

Git avanzado

Jesús Amieiro



QUADRALIA

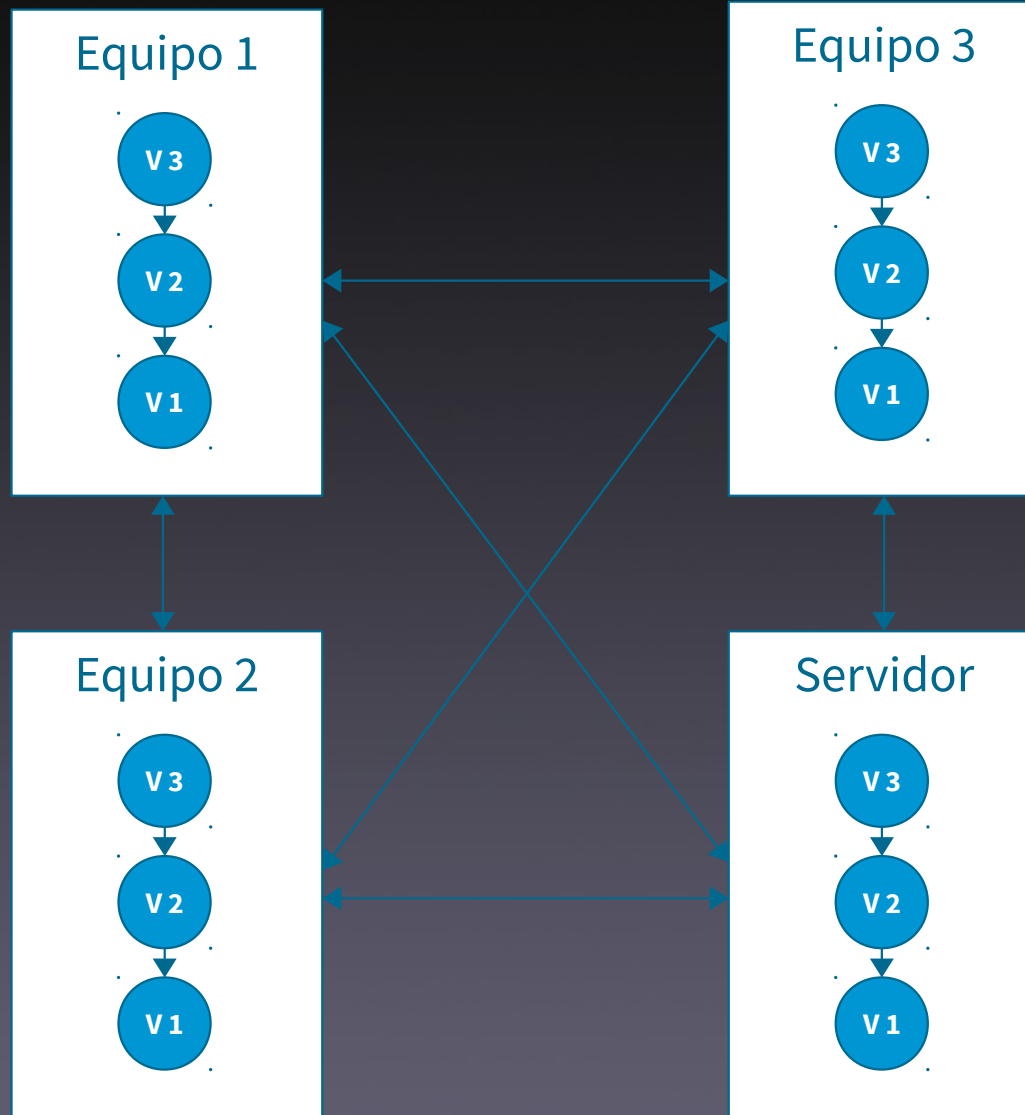
la semana

php.com

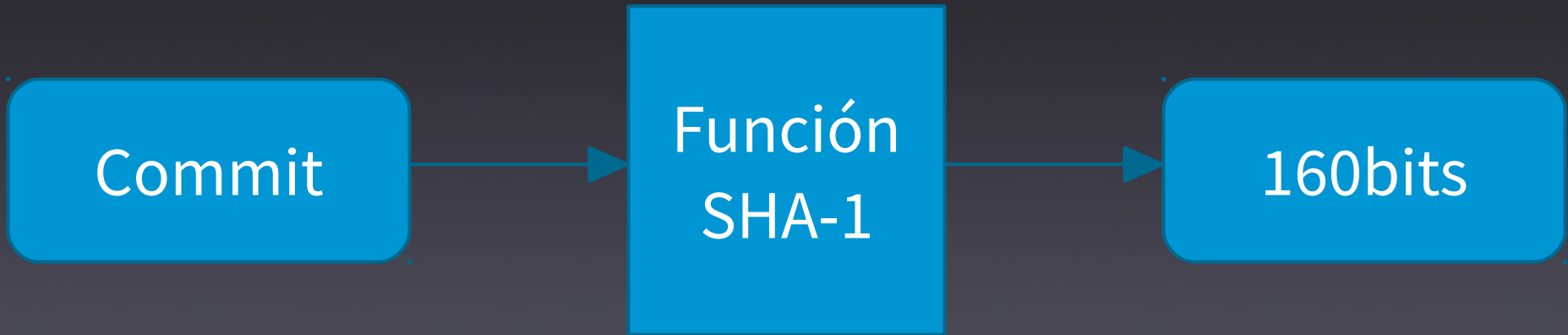
Git avanzado

1. Git
2. Reset
3. Cherry pick
4. Stash
5. Bisect
6. Blame
7. Rebase
8. Squash
9. Reflog

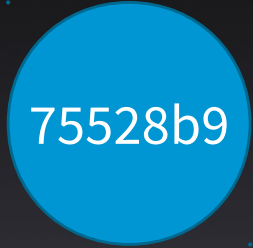
1 - Git



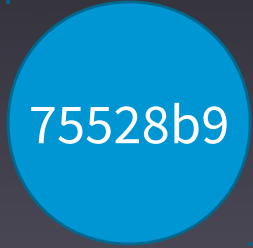
Commit



Commit 1



Commit 1



Commit 2



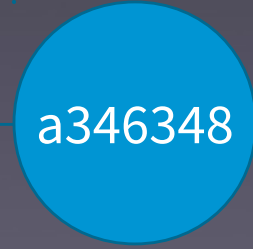
Commit 1

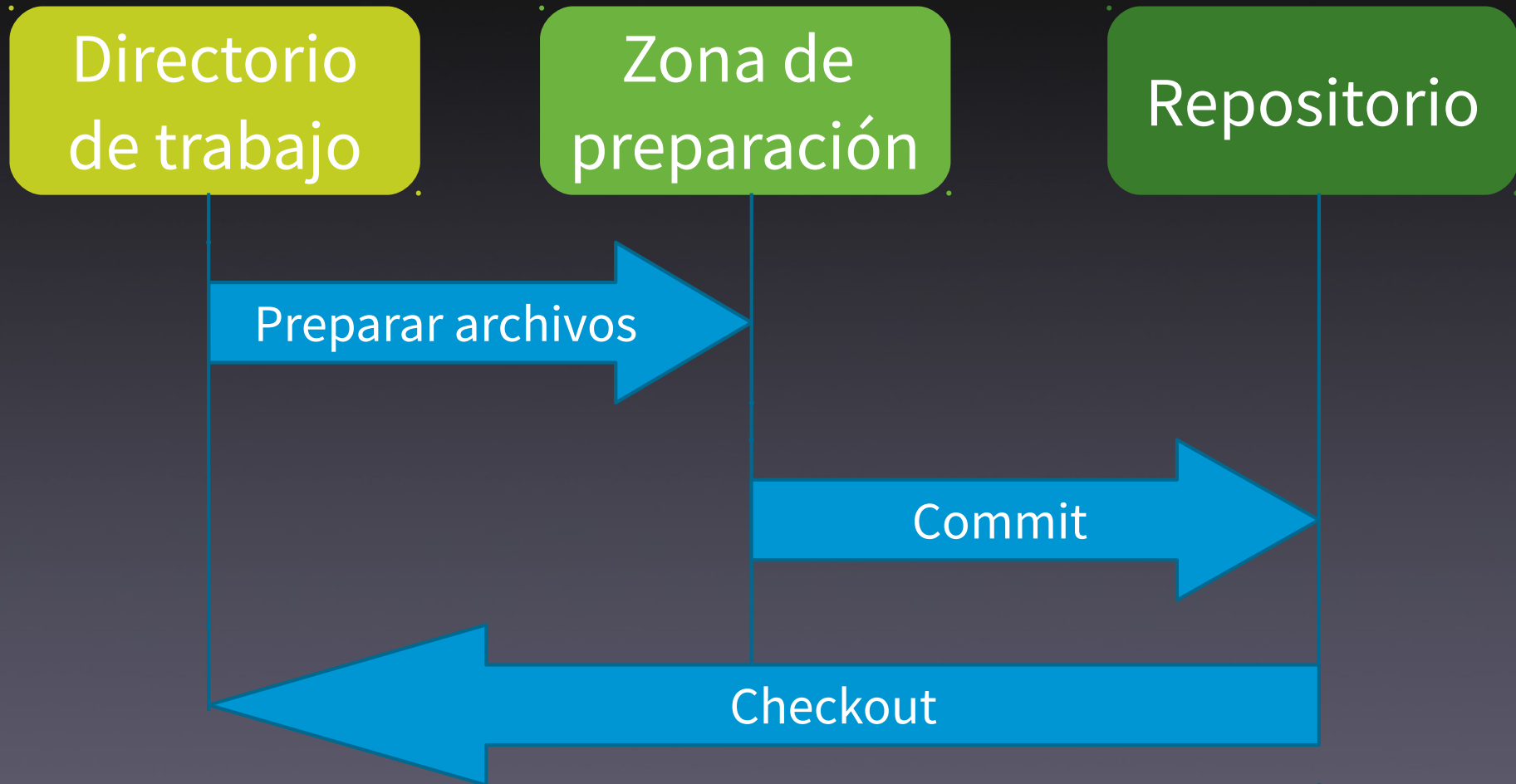


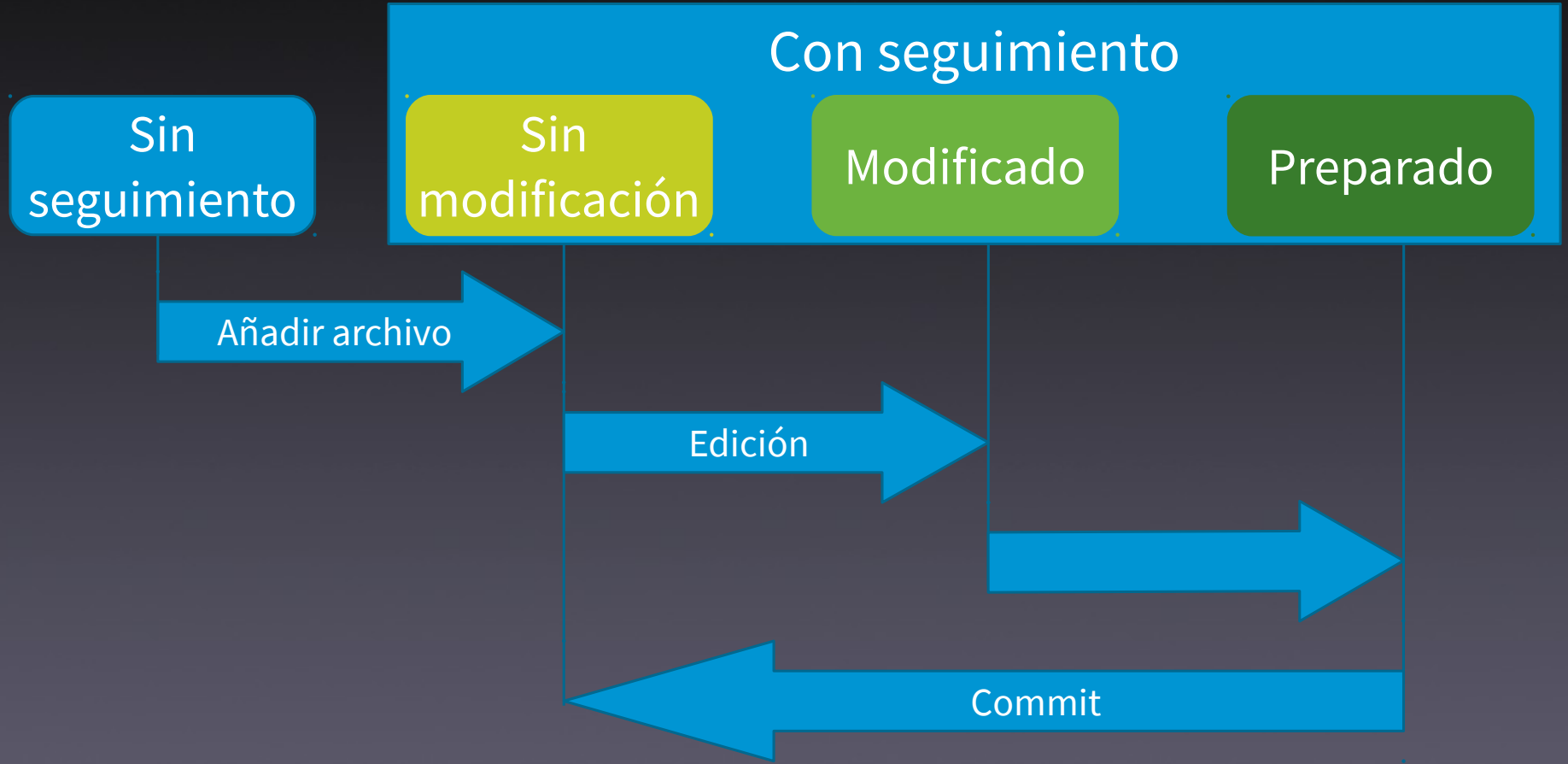
Commit 2

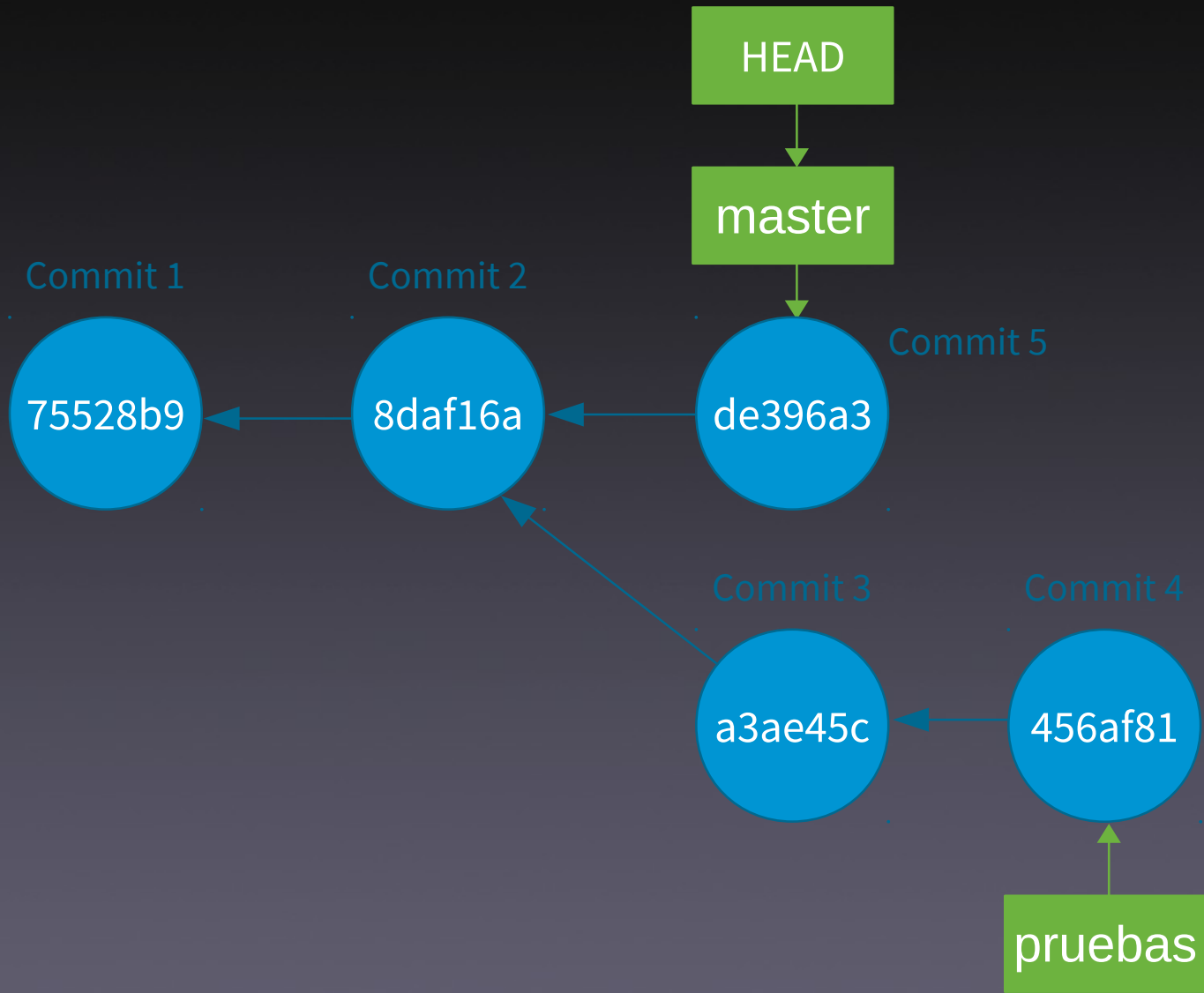


Commit 3









Repositorios remotos

- git clone
- git remote
- git push
- git fetch
- git pull





2 - reset

Directorio
de trabajo

Zona de
preparación

Repositorio

soft

mixed

hard

2.1. reset soft

- * **8e82597** (HEAD -> master) Añado línea 2 al archivo1.txt
- * **4c51d34** Añado línea 1 al archivo1.txt


