Comandos Git

- **add**

Agrega el contenido de fichero al índice.

git add [-n] [-v] [--force | -f] [--interactive | -i] [--patch | -p]

[--edit | -e] [--[no-]all | --[no-]ignore-removal | [--update | -u]]

[--intent-to-add | -N] [--refresh] [--ignore-errors] [--ignore-missing]

[--] [<pathspec>…]

- **am**

Aplica una serie de parches a partir de un buzón de correos.

git am [--signoff] [--keep] [--[no-]keep-cr] [--[no-]utf8]

[--3way] [--interactive] [--committer-date-is-author-date]

[--ignore-date] [--ignore-space-change | --ignore-whitespace]

[--whitespace=<option>] [-C<n>] [-p<n>] [--directory=<dir>]

[--exclude=<path>] [--include=<path>] [--reject] [-q | --quiet]

[--[no-]scissors] [-S[<keyid>]] [--patch-format=<format>]

[(<mbox> | <Maildir>)…]

git am (--continue | --skip | --abort)

- **annotate**

Anota líneas de fichero con información del “commit”.

git annotate [options] file [revision]

- **apply**

Aplica un parche a ficheros y/o al índice.

git apply [--stat] [--numstat] [--summary] [--check] [--index] [--3way]

[--apply] [--no-add] [--build-fake-ancestor=<file>] [-R | --reverse]

[--allow-binary-replacement | --binary] [--reject] [-z]

[-p<n>] [-C<n>] [--inaccurate-eof] [--recount] [--cached]

[--ignore-space-change | --ignore-whitespace]

[--whitespace=(nowarn|warn|fix|error|error-all)]

[--exclude=<path>] [--include=<path>] [--directory=<root>]

[--verbose] [<patch>…]

- **archive**

Crea un archivo de ficheros a partir de un árbol nombrado.

git archive [--format=<fmt>] [--list] [--prefix=<prefix>/] [<extra>]

[-o <file> | --output=<file>] [--worktree-attributes]

[--remote=<repo> [--exec=<git-upload-archive>]] <tree-ish>

[<path>…]

- **bisect**

Encuentra por medio de búsqueda binaria el cambio que introdujo un “bug”.

git bisect <subcommand> <options>

- **blame**

Muestra qué revisión y autor modificó por última vez cada línea de un archivo.

git blame [-c] [-b] [-l] [--root] [-t] [-f] [-n] [-s] [-e] [-p] [-w] [--incremental]

[-L <range>] [-S <revs-file>] [-M] [-C] [-C] [-C] [--since=<date>]

[--abbrev=<n>] [<rev> | --contents <file> | --reverse <rev>] [--] <file>

- **branch**

Enlista, crea, o elimina ramas.

git branch [--color[=<when>] | --no-color] [-r | -a]

[--list] [-v [--abbrev=<length> | --no-abbrev]]

[--column[=<options>] | --no-column]

[(--merged | --no-merged | --contains) [<commit>]] [<pattern>…]

git branch [--set-upstream | --track | --no-track] [-l] [-f] <branchname> [<start-point>]

git branch (--set-upstream-to=<upstream> | -u <upstream>) [<branchname>]

git branch --unset-upstream [<branchname>]

git branch (-m | -M) [<oldbranch>] <newbranch>

git branch (-d | -D) [-r] <branchname>…

git branch --edit-description [<branchname>]

- **bundle**

Mueve objetos y referencias por fichero.

git bundle create <file> <git-rev-list-args>

git bundle verify <file>

git bundle list-heads <file> [<refname>…]

git bundle unbundle <file> [<refname>…]

- **cat-file**

Provee información de contenido o de tipo y tamaño para objetos de repositorio.

git cat-file (-t | -s | -e | -p | <type> | --textconv) <object>

git cat-file (--batch | --batch-check) < <list-of-objects>

- **check-attr**

Muestra información de “gitattributes”.

git check-attr [-a | --all | attr…] [--] pathname…

git check-attr --stdin [-z] [-a | --all | attr…] < <list-of-paths>

- **check-ignore**

Depura el “gitignore” / excluye ficheros.

git check-ignore [options] pathname…

git check-ignore [options] --stdin < <list-of-paths>

- **check-mailmap**

Muestra nombres canónicos y direcciones de email de contactos.

git check-mailmap [options] <contact>…

- **check-ref-format**

Asegura que un nombre de referencia está bien formado.

