



git

TEMA 04

CONTROL DE VERSIONS: GIT *(PART III)*

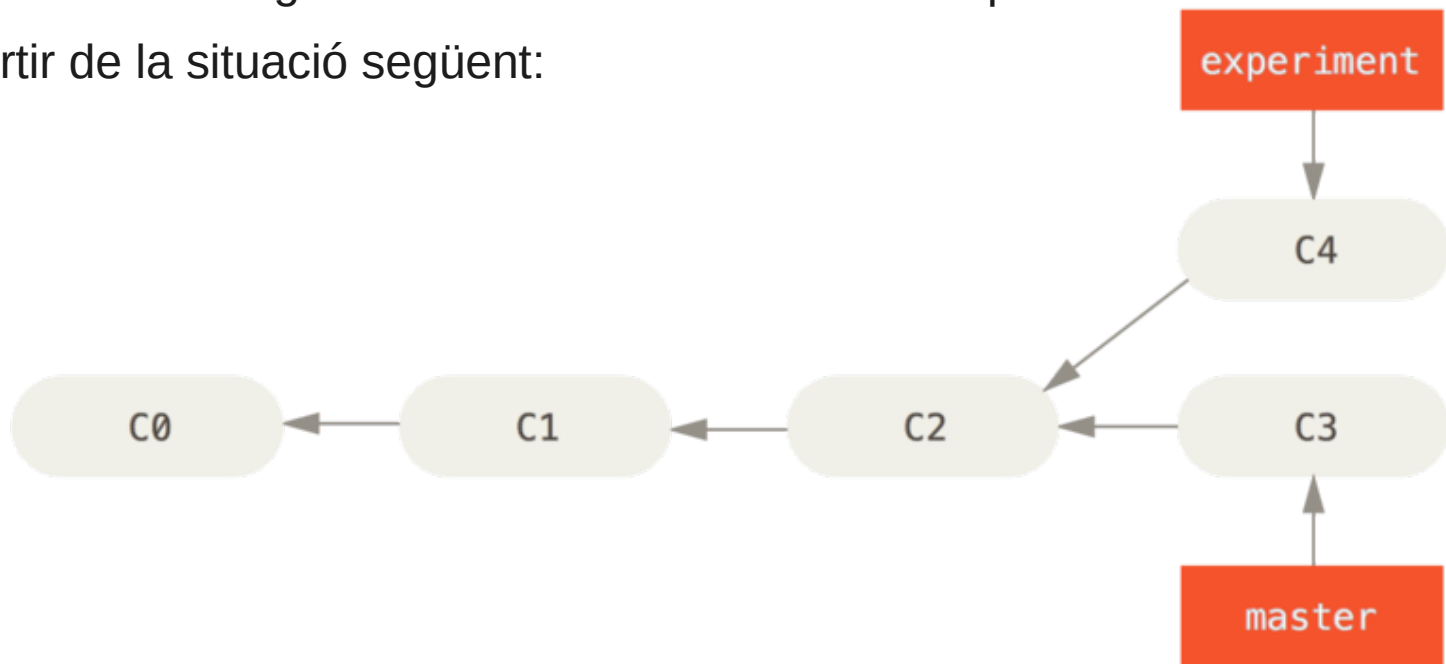
1) rebase

2) Altres usos de rebase

DESPLEGAMENT D'APLICACIONS WEB

TEMA 04: CONTROL DE VERSIONS: GIT (PART III): rebase

- **rebase**: altra manera d'integrar canvis fets en diferents branques.
- **Exemple** a partir de la situació següent:

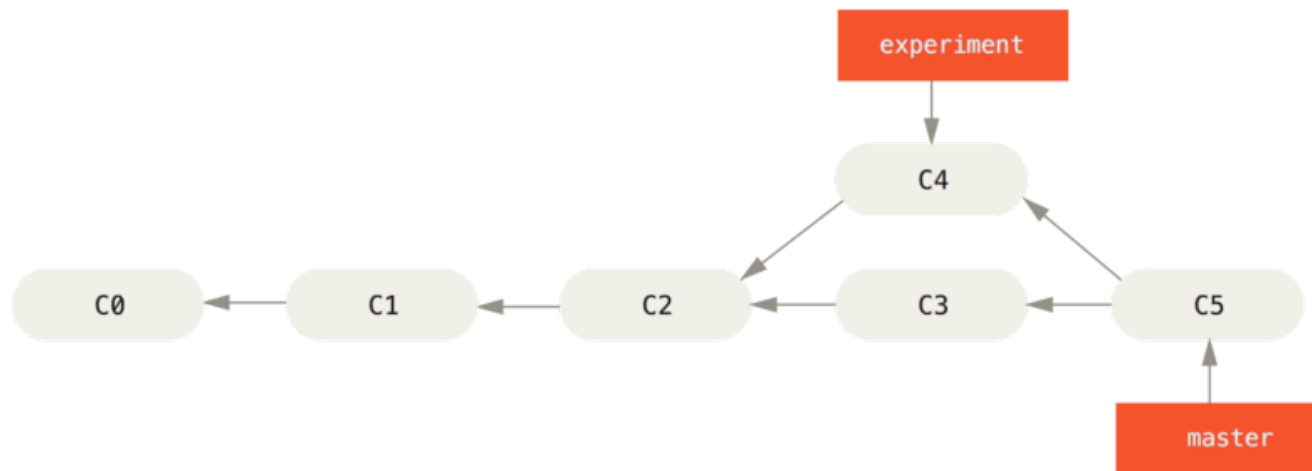


DESPLEGAMENT D'APLICACIONS WEB

TEMA 04: CONTROL DE VERSIONS: GIT (PART III): rebase

- Forma més fàcil d'integrar les branques: **merge**.
- Fusió a tres bandes: dos últims **commits** en cada branca (C3 i C4) i el més recent ancestre comú de les dues (C2), creant un nou **commit** (C5):
- **Hem d'estar en la branca a la que volem fusionar**
→ **master**

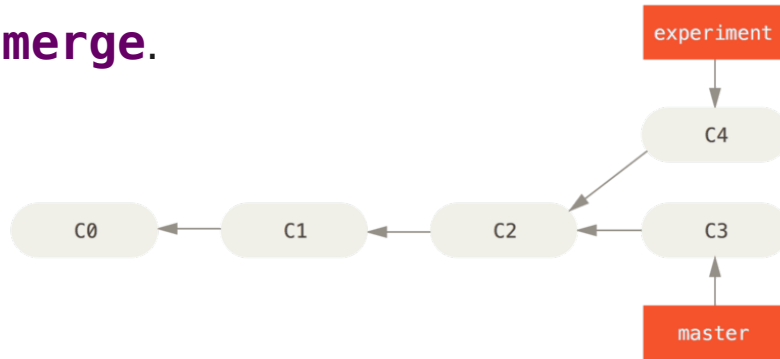
git checkout master
git merge experiment



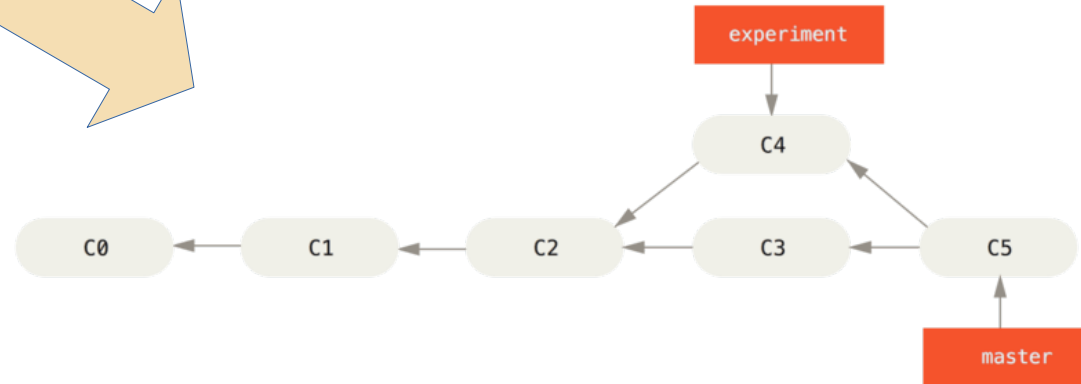
DESPLEGAMENT D'APLICACIONS WEB

TEMA 04: CONTROL DE VERSIONS: GIT (PART III): rebase

merge.