```
# Hago el soft reset
git reset --soft HEAD~1
```

```
# Compruebo las diferencias entre las zonas
git status
```

```
On branch master
```

```
Changes to be committed:
```

```
(use "git restore --staged <file>..." to unstage)
```

```
    modified:   archivo1.txt
```

```
# Compruebo el contenido de archivo1.txt
```

```
cat archivo1.txt
```

```
Línea 1 de archivo1.txt
```

```
Línea 2 de archivo1.txt
```

```
# Diff entre la zona de trabajo y la de preparación
git diff
```

```
# Diff entre la zona de preparación y el HEAD actual
git diff --staged
```

```
diff --git a/archivo1.txt b/archivo1.txt
```

```
index f3638d1..42e34e6 100644
```

```
--- a/archivo1.txt
```

```
+++ b/archivo1.txt
```

```
@@ -1 +1,2 @@
```

```
 Línea 1 de archivo1.txt
```

```
+Línea 2 de archivo1.txt
```

2.2. reset mixed

```
# Hago el mixed reset
git reset --mixed HEAD~1
```

```
# Compruebo las diferencias entre las zonas
git status
```

On branch master

Changes not staged for commit:

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

(use "git restore <file>..." to discard changes in working directory)

modified: archivo1.txt

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

```
# Compruebo el contenido de archivo1.txt
```

```
cat archivo1.txt
```

```
Línea 1 de archivo1.txt
```

```
Línea 2 de archivo1.txt
```

```
# Diff entre la zona de trabajo y la de preparación
```

```
git diff
```

```
diff --git a/archivo1.txt b/archivo1.txt
```

```
index f3638d1..42e34e6 100644
```

```
--- a/archivo1.txt
```

```
+++ b/archivo1.txt
```

```
@@ -1 +1,2 @@
```

```
 Línea 1 de archivo1.txt
```

```
+Línea 2 de archivo1.txt
```

```
# Diff entre la zona de preparación y el HEAD actual
```

```
git diff --staged
```