git check-ref-format [--normalize]

[--[no-]allow-onelevel] [--refspec-pattern]

<refname>

git check-ref-format --branch <branchname-shorthand>

- **checkout**

Envía una rama o rutas al árbol de trabajo.

git checkout [-q] [-f] [-m] [<branch>]

git checkout [-q] [-f] [-m] --detach [<branch>]

git checkout [-q] [-f] [-m] [--detach] <commit>

git checkout [-q] [-f] [-m] [[-b|-B|--orphan] <new_branch>] [<start_point>]

git checkout [-f|--ours|--theirs|-m|--conflict=<style>] [<tree-ish>] [--] <paths>…

git checkout [-p|--patch] [<tree-ish>] [--] [<paths>…]

- **checkout-index**

Copia ficheros del índice al árbol de trabajo.

git checkout-index [-u] [-q] [-a] [-f] [-n] [--prefix=<string>]

[--stage=<number>|all]

[--temp]

[-z] [--stdin]

[--] [<file>…]

- **cherry**

Encuentra “commits” que aún faltan por aplicar al “upstream”.

git cherry [-v] [<upstream> [<head> [<limit>]]]

- **cherry-pick**

Aplica los cambios introducidos por algunos “commits” existentes.

git cherry-pick [--edit] [-n] [-m parent-number] [-s] [-x] [--ff]

[-S[<key-id>]] <commit>…

git cherry-pick --continue

git cherry-pick --quit

git cherry-pick --abort

- **citool**

Alternativa gráfica para git-commit.

git citool

- **clean**

Remueve ficheros sin rastrear del árbol de trabajo.

git clean [-d] [-f] [-i] [-n] [-q] [-e <pattern>] [-x | -X] [--] <path>…

- **clone**

Clona un repositorio en un nuevo directorio.

git clone [--template=<template_directory>]

[-l] [-s] [--no-hardlinks] [-q] [-n] [--bare] [--mirror]

[-o <name>] [-b <name>] [-u <upload-pack>] [--reference <repository>]

[--separate-git-dir <git dir>]

[--depth <depth>] [--[no-]single-branch]

[--recursive | --recurse-submodules] [--] <repository>

[<directory>]

- **column**

Muestra datos en columnas.

git column [--command=<name>] [--[raw-]mode=<mode>] [--width=<width>]

[--indent=<string>] [--nl=<string>] [--padding=<n>]

- **commit**

Registra cambios al repositorio.

git commit [-a | --interactive | --patch] [-s] [-v] [-u<mode>] [--amend]

[--dry-run] [(-c | -C | --fixup | --squash) <commit>]

[-F <file> | -m <msg>] [--reset-author] [--allow-empty]

[--allow-empty-message] [--no-verify] [-e] [--author=<author>]

[--date=<date>] [--cleanup=<mode>] [--[no-]status]

[-i | -o] [-S[<key-id>]] [--] [<file>…]

- **commit-tree**

Crea un nuevo objeto “commit”.

git commit-tree <tree> [(-p <parent>)…] < changelog

git commit-tree [(-p <parent>)…] [-S[<keyid>]] [(-m <message>)…]

[(-F <file>)…] <tree>

- **config**

Obtiene y asigna opciones de repositorio o globales.

git config [<file-option>] [type] [-z|--null] name [value [value_regex]]

git config [<file-option>] [type] --add name value

git config [<file-option>] [type] --replace-all name value [value_regex]

git config [<file-option>] [type] [-z|--null] --get name [value_regex]

git config [<file-option>] [type] [-z|--null] --get-all name [value_regex]

git config [<file-option>] [type] [-z|--null] --get-regexp name_regex [value_regex]

git config [<file-option>] [type] [-z|--null] --get-urlmatch name URL

git config [<file-option>] --unset name [value_regex]

git config [<file-option>] --unset-all name [value_regex]

git config [<file-option>] --rename-section old_name new_name

git config [<file-option>] --remove-section name

git config [<file-option>] [-z|--null] -l | --list

git config [<file-option>] --get-color name [default]

git config [<file-option>] --get-colorbool name [stdout-is-tty]

git config [<file-option>] -e | --edit

- **count-objects**

Cuenta el número de objetos no empaquetados y su consumo de disco.

git count-objects [-v] [-H | --human-readable]

- **credential**

Recupera y guarda credenciales de usuario.

