

```
alias gitgraph="git log --all --decorate --oneline --graph"
```

Ejercicio 1:

1. 'git init'
Initialized empty Git repository in /home/alumnos/ccasado/ISI/repo-git-1/.git/
[Creamos un repositorio local]
2. 'touch f1'
'git add f1'
'touch f2'
'git add f2'
[Creamos los ficheros f1 y f2]
3. 'git commit -m "Creados f1 y f2"'
[commit para añadir los ficheros junto con su comentario correspondiente]
4. 'gitgraph'
[* 1abdf56 (HEAD -> master) Creados f1 y f2]
5. 'echo 'hola' > f1'
[escribo en el fichero f1: 'hola']
6. 'git commit -a -m "f1 ahora contiene 'hola'"'
[actualiza f1 y hace un comentario sobre la acción anterior]
7. 'git rm f2'
'git commit -a -m "borrado f2"'
[borra f2 y actualiza el estado y comenta la acción]
8. 'git reset --soft HEAD^'
'git checkout HEAD f2'
[lo primero deshace el último commit, lo segundo trae el fichero a working area]
9. 'git commit --amend -m "modificado f1"'
[no haces un nuevo commit sino que modifico el anterior]

FIN:

```
'gitgraph'  
* ca27dff (HEAD -> master) modificado f1  
* d94fd00 f1 ahora contiene 'hola'  
* 1abdf56 Creados f1 y f2
```

Ejercicio 2:

1. 'git init'
Initialized empty Git repository in /home/alumnos/ccasado/ISI/repo-git-2/.git/
[Creamos un repositorio local]
2. 'touch s1'
'git add s1'
'git commit -m "C1: añadido s1"'
[me creo un fichero s1 y en atom por ejemplo, lo relleno con la lista adjunta en el ejercicio, lo añado y hago el commit]
3. 'touch s2'
'git add s2'
[me creo un fichero s2 y en atom por ejemplo, lo relleno con la lista adjunta en el ejercicio y lo añado]
4. Mediante un editor de texto elimino 'Queso'.
5. 'git add s1'
[Añado al stage area s1 y s2]
6. 'git diff'
[Comparacion entre lo que hay en stage area y working area que al haber hecho add ya hemos incluido lo de working area en stage area.]
7. 'git diff --staged'
[Comparamos ahora entre stage area y HEAD, vemos que hay diferencias puesto que no hemos hecho commit de los cambios en s1 y no he añadido s2].
8. 'git commit -am "C2: s1 no tiene queso, s2 tiene pan"'
[Creado u nuevo commit pasando a HEAD lo que tenemos nuevo en stage area.]
9. 'git log -p'
commit 32fe3b514f68695c6f7933559bc97852ea6f94d8
Author: cgomezcasa <cayegomez1995@gmail.com>
Date: Wed Oct 25 11:47:47 2017 +0200

C2: s1 no tiene queso, s2 tiene pan

```
diff --git a/s1 b/s1
index d53da07..3d810d7 100644
--- a/s1
+++ b/s1
@@ -1,5 +1,4 @@
Leche
Tomate pelado
Tomate ecológico
-Queso
Yoghourt griego
diff --git a/s2 b/s2
new file mode 100644
```

index 00000000..65956e6

--- /dev/null

+++ b/s2

@@ -0,0 +1,6 @@

+Leche

+Tomate pelado

+Tomate ecológico

+Queso

+Yoghourt griego

+Pan

commit 354b13b3966ea63b07551b77843f9c919313440f

Author: cgomezcasa <cayegomez1995@gmail.com>

Date: Mon Oct 23 12:49:26 2017 +0200

C1: añadido s1

diff --git a/s1 b/s1

new file mode 100644

index 00000000..d53da07

--- /dev/null

+++ b/s1

@@ -0,0 +1,5 @@

+Leche

+Tomate pelado

+Tomate ecológico

+Queso

+Yoghourt griego

10. 'git rm s2'
'git commit -am "C3: borrado s2"'
[borrado s2 y luego commit indicando que ha sido borrado]
11. 'atom s1'
[mediante un editor de texto eliminamos las lineas de s1]
12. 'git diff'
[comparamos las versiones de working area y stage area respecto a s1 viendo que stage area aún contiene todo lo que hemos borrado en working area en el paso anterior].
13. 'git checkout -- s1'
[recuperamos versión de s1 del stage area]
14. 'git diff'
[de nuevo comprobamos diferencias entre las dos areas stage y working, que en éste caso ya no hay al haber llevado en el anterior punto a working area lo que habia en stage]
15. 'atom s1'
[Editamos el fichero s1]
16. 'git add s1'
[Añadimos lo que tenemos en working area respecto a S1 a stage area]

