

Manejo básico de Git I

Resumen de Contenidos

Instalación

- <https://git-scm.com/download>

Configuración

Opciones obligatorias (nombre y correo)

```
git config --global user.name "Nombre y apellido"
git config --global user.email CORREO@ELECTRONICO
```

Editor de preferencia.

Como ejemplos se incluyen el Notepad y el Notepad ++ en Windows

Ejecutar sólo una de los tres comandos siguientes

```
git config --global core.editor notepad # Notepad de Windows
```

```
git config --global core.editor "'C:/Program Files/Notepad++/notepad++.exe' -multiInst -notabbar -nosession -noPlugin" # Notepad ++
```

```
git config --global core.editor "'C:/Program Files (x86)/Notepad++/notepad++.exe' -multiInst -notabbar -nosession -noPlugin" # Notepad ++ 32 bit (Nota: a mi no me funciona!)
```

Almacenamiento de credenciales para no pedir usuario y contraseña de GitHub

cada vez que se suban cambios al servidor

Ejecutar sólo uno de los dos comandos siguientes en función del sistema

```
git config --global credential.helper cache # Para Linux
```

```
git config --global credential.helper wincred # Para Windows
```

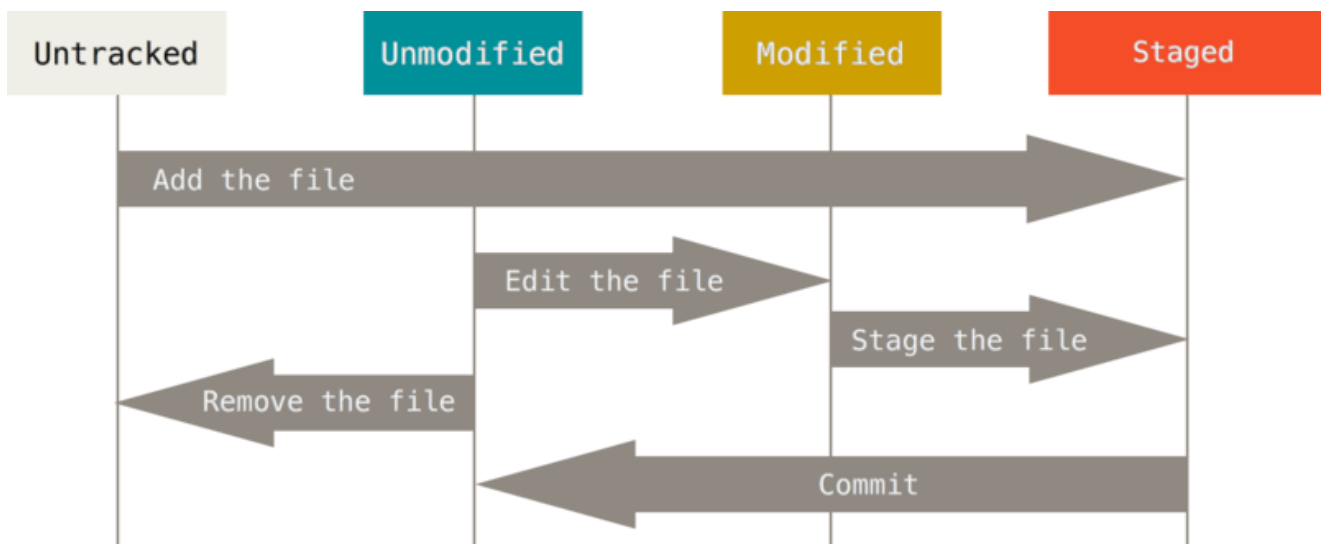
#Consultar la configuración

```
git config --list
```

Creación de repositorios

```
git init
```

Ciclo de vida



Revisando el estado

```
git status
```

Ignorar archivos

- Archivo .gitignore
- Plantillas de archivos [.gitignore](#).

Visualizar cambios

```
git diff  
git diff <archivo>
```

Añadir archivos al área de preparación (stage)

```
git add <archivo> # Añadir archivos individuales  
git add . # Añadir todos los archivos nuevos o modificados
```

Visualizar cambios de los archivos en el área de preparación

```
git diff --staged  
git diff --staged <archivo>
```

