

Git – GitHub Cheat Sheet

Git instalación

| Acción | Comando |
|----------------|------------------|
| Instalar Git | Ir a Git SCM |
| Ver la versión | \$ git --version |

Repositorio

| Acción | Comando |
|--|----------------------------------|
| Crear un repositorio* | \$ git init |
| Ver la situación actual de los archivos | \$ git status |
| Añadir un archivo al repositorio | \$ git add file_name.ext |
| Añadir todos los archivos que contiene la carpeta | \$ git add . \$ git add -A |
| Mostrar las diferencias de archivos que no se han enviado aún al staging area | \$ git diff |
| Borrar el caché de un archivo** | \$ git rm --cached file_name.ext |
| * Primero se abre en la terminal la carpeta que contiene el proyecto del que se quiere inicializar un repositorio: \$ cd folder_with_project | |
| **Esto se hace cuando se quiere agregar un archivo al .gitignore (archivo que permite guardar archivos que no se quiere que sean públicos en el repositorio) | |

Repositorio remoto (GitHub)

| Acción | Comando |
|---|--|
| Conectar un repositorio local con un repositorio remoto (en GitHub) * | \$ git remote add origin repository_url.git |
| Listar los repositorios remotos a los que se está conectado | \$ git remote -v |
| Cambiar la url del repositorio remoto al que se está conectado | \$ git remote set-url origin new_repository_url |
| Actualizar el repositorio local para que tenga los últimos cambios del repositorio remoto | \$ git pull origin master \$ git pull origin branch_name |
| Subir los últimos cambios del repositorio local al repositorio remoto | \$ git push -u origin master \$ git push origin branch_name |
| Crear una copia al repositorio local, clonando un repositorio remoto | \$ git clone repository_url |
| * Se hace una sola vez para conectar los repositorios. | |

Configuraciones

| Acción | Comando |
|---|--|
| Configurar el nombre de usuario para todo el sistema | \$ git config --global user.name "my_username" |
| Configurar el nombre de usuario solo para el repositorio actual | \$ git config user.name "my_username" |
| Configurar el email para todo el sistema | \$ git config --global user.email "my_email" |
| Configurar el email solo para el repositorio actual | \$ git config user.email "my_email" |

| | |
|--|--------------------------------------|
| Habilitar la colorización en la línea de comando | \$ git config --global color.ui auto |
| Eliminar todas las configuraciones globales realizadas | \$ git config --global --unset-all |

Pasos generales para un proyecto

| | |
|---|---|
| 1) Abrir en la terminal la carpeta que contiene el proyecto. | \$ cd folder_with_project |
| 2) Crear un repositorio local en la carpeta del proyecto.* | \$ git init |
| 3) Añadir al repositorio local los archivos que se encuentran en la carpeta del proyecto. | \$ git add . \$ git add file_name.ext |
| 4) Realizar el commit de los archivos que están en staging area. | \$ git commit -m "Commit message" |
| Si se quiere subir a un repositorio remoto (Github): | |
| 5) Conectar el repositorio local con un repositorio remoto (en GitHub)** | \$ git remote add origin repository_url.git |
| 6) Subir los últimos cambios del repositorio local al repositorio remoto | \$ git push origin branch_name |
| * Se crea solo una vez. ** Se conecta solo una vez. | |

Commit

| Acción | Comando |
|---|--|
| Realizar el commit de los archivos que están en staging area | \$ git commit -m "Commit message" |
| Mostrar todos los commits realizados | \$ git log \$ git log --oneline |
| Revertir el commit a uno anterior* | \$ git revert commit_ID |
| Deshacer todos los commits que se hicieron después del commit_ID, preservando los cambios localmente* | \$ git reset commit_ID |
| Enumerar el historial de versión para un archivo, incluidos los cambios de nombre | \$ git log --follow file_name |
| Cambiar el nombre de un archivo | \$ git mv original_file_name new_file_name |
| *Para revertir o resetear el commit, se necesita el commit_ID del commit al cual se quiere volver, el cual lo encontramos cuando buscamos los commits realizados con \$ git log | |

Ramas

| Acción | Comando |
|--|---------------------------------------|
| Crear una nueva rama (branch) | \$ git branch new_branch_name |
| Ver todas las ramas del repositorio | \$ git branch |
| Cambiar de rama a donde se quiere subir el archivo | \$ git checkout branch_name |
| Crear una nueva rama y "cambiarse" a ella | \$ git checkout -b new_branch_name |
| Volver a la rama master | \$ git checkout master |
| Unir una rama con la rama activa actualmente | \$ git merge merging_branch_name |
| Borrar una rama | \$ git branch -d deleting_branch_name |
| Forzar el borrado de una rama (force-delete) | \$ git branch -D deleting_branch_name |