17. 'git diff'
[de nuevo comprobamos diferencias entre las dos areas stage y working, que en éste caso no hay ya que hemos añadido los cambios anteriores qe hicimos en s1 a stage area]
18. 'git diff --staged'
'git status'
[comprobamos diferencias entre stage area y HEAD mediante el primer comando]
En la rama master
Cambios para hacer commit:
(use «git reset HEAD <archivo>...» para sacar del stage)

modificado: s1

[Con el segundo comando vemos que nos sale en verde de modo que tenemos la misma versión en working y stage area pero no en HEAD]

19. a) 'git reset -- s1'
[Pasamos la versión de HEAD a stage area]
b) 'git diff --staged'
'git status'
[Observamos que ya no hay diferencia entre el HEAD y stage area puesto que a stage area le hemos pasado la versión que había en HEAD]
En la rama master
Cambios no preparados para el commit:
(use «git add <archivo>...» para actualizar lo que se confirmará)
(use «git checkout -- <archivo>...» para descartar cambios en el directorio de trabajo)

modificado: s1

no hay cambios agregados al commit (use «git add» o «git commit -a»)

[Te está diciendo que hay diferencia entre la versión que tenemos entre working y stage area (rojo)]

- c) 'git checkout -- s1'
[Traigo la versión de stage area a working area]
d) 'git diff'
'git diff --staged'
'git status'

[Los dos primeros comandos nos muestran que es todo lo mismo, que no falta nada y el tercer comando que en las tres areas tenemos lo mismo]

En la rama master

nothing to commit, working directory clean

20. [Todas las comparaciones hechas en el punto anterior]
'gitgraph'
* 378c093 (HEAD -> master) C3: borrado s2
* 32fe3b5 C2: s1 no tiene queso, s2 tiene pan
* 354b13b C1: añadido s1

21. 'ls'
[Vemos que s2 no está en working area]
'git log'

`'git checkout 32fe3b514f68695c6f7933559bc97852ea6f94d8 s2'`

[Queremos traer el fichero de HEAD que borramos en el último commit, por ello buscamos el id del commit que queremos recuperar con el primer comando y con el segundo lo recuperamos]

22. `'git commit -a -m "C4: recuperado s2"'`
[hago un nuevo commit con el comentario 4]
26. `'*.java
/logs*.log'`
[En .gitignore incluyo todo lo anterior]
27. `'git add .'`
[Añade todo lo cambiado en el directorio actual]

Ejercicio 3:

1. `'git init'`
Initialized empty Git repository in /home/ccasado/Documents/ISI-git/repo-git-3/.git/
[Creamos un repositorio local]
2. `'touch s1'`
`'git add s1'`
`'git commit -m "C1: añadido s1"'`
[me creo un fichero s1 y en atom por ejemplo, lo relleno con la lista adjunta en el ejercicio, lo añado y hago el commit]
5. `'git add s2'`
`'git commit -m "C2: añadido s2"'`
6. `'git branch devel'`
7. `'git branch bugFix'`
8. `* 2c57ec4 (HEAD, master, devel, bugFix) C2: creado s2`
`* 0cf8731 C1: creado s1`

bugFix
devel
* master
9. `'git checkout devel'`
11. `'git add s1'`
`'git commit -m "C3: añadida Fregona"'`
12. `'cat s1'`
Leche
Tomate pelado
Tomate ecológico
Queso

Yoghourt griego
Fregona

13. 'git checkout bugFix'

14. 'cat s1'
Leche
Tomate pelado
Tomate ecológico
Queso
Yoghourt griego

15. 'git checkout devel'

16. 'cat s1 '
Leche
Tomate pelado
Tomate ecológico
Queso
Yoghourt griego
Fregona

17. 'git checkout bugFix'