Confirmar cambios (commit)

```
git commit -m "MENSAJE"
```

Historial de cambios

```
git log  
git log --graph
```

Ver cambios realizados en anteriores commits

```
git show <commit>
```

Quitar archivo del área de preparación

```
git reset HEAD <archivo>
```

Eliminar las modificaciones con respecto al último commit

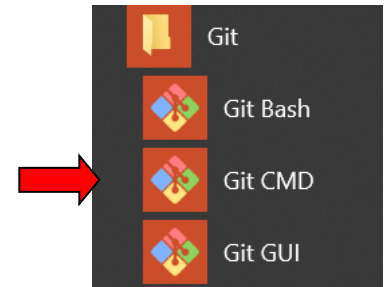
```
# ¡PELIGRO! Todos los cambios que se hayan hecho al archivo desde el último commit  
# se eliminarán  
git checkout -- <archivo>
```

Etiquetado

```
git tag
```

Manejo básico de Git. Práctica I

Tareas



1. Instala Git en tu sistema operativo.
2. Ejecuta Git CMD
3. Adjunta una captura de pantalla en la que aparezca el resultado de la ejecución del comando `git --version`.
4. Realiza tu configuración de Git (nombre, correo electrónico, editor de preferencia y almacenamiento de credenciales). Adjunta una captura de pantalla con el resultado de la ejecución de los comandos de configuración.
5. Crea una carpeta denominada S1R1. Realiza las siguientes acciones en ella:
 1. Crea un repositorio Git.
 2. Crea un fichero denominado `libros.txt`. Añade tres títulos de libros cada uno en una línea distinta.
 3. Haz un primer *commit*.
 4. Añade dos libros al archivo `libros.txt`.
 5. Haz un segundo *commit*.
 6. Crea un fichero denominado `peliculas.txt`. Añade tres títulos de películas a dicho archivo.
 7. Haz una captura de pantalla del comando `git status`.
 8. Crea un fichero denominado `comidas.txt`. Añade tres nombres de comidas a dicho archivo.
 9. Haz un tercer *commit* que incluya los archivos `peliculas.txt` y `comidas.txt`.
 10. Elimina el archivo `comidas.txt` desde el navegador de archivos.
 11. Añade dos películas más al archivo `peliculas.txt`.
 12. Haz una captura de pantalla que muestre los cambios en el directorio de trabajo.
 13. Añade los cambios al área de preparación.
 14. Haz una captura de pantalla del comando `git status`. Debe indicar que se ha borrado el archivo `comidas.txt` y que se ha modificado el archivo `peliculas.txt`.
 15. Haz un cuarto *commit*.
 16. Crea un archivo denominado `datos.bak`. Añade tres títulos de libros a dicho archivo.
 17. Crea una subcarpeta denominada `output`. Crea un archivo denominado `salida.txt` en su interior. Escribe tu nombre y apellidos en dicho archivo.
 18. Haz una captura de pantalla del comando `git status`. Deben aparecer los archivos `datos.bak` y `output/salida.txt` como archivos nuevos.
 19. Crea un archivo `.gitignore` para que los ficheros con extensión `.bak` y el contenido de la carpeta `output/` no se incluyan en el repositorio.
 20. Haz una nueva captura de pantalla del comando `git status`. Ahora no deben aparecer los archivos `datos.bak` y `output/salida.txt` como archivos nuevos, sino que en su lugar debe aparecer únicamente el archivo `.gitignore`.
 21. Haz un último *commit* para incluir el archivo `.gitignore` en el repositorio.
 22. Haz una captura de pantalla que muestre el histórico de cambios del repositorio.

Manejo básico de Git II

Resumen de Contenidos

Ramas

Definición de ramas

- Una rama es un puntero que apunta a un determinado commit.
- Un repositorio debe tener una rama como mínimo.
- El nombre de la rama que se crea por defecto es master. Este nombre no es especial ni tiene una función o significado especial.
- Existe un puntero especial llamado HEAD que apunta a la rama en la que estamos en ese momento.
- Al cambiar de rama se modifica el contenido del directorio de trabajo: éste se muestra tal como estaba en la rama a la que hemos saltado.
- La creación y el cambio de ramas se realizan de forma instantánea: no tienen apenas coste.
- El trabajo con ramas es muy interesante por los siguientes motivos:
 - Se pueden hacer pruebas sin modificar el código en producción.
 - Se puede separar el trabajo en tareas o subproyectos que no afecten unos a otros.
 - Cada miembro del equipo puede trabajar sin ser interferido por los demás.