2.3. reset hard

```
# Hago el hard reset
git reset --hard HEAD~1
HEAD is now at 877e104 Añado línea 1 al archivo1.txt

# Compruebo las diferencias entre las zonas
git status

nothing to commit, working tree clean

# Compruebo el contenido de archivo1.txt
cat archivo1.txt
Línea 1 de archivo1.txt

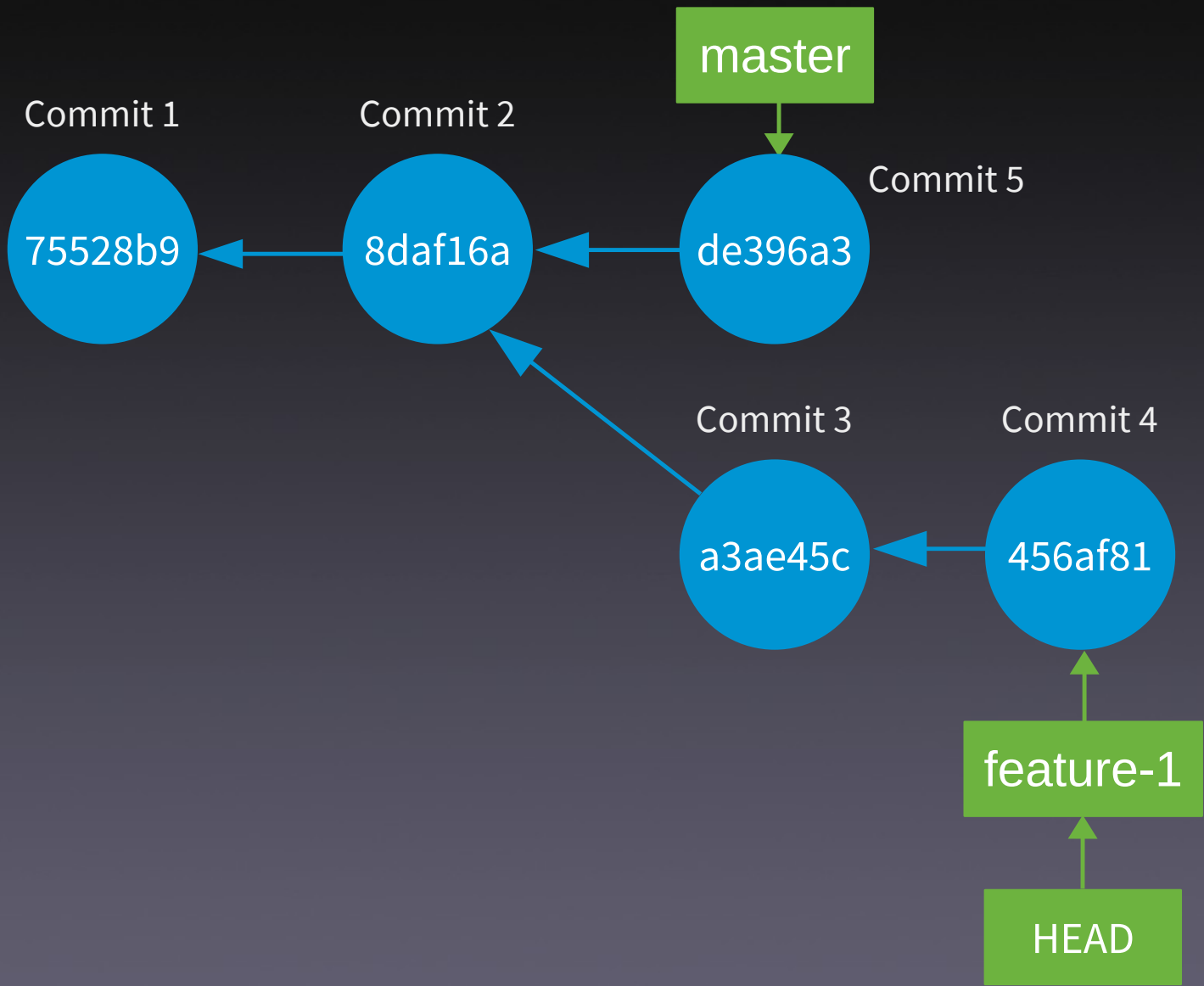
# Diff entre la zona de trabajo y la de preparación
git diff

# Diff entre la zona de preparación y el HEAD actual
git diff --staged
```





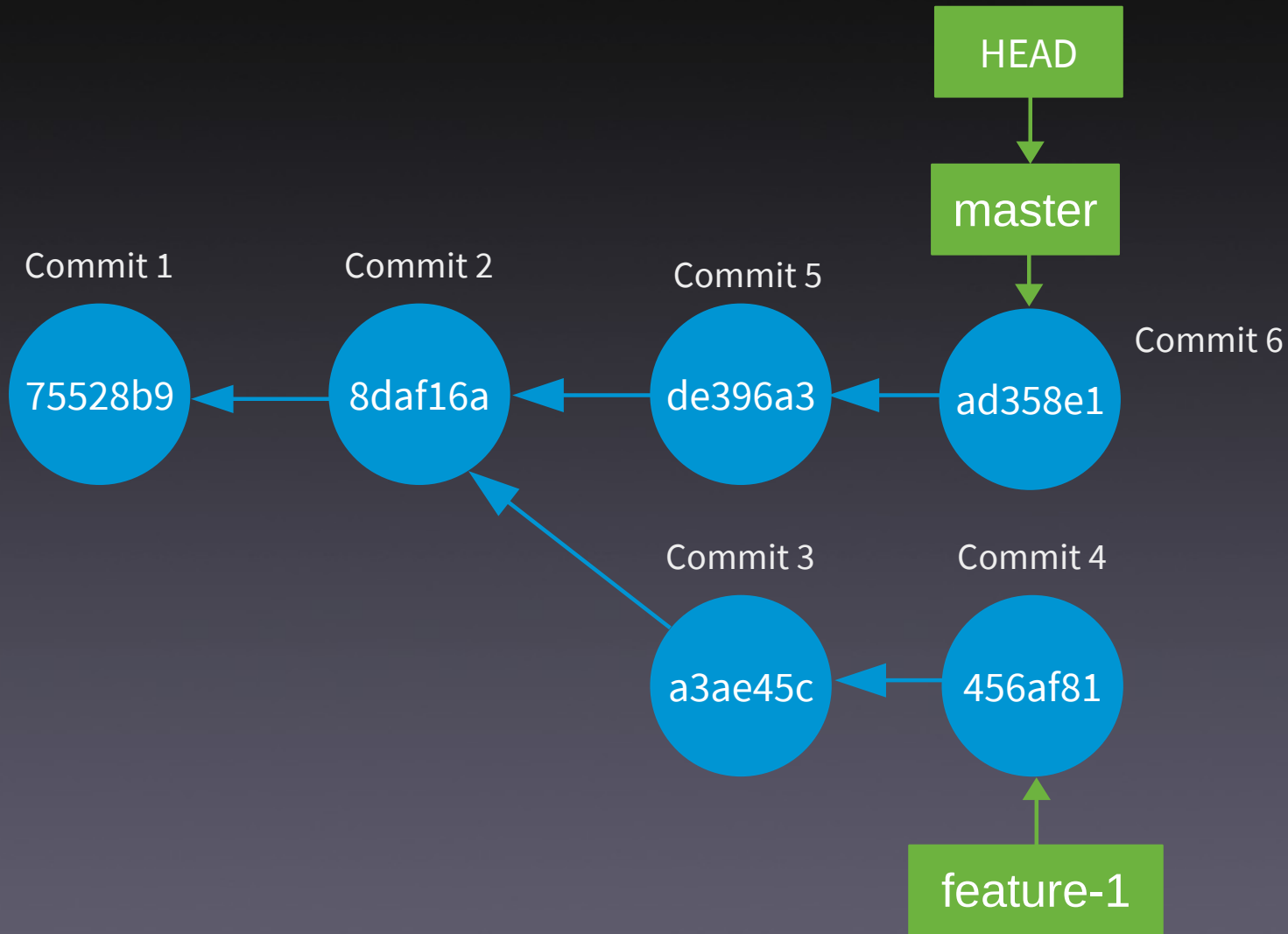
3 - cherry-pick



git cherry-pick

```
$ git checkout master
```

```
$ git cherry-pick 456af81
```



Cherry pick

1. Ejemplo básico

- * 9dbb447 (HEAD -> master) Añado fichero5.txt
 - * 5c22fd9 (funcionalidad-1) Añado fichero4.txt
 - * 1b82298 Añado fichero3.txt
 - * c4a0eb9 Añado .gitignore
-
- * 6435caa Añado archivo2.txt
 - * cdd8261 Añado archivo1.txt

```
# Llevo el antepenúltimo commit de la rama funcionalidad-1  
# a la rama master
```

```
git cherry-pick $(git rev-parse funcionalidad-1~2)
```

```
# Visualizo la estructura del repositorio
```

```
git log --graph --oneline --decorate --all
```

