**Montoya Cárdenas Andrea**

Comandos GIT

**git help comando**  
 Muestra ayuda del comando especificado

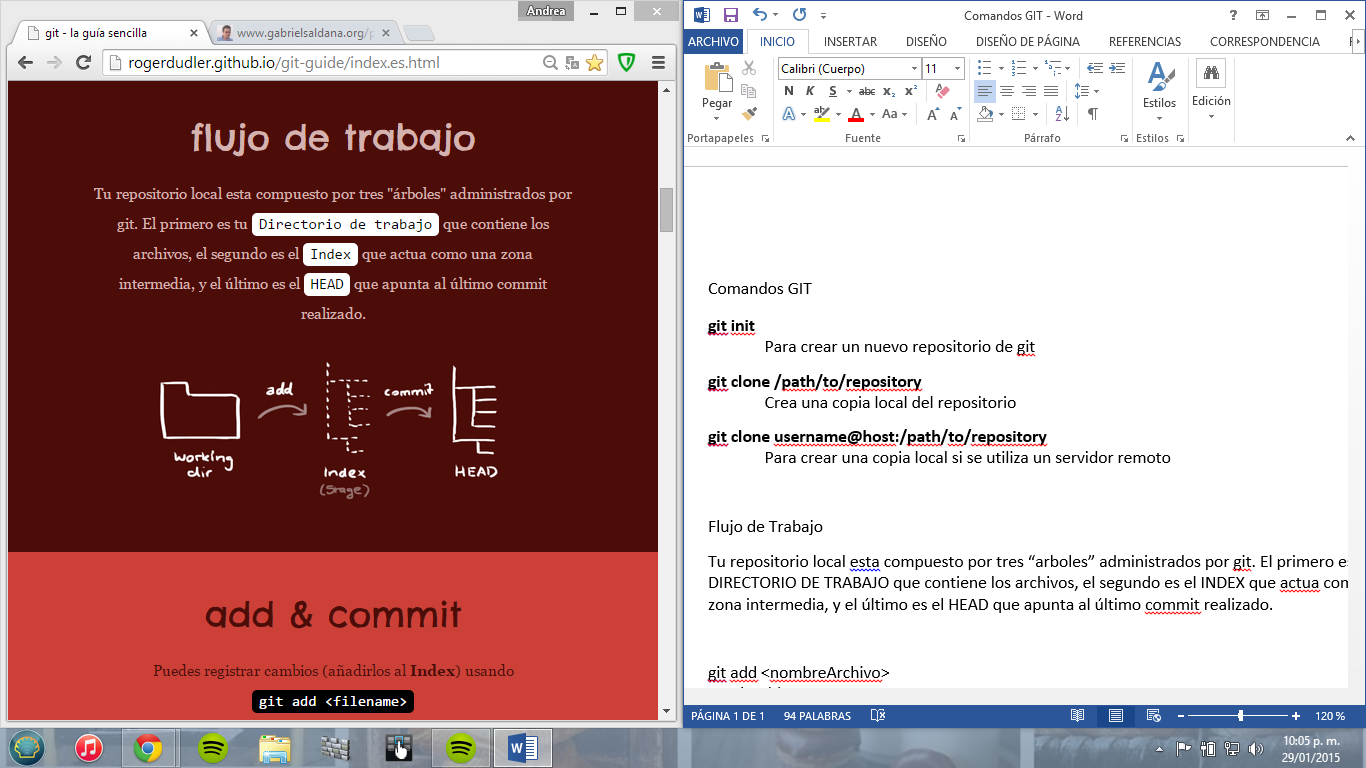
**git init**  
 Para crear un nuevo repositorio de git

**git clone /path/to/repository**  
 Crea una copia local del repositorio

**git clone username@host:/path/to/repository**  
 Para crear una copia local si se utiliza un servidor remoto

**Flujo de Trabajo**

Tu repositorio local está compuesto por tres “arboles” administrados por git. El primero es tu DIRECTORIO DE TRABAJO que contiene los archivos, el segundo es el INDEX que actúa como una zona intermedia, y el último es el HEAD que apunta al último commit realizado.