git credential <fill|approve|reject>

- **credential-store**

Ayudante para guardar credenciales en disco.

git config credential.helper 'store [options]'

- **daemon**

Un servidor realmente simple para repositorios Git.

git daemon [--verbose] [--syslog] [--export-all]

[--timeout=<n>] [--init-timeout=<n>] [--max-connections=<n>]

[--strict-paths] [--base-path=<path>] [--base-path-relaxed]

[--user-path | --user-path=<path>]

[--interpolated-path=<pathtemplate>]

[--reuseaddr] [--detach] [--pid-file=<file>]

[--enable=<service>] [--disable=<service>]

[--allow-override=<service>] [--forbid-override=<service>]

[--access-hook=<path>] [--[no-]informative-errors]

[--inetd |

[--listen=<host_or_ipaddr>] [--port=<n>]

[--user=<user> [--group=<group>]]]

[<directory>…]

- **describe**

Muestra la etiqueta más reciente que es accesible a partir de un “commit”.

git describe [--all] [--tags] [--contains] [--abbrev=<n>] <commit-ish>…

git describe [--all] [--tags] [--contains] [--abbrev=<n>] --dirty[=<mark>]

- **diff**

Muestra cambios entre “commits”, “commit” y el árbol de trabajo, etc.

git diff [options] [<commit>] [--] [<path>…]

git diff [options] --cached [<commit>] [--] [<path>…]

git diff [options] <commit> <commit> [--] [<path>…]

git diff [options] <blob> <blob>

git diff [options] [--no-index] [--] <path> <path>

- **diff-files**

Compara ficheros en el árbol de trabajo y el índice.

git diff-files [-q] [-0|-1|-2|-3|-c|--cc] [<common diff options>] [<path>…]

- **diff-index**

Compara un árbol con el árbol de trabajo o índice.

git diff-index [-m] [--cached] [<common diff options>] <tree-ish> [<path>…]

- **difftool**

Muestra cambios utilizando herramientas “diff” comunes.

git difftool [<options>] [<commit> [<commit>]] [--] [<path>…]

- **fast-export**

Exportador de datos Git.

git fast-export [options] | git fast-import

- **fast-import**

“Backend” para importadores rápidos de datos Git.

frontend | *git fast-import* [options]

- **fetch**

Descarga objetos y referencias de otro repositorio.

git fetch [<options>] [<repository> [<refspec>…]]

git fetch [<options>] <group>

git fetch --multiple [<options>] [(<repository> | <group>)…]

git fetch --all [<options>]

- **fetch-pack**

Recibe objetos perdidos de otro repositorio.

git fetch-pack [--all] [--quiet|-q] [--keep|-k] [--thin] [--include-tag]

[--upload-pack=<git-upload-pack>]

[--depth=<n>] [--no-progress]

[-v] <repository> [<refs>…]

- **filter-branch**

Reescribe ramas.

git filter-branch [--env-filter <command>] [--tree-filter <command>]

[--index-filter <command>] [--parent-filter <command>]

[--msg-filter <command>] [--commit-filter <command>]

[--tag-name-filter <command>] [--subdirectory-filter <directory>]

[--prune-empty]

[--original <namespace>] [-d <directory>] [-f | --force]

[--] [<rev-list options>…]

- **fmt-merge-msg**

Produce un mensaje “commit” de unión.

git fmt-merge-msg [-m <message>] [--log[=<n>] | --no-log] <$GIT_DIR/FETCH_HEAD

git fmt-merge-msg [-m <message>] [--log[=<n>] | --no-log] -F <file>

- **for-each-ref**

Emite información en cada referencia.

git for-each-ref [--count=<count>] [--shell|--perl|--python|--tcl]

[(--sort=<key>)…] [--format=<format>] [<pattern>…]

- **format-patch**

Prepara parches para presentación de e-mail.

git format-patch [-k] [(-o|--output-directory) <dir> | --stdout]

[--no-thread | --thread[=<style>]]

[(--attach|--inline)[=<boundary>] | --no-attach]

[-s | --signoff]

[--signature=<signature> | --no-signature]

[-n | --numbered | -N | --no-numbered]

[--start-number <n>] [--numbered-files]

[--in-reply-to=Message-Id] [--suffix=.<sfx>]

[--ignore-if-in-upstream]