- * 4b1491d (HEAD -> master) Añado .gitignore
- * 9dbb447 Añado fichero5.txt
- | * 5c22fd9 (funcionalidad-1) Añado fichero4.txt
- | * 1b82298 Añado fichero3.txt
- | * c4a0eb9 Añado .gitignore
- | /
- * 6435caa Añado archivo2.txt
- * cdd8261 Añado archivo1.txt

git cherry-pick

- 2 o más commits
- Editar el mensaje de commit: -e
- Aplicar el contenido sin hacer el commit

Cherry pick

2. Aplicar el contenido sin hacer el commit

```
* 339e4ec (HEAD -> master) Añado fichero5.txt
| * d309d2e (funcionalidad-1) Añado fichero4.txt
| * 5e2e675 Añado fichero3.txt
| * d8f66e4 Añado .gitignore
| /
* 993adfa Añado archivo2.txt
* a0b67f7 Añado archivo1.txt
```

```
# Llevo el último y el penúltimo commit de la rama
# funcionalidad-1 a la rama master sin comitearlo
git cherry-pick -n $(git rev-parse funcionalidad-1) \
$(git rev-parse funcionalidad-1~1)
```

```
# Compruebo que los dos archivos
# están listos para ser commiteados
git status
```

On branch master

Changes to be committed:

(use "git restore --staged <file>..." to unstage)

new file: archivo3.txt

new file: archivo4.txt

```
# Elimino archivo3.txt de la zona de preparación
```

```
git restore --staged archivo3.txt
```

```
git status
```

```
On branch master
```

```
Changes to be committed:
```

```
(use "git restore --staged <file>..." to unstage)
```

```
new file:   archivo4.txt
```

```
Untracked files:
```

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

```
archivo3.txt
```

```
# Elimino archivo3.txt
```

```
rm archivo3.txt
```

```
git status
```

On branch master

Changes to be committed:

(use "git restore --staged <file>..." to unstage)

```
new file:   archivo4.txt
```

```
git commit -m "Añado fichero4.txt"
```

```
# Visualizo la estructura del repositorio
```

```
git log --graph --oneline --decorate --all
```

```
* 67c9a1f (HEAD -> master) Añado fichero4.txt
* 339e4ec Añado fichero5.txt
| * d309d2e (funcionalidad-1) Añado fichero4.txt
| * 5e2e675 Añado fichero3.txt
| * d8f66e4 Añado .gitignore
|/
* 993adfa Añado archivo2.txt
* a0b67f7 Añado archivo1.txt
```



A young child with long, wavy blonde hair is shown from the chest up, wearing a white long-sleeved shirt. The child's face is completely obscured by their hands, which are pressed against their eyes and nose. The background is a dark, solid color. The text "4 - stash" is overlaid in white, sans-serif font across the center of the image.

4 - stash


```
$ git log --graph --oneline --decorate --all
* af08e72 (HEAD -> funcionalidad-1) Añado fichero4.txt
* 432f96c Añado fichero3.txt
* e03b5ce Añado .gitignore
* 8968b49 (master) Añado archivo2.txt
* bd75510 Añado archivo1.txt
```

```
$ git status
```

```
On branch funcionalidad-1
```

```
Changes not staged for commit:
```

```
  (use "git add <file>..." to update what will be committed)
```

```
  (use "git restore <file>..." to discard changes in working directory)
```

```
    modified:   archivo4.txt
```

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

```
# Trato de cambiarme a la rama master
```

```
$ git checkout master
```

```
error: Your local changes to the following files would be overwritten by checkout:  
error: 
```

```
    archivo4.txt
```

```
Please commit your changes or stash them before you switch branches.
```

```
Aborting
```

```
# Hago el stash
```

```
$ git stash
```

```
Saved working directory and index state WIP on funcionalidad-1: af08e72 Añadido fichero4.txt
```

```
# Compruebo el estado del repo
```

```
$ git status
```

```
On branch funcionalidad-1
```

```
nothing to commit, working tree clean
```

```
# Compruebo los elementos en el stash
```

```
$ git stash list
```

```
stash@{0}: WIP on funcionalidad-1: af08e72 Añadido fichero4.txt
```

```
# Me cambio a la rama master
```

```
git checkout master
```

```
# Añado 1 fichero y 1 commit a la rama master
```

```
echo "Línea 1 de archivo5.txt" > archivo5.txt
```

```
git add .
```

```
git commit -m "Añado fichero5.txt"
```

```
# Me cambio a la rama funcionalidad-1
```

```
git checkout funcionalidad-1
```

```
# Recupero el elemento {0} del stash
# git stash apply stash@{0}
git stash pop stash@{0}
```

```
$ git stash pop stash@{0}
On branch funcionalidad-1
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   archivo4.txt

no changes added to commit (use "git add" and/or "git commit -a")
Dropped stash@{0} (93af03cc937ee9b742a19a88eff84bd83e7c274b)
```



A close-up photograph of a hard-boiled egg, cracked open, resting on a silver fork. The egg is split into two halves, with the bright yellow yolk exposed. The egg white is solid and white. The background is dark, making the egg and the silver fork stand out. The text "5 - bisect" is overlaid in white, sans-serif font across the center of the image.

5 - bisect

-3



```
# Clono el repositorio de ejemplo
```

```
$ git clone https://github.com/bradleyboy/bisectercise.git
```

```
Cloning into 'bisectercise'...
```

```
remote: Enumerating objects: 4, done.
```

```
remote: Counting objects: 100% (4/4), done.
```

```
remote: Compressing objects: 100% (4/4), done.
```

```
Receiremote: Total 308 (delta 0), reused 0 (delta 0), pack-reused 304vi
```

```
Receiving objects: 67% (207/308)
```

```
Receiving objects: 100% (308/308), 32.17 KiB | 1.69 MiB/s, done.
```

```
Resolving deltas: 100% (97/97), done.
```

```
# Accedo al directorio
```

```
$ cd bisectercise
```

```
# Primer commit del repo
```

```
$ git rev-list --max-parents=0 HEAD
```

```
4d83cfcbaef648345571d77db867b6f9e4146ba7
```

```
# Inicio el bisect
```

```
# git bisect start HEAD 4d83cf
```

```
$ git bisect start
```

```
$ git bisect bad HEAD
```

```
$ git bisect good 4d83cf
```

```
Bisecting: 50 revisions left to test after this (roughly 6 steps)
```

```
[96c49431eec7982ba36d07573d939db842ef3198] commit 52
```

```
$ git bisect good
```

```
Bisecting: 25 revisions left to test after this (roughly 5 steps)
```

```
[27c58746f430da1374410eab456c559c14599090] commit 77
```

```
$ git bisect bad
```

```
Bisecting: 12 revisions left to test after this (roughly 4 steps)
```

```
[b47892adec22ee3b0330aff37cbc5e695dfb99d6] commit 64
```

```
$ git bisect bad
```

```
Bisecting: 5 revisions left to test after this (roughly 3 steps)
```

```
[98acf91bfbf47a63ca5be01043b4e728bbee35ed] commit 58
```

```
$ git bisect good
```

```
b47892adec22ee3b0330aff37cbc5e695dfb99d6 is the first bad commit
```

```
commit b47892adec22ee3b0330aff37cbc5e695dfb99d6
```

```
Author: Brad Daily <bdaily@rsglab.com>
```

```
Date: Sat Dec 22 13:41:36 2018 -0500
```

```
commit 64
```

```
index.html | 6 +++---
```

```
1 file changed, 3 insertions(+), 3 deletions(-)
```

