Operaciones comunes.

```
$ git status
# On branch master
# Untracked files:
#   (use "git add <file>..." to include in what will be committed)
#
# README
nothing added to commit but untracked files present (use "git add" to track)
```

para añadir al índice un archivo, marcándolo para que forme parte del próximo `commit (tracking)`

```
git add README
```

Si un archivo que ha sido previamente añadido a la*staging area* es modificado debe ejecutarse el comando `git add <archivo>` de nuevo para *stage* la nueva versión del archivo

Ejemplo de archivo .gitignore

```
# a comment - this is ignored
*.a       # no .a files
!lib.a   # but do track lib.a, even though you're ignoring .a files above
/TODO     # only ignore the root TODO file, not subdir/TODO
build/    # ignore all files in the build/ directory
doc/*.txt # ignore doc/notes.txt, but not doc/server/arch.txt
```

vemos el uso de carácteres comodin, los comentarios y el uso de ! para la negación

diff

`git diff` muestra las diferencias entre lo que hay en el `working directory` y lo que hay en el `staging area`. Por otra parte `git diff --cached` muestra lo que está *staged*, lo que irá en el próximo *commit*

"Commiting" los cambios

```
git commit -m "Story 182: Fix benchmarks for speed"
```

Para evitar añadir previamente a *stage area* los archivos **modificados** que ya se habían añadido previamente se usa la opción `-a`

```
git commit -a -m 'added new benchmarks'
```

Eliminar archivos

Si se borra un archivo `rm leeme.txt` cuando se invoca `git status` se mostrará como modificado pero no *staged* Para que git se olvide del archivo y deje de seguirle la pista, se usa el comando `rm git rm leeme.txt`

Se puede usar sólamente el segundo comando, sin borrar previamente el archivo: git lo eliminará de la *working area*

Se puede renombrar un archivo mediante `git mv file_from file_to`

Para borrar un archivo del repositorio, pero no del sistema de ficheros `git rm --cached mylogfile.log`

1 Revisando el historial de commits

`git log -p -2` muestra las diferencias introducidas en cada commit. `-2` se usa para limitar la salida a dos lineas

`git log --stat` muestra algunas estadísticas abreivada para cada commit. Otra opción interesante es `--pretty`

Otra opción interesante es `format` (ver pág 29 de Pro git)

Para limitar el número de salidas `git log --since=2.weeks` Otras opciones

Opción	**Descripción**
-(n)	Muestra sólo los n últimos commits
–since, –after	Muestra sólo los hechos después de una fecha
–until, –before	los hechos después de una fecha
– author	los hechos por el autor
– committer	los hechos por el committer

Un comando para mostrar el log `git log --pretty`oneline =