[--subject-prefix=Subject-Prefix] [(--reroll-count|-v) <n>]

[--to=<email>] [--cc=<email>]

[--[no-]cover-letter] [--quiet] [--notes[=<ref>]]

[<common diff options>]

[<since> | <revision range>]

- **fsck**

Verifica la conectividad y validez de los objetos en la base de datos.

git fsck [--tags] [--root] [--unreachable] [--cache] [--no-reflogs]

[--[no-]full] [--strict] [--verbose] [--lost-found]

[--[no-]dangling] [--[no-]progress] [<object>*]

- **fsck-objects**

Verifica la conectividad y validez de los objetos en la base de datos.

git fsck-objects …

- **gc**

Limpia ficheros innecesarios y optimiza el repositorio local.

git gc [--aggressive] [--auto] [--quiet] [--prune=<date> | --no-prune] [--force]

- **get-tar-commit-id**

Extrae identificación de “commit” de un fichero creado usando git-archive.

git get-tar-commit-id < <tarfile>

- **git**

El rastreador estúpido de contenido.

git [--version] [--help] [-C <path>] [-c <name>=<value>]

[--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]

[-p|--paginate|--no-pager] [--no-replace-objects] [--bare]

[--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]

<command> [<args>]

- **grep**

Imprime líneas iguales a un patrón.

git grep [-a | --text] [-I] [--textconv] [-i | --ignore-case] [-w | --word-regexp]

[-v | --invert-match] [-h|-H] [--full-name]

[-E | --extended-regexp] [-G | --basic-regexp]

[-P | --perl-regexp]

[-F | --fixed-strings] [-n | --line-number]

[-l | --files-with-matches] [-L | --files-without-match]

[(-O | --open-files-in-pager) [<pager>]]

[-z | --null]

[-c | --count] [--all-match] [-q | --quiet]

[--max-depth <depth>]

[--color[=<when>] | --no-color]

[--break] [--heading] [-p | --show-function]

[-A <post-context>] [-B <pre-context>] [-C <context>]

[-W | --function-context]

[-f <file>] [-e] <pattern>

[--and|--or|--not|(|)|-e <pattern>…]

[[--[no-]exclude-standard] [--cached | --no-index | --untracked] | <tree>…]

[--] [<pathspec>…]

- **gui**

Una interfaz gráfica portable de Git.

git gui [<command>] [arguments]

- **hash-object**

Computa la identificación de objeto y opcionalmente crea un “blob” de un fichero.

git hash-object [-t <type>] [-w] [--path=<file>|--no-filters] [--stdin] [--] <file>…

git hash-object [-t <type>] [-w] --stdin-paths [--no-filters] < <list-of-paths>

- **help**

Muestra información de ayuda sobre Git.

git help [-a|--all] [-g|--guide]

[-i|--info|-m|--man|-w|--web] [COMMAND|GUIDE]

- **http-backend**

Implementación de lado del servidor de Git en HTTP.

git http-backend

- **http-fetch**

Descarga de un repositorio remoto Git a través de HTTP.

git http-fetch [-c] [-t] [-a] [-d] [-v] [-w filename] [--recover] [--stdin] <commit> <url>

- **http-push**

Empuja objetos con HTTP/DAV hacia otro repositorio.

git http-push [--all] [--dry-run] [--force] [--verbose] <url> <ref> [<ref>…]

- **imap-send**

Envía una colección de parches de stdin hacia un directorio IMAP.

git imap-send

- **index-pack**

Construye fichero índice empaquetado para un archivo empaquetado existente.

git index-pack [-v] [-o <index-file>] <pack-file>

git index-pack --stdin [--fix-thin] [--keep] [-v] [-o <index-file>]

[<pack-file>]

- **init**

Crea un repositorio Git vacío o reinicializa uno existente.

git init [-q | --quiet] [--bare] [--template=<template_directory>]

[--separate-git-dir <git dir>]

[--shared[=<permissions>]] [directory]

- **init-db**

Crea un repositorio Git vacío

git init-db [-q | --quiet] [--bare] [--template=<template_directory>] [--separate-git-dir <git dir>] [--shared[=<permissions>]]

- **log**

Muestra registros “commit”.

git log [<options>] [<revision range>] [[--] <path>…]

- **ls-files**

Muestra información acerca de ficheros en el índice y el árbol de trabajo.

git ls-files [-z] [-t] [-v]