Muestro información del commit problemático

```
$ git show b47892a
```

```
commit b47892adec22ee3b0330aff37cbc5e695dfb99d6 (refs/bisect/bad)
```

```
Author: Brad Daily <bdaily@rsglab.com>
```

```
Date: Sat Dec 22 13:41:36 2018 -0500
```

```
commit 64
```

```
diff --git a/index.html b/index.html
```

```
index b10625e..43a01ab 100644
```

```
--- a/index.html
```

```
+++ b/index.html
```

```
@@ -67,12 +67,12 @@
```

```
<div id="number">0</div>
```

```
<div class="actions">
```

```
- <button data-action="dec">-</button>
```

```
- <button data-action="inc">+</button>
```

```
+ <button data-action="dc">-</button>
```

```
+ <button data-action="ic">+</button>
```

```
</div>
```

```
</main>
```

```
- <footer>Version: 63</footer>
```

```
+ <footer>Version: 64</footer>
```

```
<script>
```

```
(( ) => {
```

```
# Salgo del bisect y vuelvo
```

```
# al HEAD original
```

```
$ git bisect reset
```

```
Previous HEAD position was 4437723 commit 63
```

```
Switched to branch 'master'
```

```
Your branch is up to date with 'origin/master'.
```



A grayscale image of a hand with the index finger pointing towards the text. The hand is slightly out of focus, and the background is a dark gradient.

6 - blame


```
$ git blame index.html
```

```
^4d83cfc (Brad Daily 2018-12-22 12:52:54 -0500 1) <!DOCTYPE html>
^4d83cfc (Brad Daily 2018-12-22 12:52:54 -0500 2) <html lang="en">
^4d83cfc (Brad Daily 2018-12-22 12:52:54 -0500 3)   <head>
^4d83cfc (Brad Daily 2018-12-22 12:52:54 -0500 4)     <meta charset="UTF-8" />
^4d83cfc (Brad Daily 2018-12-22 12:52:54 -0500 5)     <meta name="viewport" content="width=device-width, initial-scale=1.0" />
^4d83cfc (Brad Daily 2018-12-22 12:52:54 -0500 6)     <meta http-equiv="X-UA-Compatible" content="ie=edge" />
^4d83cfc (Brad Daily 2018-12-22 12:52:54 -0500 7)     <title>bisectercise</title>
^4d83cfc (Brad Daily 2018-12-22 12:52:54 -0500 8)
^4d83cfc (Brad Daily 2018-12-22 12:52:54 -0500 9)     <style>
^4d83cfc (Brad Daily 2018-12-22 12:52:54 -0500 10)      * {
^4d83cfc (Brad Daily 2018-12-22 12:52:54 -0500 11)        box-sizing: border-box;
^4d83cfc (Brad Daily 2018-12-22 12:52:54 -0500 12)      }
^4d83cfc (Brad Daily 2018-12-22 12:52:54 -0500 13)      html,
^4d83cfc (Brad Daily 2018-12-22 12:52:54 -0500 14)      body {
^4d83cfc (Brad Daily 2018-12-22 12:52:54 -0500 15)        height: 100vh;
^4d83cfc (Brad Daily 2018-12-22 12:52:54 -0500 16)        overflow: hidden;
^4d83cfc (Brad Daily 2018-12-22 12:52:54 -0500 17)      }
^4d83cfc (Brad Daily 2018-12-22 12:52:54 -0500 18)
^4d83cfc (Brad Daily 2018-12-22 12:52:54 -0500 19)      body {
^4d83cfc (Brad Daily 2018-12-22 12:52:54 -0500 20)        display: flex;
^4d83cfc (Brad Daily 2018-12-22 12:52:54 -0500 21)        flex-flow: column;
^4d83cfc (Brad Daily 2018-12-22 12:52:54 -0500 22)        align-items: center;
^4d83cfc (Brad Daily 2018-12-22 12:52:54 -0500 23)        justify-content: center;
^4d83cfc (Brad Daily 2018-12-22 12:52:54 -0500 24)        font-size: 180px;
^4d83cfc (Brad Daily 2018-12-22 12:52:54 -0500 25)        font-family: Arial, Helvetica, sans-serif;
^4d83cfc (Brad Daily 2018-12-22 12:52:54 -0500 26)        color: #222;
^4d83cfc (Brad Daily 2018-12-22 12:52:54 -0500 27)        margin: 0;
^4d83cfc (Brad Daily 2018-12-22 12:52:54 -0500 28)      }
^4d83cfc (Brad Daily 2018-12-22 12:52:54 -0500 29)
^4d83cfc (Brad Daily 2018-12-22 12:52:54 -0500 30)      main {
^4d83cfc (Brad Daily 2018-12-22 12:52:54 -0500 31)        display: flex;
```

Muestro información del commit problemático

git show b47892a

commit b47892adec22ee3b0330aff37cbc5e695dfb99d6 (refs/bisect/bad)

Author: Brad Daily <bdaily@rsglab.com>

Date: Sat Dec 22 13:41:36 2018 -0500

commit 64

diff --git a/index.html b/index.html

index b10625e..43a01ab 100644

--- a/index.html

+++ b/index.html

@@ -67,12 +67,12 @@

<div id="number">0</div>

<div class="actions">

- <button data-action="dec">-</button>

- <button data-action="inc">+</button>

+ <button data-action="dc">-</button>

+ <button data-action="ic">+</button>

</div>

</main>

- <footer>Version: 63</footer>

+ <footer>Version: 64</footer>

<script>

(() => {

```
$ git blame -L 67,76 index.html
```

```
^4d83cfc (Brad Daily 2018-12-22 12:52:54 -0500 67)
```

```
^4d83cfc (Brad Daily 2018-12-22 12:52:54 -0500 68)
```

```
^4d83cfc (Brad Daily 2018-12-22 12:52:54 -0500 69)
```

```
b47892ad (Brad Daily 2018-12-22 13:41:36 -0500 70)
```

```
b47892ad (Brad Daily 2018-12-22 13:41:36 -0500 71)
```

```
^4d83cfc (Brad Daily 2018-12-22 12:52:54 -0500 72)
```

```
^4d83cfc (Brad Daily 2018-12-22 12:52:54 -0500 73)
```

```
^4d83cfc (Brad Daily 2018-12-22 12:52:54 -0500 74)
```

```
7808b094 (Brad Daily 2018-12-22 13:41:54 -0500 75)
```

```
^4d83cfc (Brad Daily 2018-12-22 12:52:54 -0500 76)
```

```
<div id="number">0</div>
```

```
<div class="actions">
```

```
  <button data-action="dc">-</button>
```

```
  <button data-action="ic">+</button>
```

```
</div>
```

```
</main>
```

```
<footer>Version: 100</footer>
```





