la cual se toman las ramas y a quien le afectan los cambios realizados en cada Commit. Una vez “dev” está de acuerdo a las necesidades u objetivos, se integra a “master” actualizándolo.

Estados de un archivo en GIT

Un archivo puede estar:

- Confirmado, es el archivo en el que ya se han finalizado los cambios y se convierte en una nueva revisión, una actualización.
- Modificado, es el archivo en el que hemos estado trabajando y se está preparando para enviar en una nueva revisión.
- Preparado, es el archivo o cambios que está listo para ir a una revisión.

Como se configura un repositorio

Para configurar un repositorio por primera vez se realizan los siguientes pasos:

- Archivo de configuración:
Utilizando *git config*, se selecciona donde se guardarán las variables de configuración que controlan los aspectos de cómo funciona y opera Git.
 - *[path]/etc/gitconfig* : contiene los valores de todos los usuarios y todos sus repositorios.
 - *~/.gitconfig* or *~/.config/git/config* : contiene los valores específicos y personales del usuario ejecutando.
 - *Config* : archivo de configuración en el directorio Git o en el repositorio actual.
- Identidad
Se ingresa el nombre y el correo electrónico

```
$ git config --global user.name "John Doe"
$ git config --global user.email johndoe@example.com
```
- Editor
Se configura el editor de texto que se utilizará para enviar un mensaje, en el ejemplo Emacs:

```
$ git config --global core.editor emacs
```

- Se designa el nombre de la rama master
`$ git config --global init.defaultBranch main`
- Configuración del repositorio y commit
 Se navega al directorio de la aplicación o pagina web por medio de lo siguiente, donde el username es el usuario Shell y example.com el sitio web.
`[server]$ cd /home/username/example.com`

Para inicializar un Nuevo repositorio se usa:

```
[server]$ git init
Initialized empty Git repository in /home/username/example.com/.git/
```

Agregar todos los archivos de la aplicación al repositorio:

```
[server]$ git add .
```

Verificar el estado de los archivos en el área de preparación:

```
[server]$ git status
```

Initial commit

Changes to be committed:
 (use "git rm --cached ..." to unstage)

```
new file: favicon.ico
new file: images/image1.jpg
new file: index.html
new file: mystyles.css
```

Se envian los cambios al repositorio de Git

```
[server]$ git commit -m "Committing website files for the first time"
[master (root-commit) e11cdb7] Committing website files for the first time
6 files changed, 34 insertions(+)
create mode 100644 favicon.ico
create mode 100644 images/image1.jpg
create mode 100644 images/image2.jpg
create mode 100644 images/image3.jpg
create mode 100644 index.html
create mode 100644 mystyles.css
```

Se verifican el historial de confirmaciones:

```
[server]$ git log --oneline
e11cdb7 Committing website files for the first time
```

Comandos en GIT

- Git init: inicia un repositorio vacío en una carpeta específica.
- Git clone: Genera una copia de un repositorio existente.
- Git add "nombre_archivo": añade un archivo específico al Staging Area.
- Git add. : añade todos los archivos del directorio al Staging Area.
- Git commit -am "mensaje": se confirman los cambios realizados, el mensaje es para realizar una breve descripción de los cambios realizados, esto envía las nuevas características o cambios al repositorio.
- Git revert "hash_commit": revertir el commit identificado en "hash_commit"
- Git push origin "nombre_rama": subir el branch (rama) al servidor remoto
- Git status: mostrar el estado actual del branch (rama) y los cambios que no han sido aceptados.

Bibliografía

- Que es GIT y para que sirve
Juan Carlos Rubio, 2019
<https://openwebinars.net/blog/que-es-git-y-para-que-sirve/>
- 1.6 Getting Started - First-Time Git Setup
<http://git-scm.com/book/en/v2/Getting-Started-First-Time-Git-Setup>
- ¿Cómo configurar un repositorio en Git?
<https://help.dreamhost.com/hc/es/articles/216445317--C%C3%B3mo-configurar-un-repositorio-Git->