19. 'git log'
commit 2c57ec4d2c8c674494736286a1acdba3cf2ad521
Author: Cayetana Gómez Casado <cayegomez1995@gmail.com>
Date: Sun Nov 12 05:35:40 2017 -0800

C2: creado s2

commit 0cf8731ef356b4c083e640943800550ec974c7af
Author: Cayetana Gómez Casado <cayegomez1995@gmail.com>
Date: Sun Nov 12 05:32:55 2017 -0800

C1: creado s1

[Te muestra historial de commits]

'gitgraph'
* 3e9d29b (devel) C3: añadida Fregona
* 2c57ec4 (HEAD, master, bugFix) C2: creado s2
* 0cf8731 C1: creado s1
[commits y ramas hasta el momento]

'git status'
On branch bugFix
Changes not staged for commit:
(use "git add <file>..." to update what will be committed)
(use "git checkout -- <file>..." to discard changes in working directory)

modified: s1

Untracked files:

(use "git add <file>..." to include in what will be committed)

s1~

no changes added to commit (use "git add" and/or "git commit -a")

[Diferencias entre directorio de trabajo y staged area puesto que no he hecho add de los cambios en el fichero y entre staged area y commits que no hay cambios]

'git diff'

diff --git a/s1 b/s1

index 5aad170..9384b0e 100644

--- a/s1

+++ b/s1

@@ -1,4 +1,4 @@

-Leche

+Leche desnatada

Tomate pelado

Tomate ecológico

Queso

[Diferencias entre versión actual y versión en stage area, debido a que no hemos hecho add tras el cambio en el directorio actual]

'git diff --staged'

[Diferencia entre versión en stage area y último commit, lo cual no hay puesto que no hemos hecho ningún add nuevo tras el último commit]

20. 'git add s1'

'git commit -m "C4: Cambiada Leche por Leche desnatada"'

* 2304881 (HEAD, bugFix) C4: Cambiada Leche por Leche desnatada

| * 3e9d29b (devel) C3: añadida Fregona

|/

* 2c57ec4 (master) C2: creado s2

* 0cf8731 C1: creado s1

Merge con Fast Forward:

2. 'git checkout master'

3. 'git diff master..devel'

diff --git a/s1 b/s1

index 5aad170..494122a 100644

--- a/s1

+++ b/s1

@@ -3,3 +3,4 @@ Tomate pelado

Tomate ecológico

Queso

Yoghourt griego

+Fregona

[Te muestra la diferencia entre ramas que se trata de 'Fregona' que ha sido añadido a s1 de master]

4. 'gitgraph'
* 2304881 (bugFix) C4: Cambiada Leche por Leche desnatada
| * 3e9d29b (devel) C3: añadida Fregona
|/
* 2c57ec4 (HEAD, master) C2: creado s2
* 0cf8731 C1: creado s1

5. 'git merge devel'
[Mezcla de ambas ramas]

6. 'gitgraph'
* 2304881 (bugFix) C4: Cambiada Leche por Leche desnatada
| * 3e9d29b (HEAD, master, devel) C3: añadida Fregona
|/
* 2c57ec4 C2: creado s2
* 0cf8731 C1: creado s1

[Al mezclar las ramas master ha actualizado su versión con la de devel por lo que pasa la etiqueta HEAD al último commit de la rama master que con la mezcla es el C3]

7. 'cat s1 '
Leche
Tomate pelado
Tomate ecológico
Queso
Yoghourt griego
Fregona

Borrar ramas:

1. 'git branch --merged'
devel
* master
[ramas mezcladas con la rama master: devel]

2. 'git branch -d devel '
Deleted branch devel (was 3e9d29b).

3. 'gitgraph'
* 2304881 (bugFix) C4: Cambiada Leche por Leche desnatada
| * 3e9d29b (HEAD, master) C3: añadida Fregona
|/
* 2c57ec4 C2: creado s2
* 0cf8731 C1: creado s1

4. 'git branch -d bugFix '
error: The branch 'bugFix' is not fully merged.
If you are sure you want to delete it, run 'git branch -D bugFix'.
[No te deja borrar la rama puesto que las versiones de ambas son distintas, al no estar mezcladas perderíamos el último commit en el esquema]


```
'git diff master..bugFix '
diff --git a/s1 b/s1
index 494122a..9384b0e 100644
--- a/s1
+++ b/s1
@@ -1,6 +1,5 @@
-Leche
+Leche desnatada
Tomate pelado
Tomate ecológico
Queso
Yoghourt griego
-Fregona
```