7 - rebase

7.1 – merge fast forward

```
$ git log --graph --oneline --decorate --all
* c3168b2 (funcionalidad-1) Añado fichero4.txt
* 3d95d66 Añado fichero3.txt
* a055fa7 (HEAD -> master) Añado archivo2.txt
* 0123430 Añado archivo1.txt
```

```
# Fusiono ambas ramas
```

```
$ git merge funcionalidad-1
```

```
Updating a055fa7..c3168b2
```

```
Fast-forward
```

```
archivo3.txt | 1 +
```

```
archivo4.txt | 1 +
```

```
2 files changed, 2 insertions(+)
```

```
create mode 100644 archivo3.txt
```

```
create mode 100644 archivo4.txt
```

```
$ git log --graph --oneline --decorate --all
* c3168b2 (HEAD -> master, funcionalidad-1) Añado fichero4.txt
* 3d95d66 Añado fichero3.txt
* a055fa7 Añado archivo2.txt
* 0123430 Añado archivo1.txt
```


7.2 – merge recursive

```
$ git log --graph --oneline --decorate --all
* be6e94c (HEAD -> master) Añado fichero5.txt
| * a47b79c (funcionalidad-1) Añado fichero4.txt
| * 9020720 Añado fichero3.txt
| /
* c77326f Añado archivo2.txt
* 55664b8 Añado archivo1.txt
```

```
# Fusiono ambas ramas
```

```
$ git merge funcionalidad-1
```

```
Merge made by the 'recursive' strategy.
```

```
archivo3.txt | 1 +
```

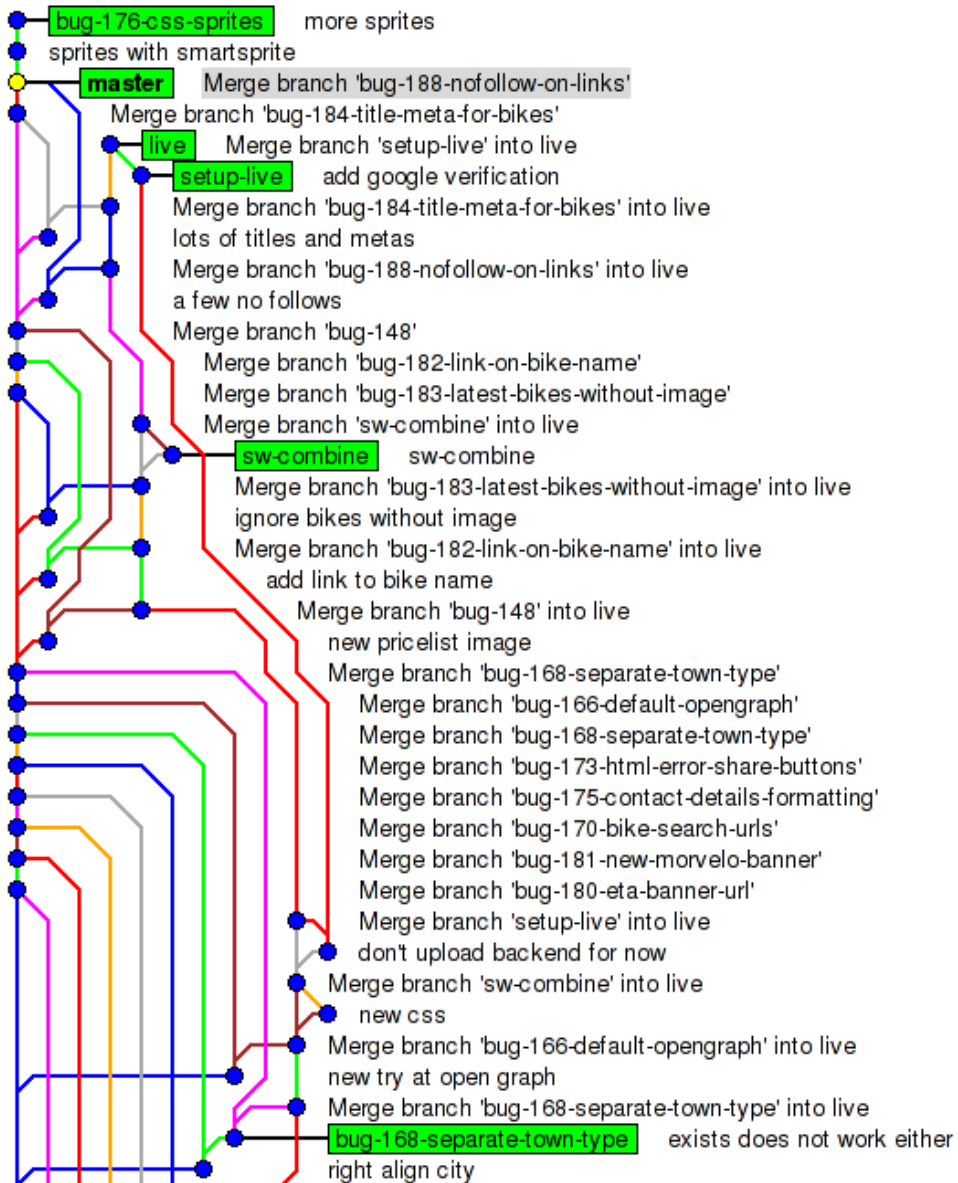
```
archivo4.txt | 1 +
```

```
2 files changed, 2 insertions(+)
```

```
create mode 100644 archivo3.txt
```

```
create mode 100644 archivo4.txt
```

```
$ git log --graph --oneline --decorate --all
* 246611f (HEAD -> master) Merge branch 'funcionalidad-1'
| \
| * 0c3538e (funcionalidad-1) Añado fichero4.txt
| * 823c8cb Añado fichero3.txt
* | 5547697 Añado fichero5.txt
| /
* a633f25 Añado archivo2.txt
* 1dc3d46 Añado archivo1.txt
```



7.3 – rebase

```
$ git log --graph --oneline --decorate --all
* c105443 (HEAD -> master) Añado fichero5.txt
| * 08cc1bd (funcionalidad-1) Añado fichero4.txt
| * 6bd2473 Añado fichero3.txt
| /
* 1911b57 Añado archivo2.txt
* 4ebfe55 Añado archivo1.txt
```

```
# Me cambio a la rama funcionalidad-1
```

```
$ git checkout funcionalidad-1
```

```
Switched to branch 'funcionalidad-1'
```

```
# Hago el rebase en la rama master
```

```
$ git rebase master
```

```
First, rewinding head to replay your work on top of it...
```

```
Applying: Añado fichero3.txt
```

```
Applying: Añado fichero4.txt
```



```
$ git log --graph --oneline --decorate --all
* b52bb01 (HEAD -> funcionalidad-1) Añado fichero4.txt
* a16130a Añado fichero3.txt
* c105443 (master) Añado fichero5.txt
* 1911b57 Añado archivo2.txt
* 4ebfe55 Añado archivo1.txt
```

```
$ git checkout master
```

```
# Hago un merge fast-forward
```

```
$ git merge funcionalidad-1
```

```
Updating c105443..b52bb01
```

```
Fast-forward
```

```
archivo3.txt | 1 +
```

```
archivo4.txt | 1 +
```

```
2 files changed, 2 insertions(+)
```

```
create mode 100644 archivo3.txt
```

```
create mode 100644 archivo4.txt
```

```
$ git log --graph --oneline --decorate --all
* b52bb01 (HEAD -> master, funcionalidad-1) Añado fichero4.txt
* a16130a Añado fichero3.txt
* c105443 Añado fichero5.txt
* 1911b57 Añado archivo2.txt
* 4ebfe55 Añado archivo1.txt
```



A close-up, top-down view of a large number of small, round, yellow-orange pumpkins. The pumpkins are densely packed together, filling the entire frame. They have a ribbed texture and short, green stems. The lighting is even, highlighting the vibrant colors of the pumpkins.