(--[cached|deleted|others|ignored|stage|unmerged|killed|modified])*

(-[c|d|o|i|s|u|k|m])*

[-x <pattern>|--exclude=<pattern>]

[-X <file>|--exclude-from=<file>]

[--exclude-per-directory=<file>]

[--exclude-standard]

[--error-unmatch] [--with-tree=<tree-ish>]

[--full-name] [--abbrev] [--] [<file>…]

- **ls-remote**

Enlista referencias en un repositorio remoto.

git ls-remote [--heads] [--tags] [-u <exec> | --upload-pack <exec>]

[--exit-code] <repository> [<refs>…]

- **ls-tree**

Enlista los contenidos de un objeto árbol.

git ls-tree [-d] [-r] [-t] [-l] [-z]

[--name-only] [--name-status] [--full-name] [--full-tree] [--abbrev[=<n>]]

<tree-ish> [<path>…]

- **mailinfo**

Extrae parche y autoría de un único mensaje e-mail.

git mailinfo [-k|-b] [-u | --encoding=<encoding> | -n] [--[no-]scissors] <msg> <patch>

- **mailsplit**

Programa separador simple UNIX mbox.

git mailsplit [-b] [-f<nn>] [-d<prec>] [--keep-cr] -o<directory> [--] [(<mbox>|<Maildir>)…]

- **merge**

Une dos o más historiales de desarrollo.

git merge [-n] [--stat] [--no-commit] [--squash] [--[no-]edit]

[-s <strategy>] [-X <strategy-option>] [-S[<key-id>]]

[--[no-]rerere-autoupdate] [-m <msg>] [<commit>…]

git merge <msg> HEAD <commit>…

git merge --abort

- **merge-base**

Encuentra tan buenos ancestros comunes como sea posible para una unión.

git merge-base [-a|--all] <commit> <commit>…

git merge-base [-a|--all] --octopus <commit>…

git merge-base --is-ancestor <commit> <commit>

git merge-base --independent <commit>…

git merge-base --fork-point <ref> [<commit>]

- **merge-file**

Ejecuta una unión de fichero de tres vías.

git merge-file [-L <current-name> [-L <base-name> [-L <other-name>]]]

[--ours|--theirs|--union] [-p|--stdout] [-q|--quiet] [--marker-size=<n>]

[--[no-]diff3] <current-file> <base-file> <other-file>

- **merge-index**

Ejecuta una unión para ficheros que necesiten unirse.

git merge-index [-o] [-q] <merge-program> (-a | [--] <file>*)

- **merge-one-file**

El programa ayudante estándar para utilizar con git-merge-index.

git merge-one-file

- **merge-tree**

Muestra una unión de tres vías sin tocar el índice.

git merge-tree <base-tree> <branch1> <branch2>

- **mergetool**

Ejecuta herramientas de resolución de conflicto de unión para resolver conflictos de unión.

git mergetool [--tool=<tool>] [-y | --[no-]prompt] [<file>…]

- **mktag**

Crea un objeto etiqueta.

git mktag < signature_file

- **mktree**

Construye un objeto árbol de texto formateado ls-tree.

git mktree [-z] [--missing] [--batch]

- **mv**

Mueve o renombra un fichero, un directorio, o un “symlink”.

git mv <options>… <args>…

- **name-rev**

Encuentra nombres simbólicos para revisiones determinadas.

git name-rev [--tags] [--refs=<pattern>]

(--all | --stdin | <commit-ish>…)

- **notes**

Añade o inspecciona notas de objeto.

git notes [list [<object>]]

git notes add [-f] [-F <file> | -m <msg> | (-c | -C) <object>] [<object>]

git notes copy [-f] (--stdin | <from-object> <to-object>)

git notes append [-F <file> | -m <msg> | (-c | -C) <object>] [<object>]

git notes edit [<object>]

git notes show [<object>]

git notes merge [-v | -q] [-s <strategy>] <notes-ref>

git notes merge --commit [-v | -q]

git notes merge --abort [-v | -q]

git notes remove [--ignore-missing] [--stdin] [<object>…]

git notes prune [-n | -v]

git notes get-ref

- **p4**

Importa de y envía a repositorios Perforce.

git p4 clone [<sync options>] [<clone options>] <p4 depot path>…

git p4 sync [<sync options>] [<p4 depot path>…]

git p4 rebase

git p4 submit [<submit options>] [<master branch name>]