Merge a 3 bandas con conflicto resuelto por git:

2. 'git merge bugFix'
3. 'gitgraph'
 - * e8668ff (HEAD, master) C5: Merge branch 'bugFix'
 - | \
 - | * 2304881 (bugFix) C4: Cambiada Leche por Leche desnatada
 - * | 3e9d29b C3: añadida Fregona
 - | /
 - * 2c57ec4 C2: creado s2
 - * 0cf8731 C1: creado s1

```
'git branch --merge'
bugFix
* master
```
4. 'cat s1 '
 - Leche desnatada
 - Tomate pelado
 - Tomate ecológico
 - Queso
 - Yoghourt griego
 - Fregona
5. 'git branch -d bugFix '
 - Deleted branch bugFix (was 2304881).

Merge a 3 bandas con conflicto que requiere solución manual:

1. 'git checkout -b new-feature '
 - Switched to a new branch 'new-feature'
3. 'git diff master..new-feature'
 - [No aparece nada]
 - 'git diff '

```
diff --git a/s1 b/s1
index e51ea94..4fc47aa 100644
--- a/s1
+++ b/s1
@@ -1,6 +1,5 @@
-Leche desnatada
Tomate pelado
-Tomate ecológico
+Tomate orlando
Queso
Yoghourt griego
Fregona
```

'git diff --staged'

[No hay diferencia entre en stage area y el último commit]

4. 'git add s1'

5. 'git diff --staged '

```
diff --git a/s1 b/s1
index e51ea94..4fc47aa 100644
--- a/s1
+++ b/s1
@@ -1,6 +1,5 @@
-Leche desnatada
Tomate pelado
-Tomate ecológico
+Tomate orlando
Queso
Yoghourt griego
Fregona
```

[Al haber hecho add los cambios del directorio de trabajo han pasado al stage area por ello la versión del stage area y el último commit ya no son iguales. En el resto de comandos no nos salta nada puesto que ambas ramas tienen el mismo último commit y el directorio de trabajo y el stage area también tienen la misma versión]

6. 'git commit -m "C6: borrada Leche desnatada, cambiado Tomate ecológico por Tomate Orlando"'

7. 'git checkout master'

9. 'git add s1'

'git commit -m "C7: cambiado Yoghourt griego por Yoghourt y eliminado Tomate ecológico"'

10. 'gitgraph'
 - * 847d06d (HEAD, master) C7: cambiado Yoghourt griego por Yoghourt y eliminado T
 - | * 1d82bba (new-feature) C6: borrada Leche desnatada, cambiado Tomate ecológic
 - |/
 - * e8668ff C5: Merge branch 'bugFix'
 - |\
 - | * 2304881 C4: Cambiada Leche por Leche desnatada
 - * | 3e9d29b C3: añadida Fregona
 - |/
 - * 2c57ec4 C2: creado s2
 - * 0cf8731 C1: creado s1
11. 'git merge new-feature '
Auto-merging s1
CONFLICT (content): Merge conflict in s1
Automatic merge failed; fix conflicts and then commit the result.
[Han modificado el mismo archivo por lo que entra en conflicto.]
12. 'git merge --abort'
13. 'gitgraph'
 - * 847d06d (HEAD, master) C7: cambiado Yoghourt griego por Yoghourt y elimina
 - | * 1d82bba (new-feature) C6: borrada Leche desnatada, cambiado Tomate ecol'
 - |/
 - * e8668ff C5: Merge branch 'bugFix'
 - |\
 - | * 2304881 C4: Cambiada Leche por Leche desnatada
 - * | 3e9d29b C3: añadida Fregona
 - |/
 - * 2c57ec4 C2: creado s2
 - * 0cf8731 C1: creado s1
15. 'git add s1'
'git commit -m "C8:..."
16. 'gitgraph'
 - * 48eb9cc (HEAD, master) C8: Merge branch 'new-feature'
 - |\
 - | * 1d82bba (new-feature) C6: borrada Leche desnatada, cambiado Tomate ecol'
 - * | 847d06d C7: cambiado Yoghourt griego por Yoghourt y eliminado Tomate eco
 - |/
 - * e8668ff C5: Merge branch 'bugFix'
 - |\
 - | * 2304881 C4: Cambiada Leche por Leche desnatada
 - * | 3e9d29b C3: añadida Fregona
 - |/
 - * 2c57ec4 C2: creado s2
 - * 0cf8731 C1: creado s1
18. 'git branch -b new-feature'
19. 'gitgraph'

```

* 48eb9cc (HEAD, master) C8: Merge branch 'new-feature'
|\
| * 1d82bba C6: borrada Leche desnatada, cambiado Tomate ecológico por Toma
* | 847d06d C7: cambiado Yoghourt griego por Yoghourt y eliminado Tomate eco
|/
* e8668ff C5: Merge branch 'bugFix'
|\
| * 2304881 C4: Cambiada Leche por Leche desnatada
* | 3e9d29b C3: añadida Fregona
|/
* 2c57ec4 C2: creado s2
* 0cf8731 C1: creado s1