**git add <nombreArchivo>  
 git add .**  
 Puedes registrar cambios (añadirlos al index).   
 Es el primer paso en el flujo de trabajo básico.

**git commit –m “Commit message”**  
 Para hacer commit a los cambios que se hicieron (add). De esta manera el archivo está incluido en el HEAD, pero aún no en el repositorio central.

Parámetros

-a   
 Organiza automáticamente los archivos que ha sido modificados y eliminados, pero los nuevos archivos que no le has dicho a Git no se ven afectados.

-m   
 Dentro de las comillas “” va el mensaje de confirmación.

**git commit –a –m “mensaje”**  
 No necesita el git add previo

**git push origin master**  
 Envia los cambios a tu repositorio, (origin y master pueden ser opcionales), se puede reemplazar master por la rama a la que quieres enviar tus archivos.

**git remote add origin <server>**  
 Si no has clonado un repositorio ya existente y quieres conectar tu repositorio local a un repositorio remoto. De esta manera podrás subir tus cambios al repositorio remoto seleccionado.

**git pull**  
 Actualizar tu repositorio local al commit más nuevo

**Ramas**

Las ramas son utilizadas para desarrollar funcionalidades aisladas unas de otras. La rama MASTER es la rama “por defecto” cuando creas un repositorio.

**git branch <nombre\_rama>**  
 Crea una nueva rama llamada nombre\_rama

**git push origin <rama>**  
 Una rama nueva no estará disponible para los demás a menos que subas la rama a tu repositorio remoto

**git checkout <nombre\_rama>**  
 Vuelves a la rama indicada

**git branch –d <nombre\_rama>**  
 Borras la rama nombre\_rama

**git checkout –b <nombre\_rama>**  
 Crea y cambia de rama

**git merge <rama>**  
 Mezcla la rama actual con <rama>

**git push origin - -delete <nombre\_rama>**  
 Elimina una rama del servidor

**git branch**  
 Lista ramas

**git branch –v**  
 Lista ramas mostrando ultimo commit

**git branch - -merged**  
 Lista ramas que han sido mezcladas con la actual. Si no tienen un \*, pueden borrarse, ya que significa que se han incorporado los cambios en la rama actual

**git branch - -no-merged**  
 Lista ramas que no han sido incorporadas a la actual

No siempre será posible fusionar automáticamente los cambios y se podrá producir CONFLICTOS. Se tiene que fusionar estos conflictos manualmente al editar los archivos mostrados por git. Después de modificarlos, necesitas marcarlos como fusionados con (git add <nombreArchivo>)

Antes de fusionar los cambios puedes revisarlos  
**git diff <fuente\_rama> <rama\_objetivo>**

**git log**  
 Revisar historial

**git rm <ruta>**  
 Remueve un archivo o directorio del árbol de trabajo

Parámetro

-f  
 Fuerza la eliminación de un archivo del repositorio

**git mv <ruta\_origen> <ruta\_destino>**  
 Mueve el archivo o directorio a una nueva ruta

Parámetro

-f  
 Sobrescribe los archivo existentes en la ruta destino

**git diff <ruta>**  
 Muestra la diferencia entre los cambios en el árbol de trabajo local

**git diff HEAD <ruta>**  
 Muestra las diferencias entre los cambios registrados y los no registrados

**git resest HEAD <ruta>**  
 Marca el archivo para que no sea incluido en el próximo commit

**git reset - -soft HEAD^**  
 Deshace commit y conserva los cambios en el árbol de trabajo local

**git reset - -hard HEAD^**  
 Restablece el árbol de trabajo local a la versión del último commit

**git clean**  
 Elimina archivos desconocidos del árbol de trabajo local

**git blame <ruta>**  
 Muestra el archivo relacionado con las modificaciones realizadas

**git fetch <repositorio\_remoto>**  
 Trae los cambios desde un repositorio remoto

**git remote**  
 Lista los repositorios remotos

**git remote add <remote\_url>**  
 Añade un repositorio remoto a la lista de repositorios registrados

**git tag *name <revisión>*** Crea una etiqueta para la revisión referida

Parámetros

-s  
 Firma la etiqueta con su llave privada

-l *[patrón]* Imprime etiquetas y opcionalmente los registros que concuerden con el patrón de búsqueda