Crear ramas

```
git branch <nombre_rama>
```

Ver ramas disponibles

```
git branch
```

Cambiar de rama

```
git checkout <nombre_rama>
```

Fusionar una rama

- Primero nos posicionamos en la rama sobre la que se va a realizar la fusión
- Para realizar la fusión ejecutar:

```
git merge <nombre_rama_a_fusionar>
```

Eliminar una rama

```
git branch -d <nombre_rama>
```

Remotos

Clonar un repositorio

```
git clone <URL_REPOSITORIO>
```

Ver remotos

```
git remote -v
```

Añadir, eliminar y renombrar remotos

```
git remote add <NOMBRE_REMOTO> <URL_REPOSITORIO> # Añadir remoto
```

```
git remote rm <NOMBRE_REMOTO> # Eliminar remoto
```

```
git remote rename <NOMBRE_ORIGINAL> <NOMBRE_NUEVO> # Renombrar remoto
```

Traer información del remoto

Este comando NO realiza la fusión en la rama local
 # Si se desean incorporar los cambios habría que realizar un git merge
 git fetch [NOMBRE_REMOTO] # El nombre del remoto por defecto es ORIGIN

Traer y fusionar cambios del remoto

git pull [NOMBRE_REMOTO] [NOMBRE_RAMA] # Pull = fetch + merge

Enviar cambios al remoto

git push [NOMBRE_REMOTO] [NOMBRE_RAMA]

Enviar los cambios de una rama al remoto y crear una rama remota asociada

git push -u NOMBRE_REMOTO NOMBRE_RAMA

Manejo básico de Git. Práctica II

Tareas

1. Crea una carpeta denominada S2R1. Realiza las siguientes acciones en ella:
 1. Crea un repositorio Git.
 2. Crea un fichero denominado actores.txt. Añade tres nombres de actores cada uno en una línea distinta.
 3. Haz un primer *commit*.
 4. Crea una rama denominada *test*.
 5. En la rama *test* crea un fichero denominado actrices.txt. Añade tres nombres de actrices y realiza un *commit* en dicha rama.
 6. Haz una captura de pantalla del resultado del comando git log --graph --all.
 7. Incorpora los cambios de la rama *test* a la rama *master*. Haz una captura de pantalla de los comandos que has utilizado y de su resultado.
 8. Posiciónate en la rama *master*.
 9. Crea una segunda rama denominada *test2*.
 10. En la rama *master*, añade una actriz al fichero actrices.txt.
 11. En la rama *test2*, añade una actriz al fichero actrices.txt.
 12. Haz una captura de pantalla del resultado del comando git log --graph --all. Debe haber dos caminos distintos: uno para la rama *master* y otro para la rama *test2*.
 13. Incorpora los cambios de la rama *test2* a la rama *master*. ¿Se produce un conflicto? De ser así realiza una captura del comando git status.
 14. Resuelve el conflicto incorporando los dos nombres de actrices.
 15. Haz una captura de pantalla del resultado del comando git log --graph --all. Observa que se ha creado un nuevo *commit* que integra los dos caminos anteriores..
2. Crea una carpeta denominada S2R2-remoto. Inicializa un repositorio Git en su interior mediante el comando git init --bare. Esta carpeta se utilizará como repositorio remoto.

3. Clona el repositorio S2R2-remoto en una carpeta denominada S2R2. Adjunta captura de pantalla del resultado del comando de clonado. A continuación realiza las siguientes acciones en el repositorio S2R2:
 1. Crea un archivo denominado `directores.txt`. Añade el nombre de tres directores de cine.
 2. Haz un *commit*.
 3. Realiza un *push* al repositorio remoto. Adjunta captura de pantalla del resultado.
 4. Crea una rama denominada *version1*.
 5. En la rama *version1* añade el nombre de dos directores de cine más al archivo `directores.txt`.
 6. Realiza un *push* de la rama al repositorio remoto de manera que quede asociada a la rama remota del mismo nombre. Adjunta captura de pantalla del resultado.

4. Clona el repositorio S2R2-remoto en una segunda carpeta denominada S2R3. Realiza las siguientes acciones sobre ella:
 1. Muestra en la consola el contenido del fichero `directores.txt` y el resultado del comando `git status`.
 2. Cambia a la rama *version1*. Muestra el resultado del comando.
 3. Muestra el contenido del fichero `directores.txt` por la pantalla. Comprueba que se muestran los 5 nombres de directores esperados. Adjunta captura de pantalla.
 4. Incorpora los cambios de la rama *version1* a la rama *master*.
 5. Sube la rama *master* actualizada al servidor. Adjunta captura de pantalla del resultado del comando.

5. Vuelve de nuevo a la carpeta S2R2 y realiza las siguientes acciones:
 1. Obtén los cambios que hay en el repositorio remoto. Adjunta captura de pantalla del resultado del comando utilizado.
 2. Actualiza la rama *master* local con el contenido de la rama *master* del repositorio remoto. Adjunta captura de pantalla del resultado del comando utilizado.
 3. Comprueba que deben aparecer los 5 nombres de directores esperados.