```

HEAD detached:

1. 'git checkout 2c57ec4d2c8c674494736286a1acdba3cf2ad521 '

2. 'git checkout master'
Previous HEAD position was 2c57ec4... C2: creado s2
Switched to branch 'master'

```

'gitgraph'
* 48eb9cc (HEAD, master) C8: Merge branch 'new-feature'
|\
| * 1d82bba C6: borrada Leche desnatada, cambiado Tomate ecológico por Toma
* | 847d06d C7: cambiado Yoghourt griego por Yoghourt y eliminado Tomate eco
|/
* e8668ff C5: Merge branch 'bugFix'
|\
| * 2304881 C4: Cambiada Leche por Leche desnatada
* | 3e9d29b C3: añadida Fregona
|/
* 2c57ec4 C2: creado s2
* 0cf8731 C1: creado s1

```

3. 'git checkout 2c57ec4d2c8c674494736286a1acdba3cf2ad521 '

4. 'git commit -am "C9: Añadido Patatas a s1" '

5. 'gitgraph'

```

* 2b5d3cd (HEAD) C9: Añadido Patatas a s1
| * 48eb9cc (master) C8: Merge branch 'new-feature'
| |\
| | * 1d82bba C6: borrada Leche desnatada, cambiado Tomate ecológico por To
| * | 847d06d C7: cambiado Yoghourt griego por Yoghourt y eliminado Tomate e
| | /
| * e8668ff C5: Merge branch 'bugFix'
| | \
| | * 2304881 C4: Cambiada Leche por Leche desnatada
| | /
| | /
| | /

```

✓

6.

'g

*

1:

1

i

1

1

i

i

i

i

V

13

iv

✱

*

7.

'g'

*

13

1

i

i

i

1

i

1

i

1

14

iv

✱

*

8.

*

*

1

i

i

1

i

1

i

```

|| * 2304881 C4: Cambiada Leche por Leche desnatada
|/
|/
| * 3e9d29b C3: añadida Fregona
|/
* 2c57ec4 C2: creado s2
* 0cf8731 C1: creado s1

```

git stash:

2. 'git checkout master'
4. 'git checkout stage '
error: Your local changes to the following files would be overwritten by checkout:
s1
Please, commit your changes or stash them before you can switch branches.
Aborting
[No te deja cambiarte puesto que he realizado cambios pero no he hecho commit de ellos.]
5. 'git stash '
Saved working directory and index state WIP on master: 48eb9cc C8: Merge branch 'new-feature'
HEAD is now at 48eb9cc C8: Merge branch 'new-feature'

'cat s1 '
Tomate pelado
Tomate orlando
Queso
Yoghourt
Fregona
7. 'git checkout stage '
Switched to branch 'stage'
8. 'git checkout master'
9. 'git stash apply'
'cat s1 '
Tomate pelado
Tomate orlando
Queso
Yoghourt
Fregona
Peras
10. 'git commit -am "C11: añadidas Peras a s1" '
11. ' git stash list'
stash@{0}: WIP on master: 48eb9cc C8: Merge branch 'new-feature'

13. 'git stash list'

stash@{0}: WIP on master: 763c1f5 C11: añadidas Peras a s1

stash@{1}: WIP on master: 48eb9cc C8: Merge branch 'new-feature'

14. 'git stash clear'