Manejo de Git

Carlos Steven Cantor Montenegro

Kimberly Cataño

Luis Alejandro Gómez

Víctor Hugo Peláez

Diana Patricia Zúñiga

Tutor: Mauricio Perdomo Vargas

Grupo: 301122_32

Manejo de Git

Repasemos que es Git, para así entrar mas en detalle sobre como es su manejo.

Git es un sistema de control de versiones creado por Linus Torvalds el cual fue creado con la intención de mantener diferentes repositorios del Kernel de Linux.

Cada directorio de trabajo de Git es un repositorio independiente.

Ahora miraremos algunos de los comandos básicos que existen dentro de Git, los cuales son:

Clonar repositorios:

Para clonar un repositorio utilizamos el comando Git clone, seguido por la url del repositorio.

```
git clone <url repositorio>
```

Manejo de Git

Agregar archivos:

Para agregar los archivos que han sido modificados utilizamos, git add.

```
git add .
```

Commit

Para hacer Commit a las modificaciones realizadas

```
git commit -m "<mensaje del commit>"
```

Manejo de Git

Push:

Subir los cambios al repositorio

```
git push origin master
```

apunta a la dirección donde deseáis subir el nuevo material

```
git remote add origin <server>
```

Manejo de Ramas

Las ramas son utilizadas para desarrollar funcionalidades aisladas unas de otras. La rama *master* es la rama "por defecto" cuando creas un repositorio. Crea nuevas ramas durante el desarrollo y fusiónalas a la rama principal cuando termines.

Crear una rama:

Para crear una rama se usa el comando checkout –b y se pasa el nombre de la rama

```
git checkout -b feature_x
```

Volver a la rama original:

Para volver a la rama principal se usa el comando checkout master

```
git checkout master
```

Manejo de Ramas

Borrar una rama:

Para borrar una rama utilizamos el comando branch –d y pasamos el nombre de la rama

```
git branch -d feature_x
```

Subir contenido a una rama del repositorio:

Para subir el contenido utilizamos el comando Push origin y pasamos el nombre de la rama.

```
git push origin <branch>
```

Actualizar y Fusionar

Actualizar repositorio:

Para actualizar el repositorio local al ultimo Commit realizado se utilizado el comando pull.

```
git pull
```

Fusionar los cambios:

Para fusionar los cambios remotos se utiliza el comando merge y se especifica el nombre de la rama.

```
git merge <branch>
```

Revisar cambios:

Para revisar los cambios en los archivos se utiliza el comando diff y se especifican las dos ramas que se están comparando.

```
git diff <source_branch> <target_branch>
```

Etiquetas

Como obtener el Commit id:

Para obtener el id del Commit utilizamos el comando log.

```
git log
```

Crear una etiqueta:

Para crear una nueva etiqueta utilizamos el comando tag acompañándolo por el numero o valor de la etiqueta y el id del Commit al cual le va a asignar la etiqueta creada

```
git tag 1.0.0 1b2e1d63ff
```

Cambios Locales

Realizar cambio:

Para realizar un cambio utilizamos el comando checkout – y el nombre del archivo.

```
git checkout -- <filename>
```

Deshacer cambios:

Para deshacer los cambios locales se trae la ultima versión del servidor y apuntar a la copia local principal.

Para esto se utilizan los siguientes comandos:

```
git fetch origin
```

```
git reset --hard origin/master
```

Otros Comandos

Git reset:

Se utiliza para resetear el índex y el directorio en el que se esta trabajando al ultimo estado.

```
git reset - -hard HEAD
```

Git rm:

Se utiliza para remover archivos.

```
git rm filename.txt
```

Otros Comandos

Git stash:

Se utiliza para salvar cambios que no están por ser comprometidos.

```
git stash
```

Git show:

Se utiliza para mostrar información sobre cualquier objeto git.

```
git show
```

Otros Comandos

Git fetch:

Se utiliza para buscar todos los objetos de un repositorio remoto que actualmente no esta en el directorio local.

```
git fetch origin
```

Git ls - tree:

Se utiliza para ver un objeto de árbol junto con el nombre y modo de cada uno de ellos, y el valor blobs SHA-1.

Otros Comandos

Git grep:

Se utiliza para buscar en los árboles de contenido cualquier frase o palabra. Por ejemplo, para buscar por www.tupaginaweb.com en todos los archivos se usaría:

```
git grep “www.tupaginaweb.com”
```

Git gc:

Se utiliza para optimizar el repositorio por medio de una recolección de basura, que limpiara archivos innecesarios y los optimizara.

Otros Comandos

Git archive:
Este comando le permite al usuario crear archivos zip o tar que contengan los constituyentes de un solo árbol de repositorio

Bibliografía

Comandos git tomado de:

https://www.hostinger.es/tutoriales/comandos-de-git

Comandos git tomado de:

http://rogerdudler.github.io/git-guide/index.es.html

Comandos git tomado de:

https://cursos.virtual.uniandes.edu.co/isis1206/manejo-de-repositorios-git/