8 - squash

```
$ git log --graph --oneline --decorate --all
* 91c3906 (master) Añado fichero5.txt
| * 294f41d (HEAD -> funcionalidad-1) Añado .gitignore
| * 290a0eb Añado fichero4.txt
| * f2501be Inicio la funcionalidad 1. Añado fichero3.txt
| /
* 2417a66 Añado archivo2.txt
* 7c160db Añado archivo1.txt
```

```
# Inicio el rebase interactivo en la  
#rama funcionalidad-1  
git rebase -i HEAD~3
```

```
pick f2501be Inicio la funcionalidad 1. Añado fichero3.txt
```

```
pick 290a0eb Añado fichero4.txt
```

```
pick 294f41d Añado .gitignore
```

```
█
```

```
# Rebase 2417a66..294f41d onto 2417a66 (3 commands)
```

```
#
```

```
# Commands:
```

```
# p, pick <commit> = use commit
```

```
# r, reword <commit> = use commit, but edit the commit message
```

```
# e, edit <commit> = use commit, but stop for amending
```

```
# s, squash <commit> = use commit, but meld into previous commit
```

```
# f, fixup <commit> = like "squash", but discard this commit's log message
```

```
# x, exec <command> = run command (the rest of the line) using shell
```

```
# b, break = stop here (continue rebase later with 'git rebase --continue')
```

```
# d, drop <commit> = remove commit
```

```
# l, label <label> = label current HEAD with a name
```

```
# t, reset <label> = reset HEAD to a label
```

```
reword f2501be Inicio la funcionalidad 1. Añado fichero3.txt
```

```
fixup 290a0eb Añado fichero4.txt
```

```
fixup 294f41d Añado .gitignore
```

```
□
```

```
# Rebase 2417a66..294f41d onto 2417a66 (3 commands)
```

```
#
```

```
# Commands:
```

```
# p, pick <commit> = use commit
```

```
# r, reword <commit> = use commit, but edit the commit message
```

```
# e, edit <commit> = use commit, but stop for amending
```

```
# s, squash <commit> = use commit, but meld into previous commit
```

```
# f, fixup <commit> = like "squash", but discard this commit's log message
```


Inicio la funcionalidad 1. Añado fichero3.txt

```
# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
#
# Date:      Wed Nov 20 07:31:00 2019 +0100
#
# interactive rebase in progress; onto 2417a66
# Last command done (1 command done):
#   reword f2501be Inicio la funcionalidad 1. Añado fichero3.txt
# Next commands to do (2 remaining commands):
#   fixup 290a0eb Añado fichero4.txt
#   fixup 294f41d Añado .gitignore
# You are currently editing a commit while rebasing branch 'funcion
```

Añado la funcionalidad 1

□

Añado fichero3.txt

Añado fichero4.txt

Añado .gitignore

Please enter the commit message for your changes. Lines starting
with '#' will be ignored, and an empty message aborts the commit.

```
$ git log --graph --oneline --decorate --all
* 605dbc4 (HEAD -> funcionalidad-1) Añado la funcionalidad 1
| * 6d42979 (master) Añado fichero5.txt
|/
* 7b94918 Añado archivo2.txt
* 8f0e307 Añado archivo1.txt
```

```
$ git rebase master
```

```
First, rewinding head to replay your work on top of it...
```

```
Applying: Añado la funcionalidad 1
```

```
$ git log --graph --oneline --decorate --all
```

```
* bfbac5a (HEAD -> funcionalidad-1) Añado la funcionalidad 1
```

```
* 6d42979 (master) Añado fichero5.txt
```

```
* 7b94918 Añado archivo2.txt
```

```
* 8f0e307 Añado archivo1.txt
```

```
$ git checkout master
```

```
$ git merge funcionalidad-1
```

```
Updating 6d42979..bfbac5a
```

```
Fast-forward
```

```
.gitignore | 1 +
```

```
archivo3.txt | 1 +
```

```
archivo4.txt | 1 +
```

```
3 files changed, 3 insertions(+)
```

```
create mode 100644 .gitignore
```

```
create mode 100644 archivo3.txt
```

```
create mode 100644 archivo4.txt
```

```
# Borro la rama funcionalidad-1
```

```
$ git branch -d funcionalidad-1
```

```
$ git log --graph --oneline --decorate --all
```

```
* bfbac5a (HEAD -> master) Añado la funcionalidad 1
```

```
* 6d42979 Añado fichero5.txt
```

```
* 7b94918 Añado archivo2.txt
```

```
* 8f0e307 Añado archivo1.txt
```





9 - reflog

```
$ git log --graph --oneline --decorate --all
* 71e4051 (master) Añado fichero5.txt
| * e22c5f8 (HEAD -> funcionalidad-1) Añado .gitignore
| * 2ef125e Añado fichero4.txt
| * 93b297b Inicio la funcionalidad 1. Añado fichero3.txt
|/
* 6b262c9 Añado archivo2.txt
* fddcc96 Añado archivo1.txt
```



```
# Borro los 3 últimos commits
```

```
$ git reset --hard HEAD~3
```

```
HEAD is now at 6b262c9 Añado archivo2.txt
```

```
$ git log --graph --oneline --decorate --all
```

```
* 71e4051 (master) Añado fichero5.txt
```

```
* 6b262c9 (HEAD -> funcionalidad-1) Añado archivo2.txt
```

```
* fddcc96 Añado archivo1.txt
```

```
$ git reflog
```

```
6b262c9 (HEAD -> funcionalidad-1) HEAD@{0}: reset: moving to HEAD~3
```

```
e22c5f8 HEAD@{1}: checkout: moving from master to funcionalidad-1
```

```
71e4051 (master) HEAD@{2}: commit: Añado fichero5.txt
```

```
6b262c9 (HEAD -> funcionalidad-1) HEAD@{3}: checkout: moving from funcionalidad-1 to master
```

```
e22c5f8 HEAD@{4}: commit: Añado .gitignore
```

```
2ef125e HEAD@{5}: commit: Añado fichero4.txt
```

```
93b297b HEAD@{6}: commit: Inicio la funcionalidad 1. Añado fichero3.txt
```

```
6b262c9 (HEAD -> funcionalidad-1) HEAD@{7}: checkout: moving from master to funcionalidad-1
```

```
6b262c9 (HEAD -> funcionalidad-1) HEAD@{8}: commit: Añado archivo2.txt
```

```
fddcc96 HEAD@{9}: commit (initial): Añado archivo1.txt
```

```
$ git reset --hard HEAD@{1}
HEAD is now at 95a3874 Añado .gitignore
```

```
$ git log --graph --oneline --decorate --all
* 7491303 (master) Añado fichero5.txt
| * 95a3874 (HEAD -> funcionalidad-1) Añado .gitignore
| * 74d11a2 Añado fichero4.txt
| * 7440492 Inicio la funcionalidad 1. Añado fichero3.txt
| /
* e1a62be Añado archivo2.txt
* 8d11797 Añado archivo1.txt
```



Preguntas