- **pack-objects**

Crea un archivo empaquetado de objetos.

git pack-objects [-q | --progress | --all-progress] [--all-progress-implied]

[--no-reuse-delta] [--delta-base-offset] [--non-empty]

[--local] [--incremental] [--window=<n>] [--depth=<n>]

[--revs [--unpacked | --all]] [--stdout | base-name]

[--keep-true-parents] < object-list

- **pack-redundant**

Encuentra ficheros paquete redundantes.

git pack-redundant [--verbose] [--alt-odb] < --all | .pack filename … >

- **pack-refs**

Empaqueta encabezados y etiquetas para acceso eficiente de repositorio.

git pack-refs [--all] [--no-prune]

- **patch-id**

Computa identificación única para un parche.

git patch-id < <patch>

- **prune**

Limpia todos los objetos inalcanzables de la base de datos de objetos.

git prune [-n] [-v] [--expire <expire>] [--] [<head>…]

- **prune-packed**

Remueve objetos extra que ya están en ficheros paquete.

git prune-packed [-n|--dry-run] [-q|--quiet]

- **pull**

Trae de e integra con otro repositorio o una rama local.

git pull [options] [<repository> [<refspec>…]]

- **push**

Actualiza referencias remotas con objetos asociados.

git push [--all | --mirror | --tags] [--follow-tags] [-n | --dry-run] [--receive-pack=<git-receive-pack>]

[--repo=<repository>] [-f | --force] [--prune] [-v | --verbose] [-u | --set-upstream]

[--force-with-lease[=<refname>[:<expect>]]]

[--no-verify] [<repository> [<refspec>…]]

- **quiltimport**

Aplica un “quilt patchset” a una rama actual.

git quiltimport [--dry-run | -n] [--author <author>] [--patches <dir>]

- **read-tree**

Lee información de árbol en el índice.

git read-tree [[-m [--trivial] [--aggressive] | --reset | --prefix=<prefix>]

[-u [--exclude-per-directory=<gitignore>] | -i]]

[--index-output=<file>] [--no-sparse-checkout]

(--empty | <tree-ish1> [<tree-ish2> [<tree-ish3>]])

- **rebase**

Envía “commits” locales al encabezado “upstream” actualizado.

git rebase [-i | --interactive] [options] [--exec <cmd>] [--onto <newbase>]

[<upstream>] [<branch>]

git rebase [-i | --interactive] [options] [--exec <cmd>] [--onto <newbase>]

--root [<branch>]

git rebase --continue | --skip | --abort | --edit-todo

- **receive-pack**

Recibe lo que es empujado a un repositorio.

git-receive-pack <directory>

- **reflog**

Maneja información “reflog”.

git reflog <subcommand> <options>

- **relink**

Vincula objetos comunes in repositorios locales.

git relink [--safe] <dir>… <master_dir>

- **remote**

Maneja un conjunto de repositorios rastreados.

git remote [-v | --verbose]

git remote add [-t <branch>] [-m <master>] [-f] [--[no-]tags] [--mirror=<fetch|push>] <name> <url>

git remote rename <old> <new>

git remote remove <name>

git remote set-head <name> (-a | --auto | -d | --delete | <branch>)

git remote set-branches [--add] <name> <branch>…

git remote set-url [--push] <name> <newurl> [<oldurl>]

git remote set-url --add [--push] <name> <newurl>

git remote set-url --delete [--push] <name> <url>

git remote [-v | --verbose] show [-n] <name>…

git remote prune [-n | --dry-run] <name>…

git remote [-v | --verbose] update [-p | --prune] [(<group> | <remote>)…]

- **remote-ext**

Enlaza transporte inteligente hacia comando externo.

git remote add <nick> "ext::<command>[<arguments>…]"

- **remote-fd**

Refleja un stream de transporte inteligente de vuelta al que llama.

"fd::<infd>[,<outfd>][/<anything>]" (as URL)

- **repack**

Empaqueta objetos no empaquetados en un repositorio.

git repack [-a] [-A] [-d] [-f] [-F] [-l] [-n] [-q] [-b] [--window=<n>] [--depth=<n>]

- **replace**

Crea, enlista, elimina referencias a objetos reemplazar.

git replace [-f] <object> <replacement>

git replace -d <object>…

git replace [--format=<format>] [-l [<pattern>]]

- **request-pull**

Genera un resumen de cambios pendientes.

git request-pull [-p] <start> <url> [<end>]

- **rerere**

Reúsa una resolución registrada de uniones en conflicto.

git rerere [clear|forget <pathspec>*|diff|remaining|status|gc]*

- **reset**

Reajusta HEAD actual al estado especificado.

git reset [-q] [<tree-ish>] [--] <paths>…

git reset (--patch | -p) [<tree-ish>] [--] [<paths>…]

git reset [--soft | --mixed [-N] | --hard | --merge | --keep] [-q] [<commit>]

- **rev-list**

Enlista objetos “commit” en orden cronológico invertido.

git rev-list [--max-count=<number>]

[--skip=<number>]

[--max-age=<timestamp>]

[--min-age=<timestamp>]

[--sparse]

[--merges]

[--no-merges]

[--min-parents=<number>]

[--no-min-parents]

[--max-parents=<number>]

[--no-max-parents]

[--first-parent]

[--remove-empty]

[--full-history]

[--not]

[--all]

[--branches[=<pattern>]]

[--tags[=<pattern>]]

[--remotes[=<pattern>]]

[--glob=<glob-pattern>]

[--ignore-missing]

[--stdin]

[--quiet]

[--topo-order]

[--parents]

[--timestamp]

[--left-right]

[--left-only]

[--right-only]

[--cherry-mark]

[--cherry-pick]

[--encoding=<encoding>]

[--(author|committer|grep)=<pattern>]

[--regexp-ignore-case | -i]

[--extended-regexp | -E]

[--fixed-strings | -F]

[--date=(local|relative|default|iso|rfc|short)]

[[--objects | --objects-edge] [--unpacked]]

[--pretty | --header]

[--bisect]

[--bisect-vars]

[--bisect-all]

[--merge]

[--reverse]

[--walk-reflogs]

[--no-walk] [--do-walk]

[--use-bitmap-index]

<commit>… [-- <paths>…]

- **rev-parse**

Selecciona y masajea parámetros.

git rev-parse [--option] <args>…

- **revert**

Revierte algunos “commits” existentes.

git revert [--[no-]edit] [-n] [-m parent-number] [-s] [-S[<key-id>]] <commit>…

git revert --continue

git revert --quit

git revert --abort

- **rm**

Remueve ficheros de un árbol de trabajo y del índice.

git rm [-f | --force] [-n] [-r] [--cached] [--ignore-unmatch] [--quiet] [--] <file>…

- **send-email**

Envía una colección de parches como emails.

git send-email [options] <file|directory|rev-list options>…

- **send-pack**

Empuja objetos con el protocolo Git a otro repositorio.

git send-pack [--all] [--dry-run] [--force] [--receive-pack=<git-receive-pack>] [--verbose] [--thin] [<host>:]<directory> [<ref>…]

- **shortlog**

Resume salida de “git log”.

git log --pretty=short | *git shortlog* [<options>]

git shortlog [<options>] [<revision range>] [[--] <path>…]

- **show**

Muestra varios tipos de objetos.

git show [options] <object>…

- **show-branch**

Muestra ramas y sus “commits”.

git show-branch [-a|--all] [-r|--remotes] [--topo-order | --date-order]

[--current] [--color[=<when>] | --no-color] [--sparse]

[--more=<n> | --list | --independent | --merge-base]

[--no-name | --sha1-name] [--topics]

[(<rev> | <glob>)…]

git show-branch (-g|--reflog)[=<n>[,<base>]] [--list] [<ref>]

- **show-index**

Muestra archivo de índice empaquetado.

git show-index < idx-file

- **show-ref**

Enlista referencias en un repositorio local.

git show-ref [-q|--quiet] [--verify] [--head] [-d|--dereference]

[-s|--hash[=<n>]] [--abbrev[=<n>]] [--tags]

[--heads] [--] [<pattern>…]

git show-ref --exclude-existing[=<pattern>] < ref-list

- **stage**

Añade contenidos de fichero al área de preparación.

git stage args…

- **stash**

Esconde los cambios en un directorio de trabajo sucio.

git stash list [<options>]

git stash show [<stash>]

git stash drop [-q|--quiet] [<stash>]

git stash (pop | apply) [--index] [-q|--quiet] [<stash>]

git stash branch <branchname> [<stash>]

git stash [save [-p|--patch] [-k|--[no-]keep-index] [-q|--quiet]

[-u|--include-untracked] [-a|--all] [<message>]]

git stash clear

git stash create [<message>]

git stash store [-m|--message <message>] [-q|--quiet] <commit>

- **status**

Muestra el status del árbol de trabajo.

git status [<options>…] [--] [<pathspec>…]

- **stripspace**

Remueve espacio en blanco innecesario.

git stripspace [-s | --strip-comments] < input

- **submodule**

Inicializa, actualiza o inspecciona submódulos.

git submodule [--quiet] add [-b <branch>] [-f|--force] [--name <name>]

[--reference <repository>] [--depth <depth>] [--] <repository> [<path>]

git submodule [--quiet] status [--cached] [--recursive] [--] [<path>…]

git submodule [--quiet] init [--] [<path>…]

git submodule [--quiet] deinit [-f|--force] [--] <path>…

git submodule [--quiet] update [--init] [--remote] [-N|--no-fetch]

[-f|--force] [--rebase|--merge] [--reference <repository>]

[--depth <depth>] [--recursive] [--] [<path>…]

git submodule [--quiet] summary [--cached|--files] [(-n|--summary-limit) <n>]

[commit] [--] [<path>…]

git submodule [--quiet] foreach [--recursive] <command>

git submodule [--quiet] sync [--] [<path>…]

- **svn**

Operación bidireccional entre un repositorio Subversion y Git.

git svn <command> [options] [arguments]

- **symbolic-ref**

Lee, modifica y elimina referencias simbólicas.

git symbolic-ref [-m <reason>] <name> <ref>

git symbolic-ref [-q] [--short] <name>

git symbolic-ref --delete [-q] <name>

- **tag**

Crea, enlista, elimina o verifica un objeto etiqueta firmado con GPG.

git tag [-a | -s | -u <key-id>] [-f] [-m <msg> | -F <file>]

<tagname> [<commit> | <object>]

git tag -d <tagname>…

git tag [-n[<num>]] -l [--contains <commit>] [--points-at <object>]

[--column[=<options>] | --no-column] [<pattern>…]

[<pattern>…]

git tag -v <tagname>…

- **unpack-file**

Crea un fichero temporal con el contenido de un “blob”.

git unpack-file <blob>

- **unpack-objects**

Desempaca objetos de un archivo empaquetado.

git unpack-objects [-n] [-q] [-r] [--strict] < <pack-file>

- **update-index**

Registra contenidos de fichero en el árbol de trabajo al índice.

git update-index

[--add] [--remove | --force-remove] [--replace]

[--refresh] [-q] [--unmerged] [--ignore-missing]

[(--cacheinfo <mode>,<object>,<file>)…]

[--chmod=(+|-)x]

[--[no-]assume-unchanged]

[--[no-]skip-worktree]

[--ignore-submodules]

[--really-refresh] [--unresolve] [--again | -g]

[--info-only] [--index-info]

[-z] [--stdin] [--index-version <n>]

[--verbose]

[--] [<file>…]

- **update-ref**

Actualiza seguramente el nombre de objeto guardado en una referencia.

git update-ref [-m <reason>] (-d <ref> [<oldvalue>] | [--no-deref] <ref> <newvalue> [<oldvalue>] | --stdin [-z])

- **update-server-info**

Actualiza fichero de información auxiliar para ayudar servidores tontos.

git update-server-info [--force]

- **upload-archive**

Envía fichero de vuelta a git-archive.

git upload-archive <directory>

- **upload-pack**

Envía objetos empaquetados de vuelta a git-fetch-pack.

git-upload-pack [--strict] [--timeout=<n>] <directory>

- **var**

Muestra una variable lógica Git.

git var (-l | <variable>)

- **verify-pack**

Valida ficheros de archivos Git empaquetados.

git verify-pack [-v|--verbose] [-s|--stat-only] [--] <pack>.idx …

- **verify-tag**

Revisa la firma GPG de etiquetas.

git verify-tag <tag>…

- **web—browse**

Script ayudante Git para lanzar un explorador web.

git web--browse [OPTIONS] URL/FILE …

- **whatchanged**

Muestra registros con diferencias que cada “commit” introduce.

git whatchanged <option>…

- **write-tree**

Crea un objeto árbol del índice actual.

git write-tree [--missing-ok] [--prefix=<prefix>/]