git checkout master
git merge experiment

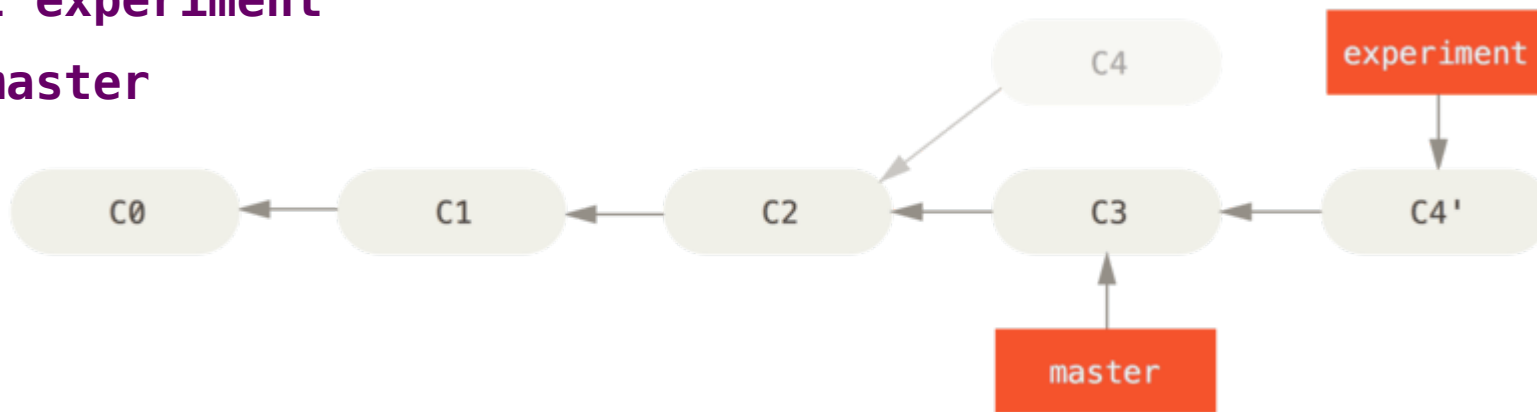


DESPLEGAMENT D'APLICACIONS WEB

TEMA 04: CONTROL DE VERSIONS: GIT (PART III): rebase

- Altra manera d'unir les branques: **rebase**.
- Prenem el **commit** que es va introduir en C4 i tornem a aplicar-lo a C3.
- **Hem d'estar en la branca que volem fusionar**
 - **experiment**

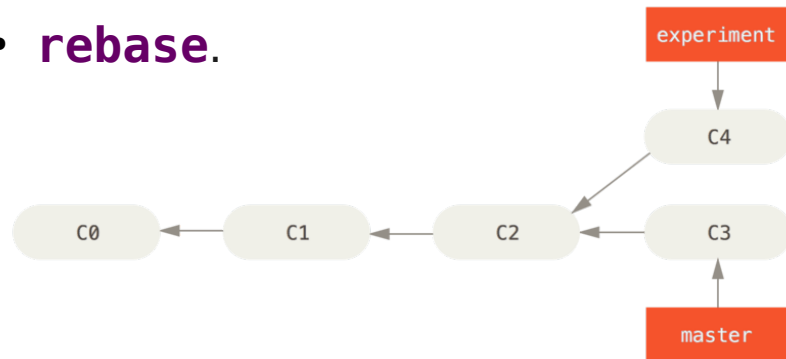
git checkout experiment
git rebase master



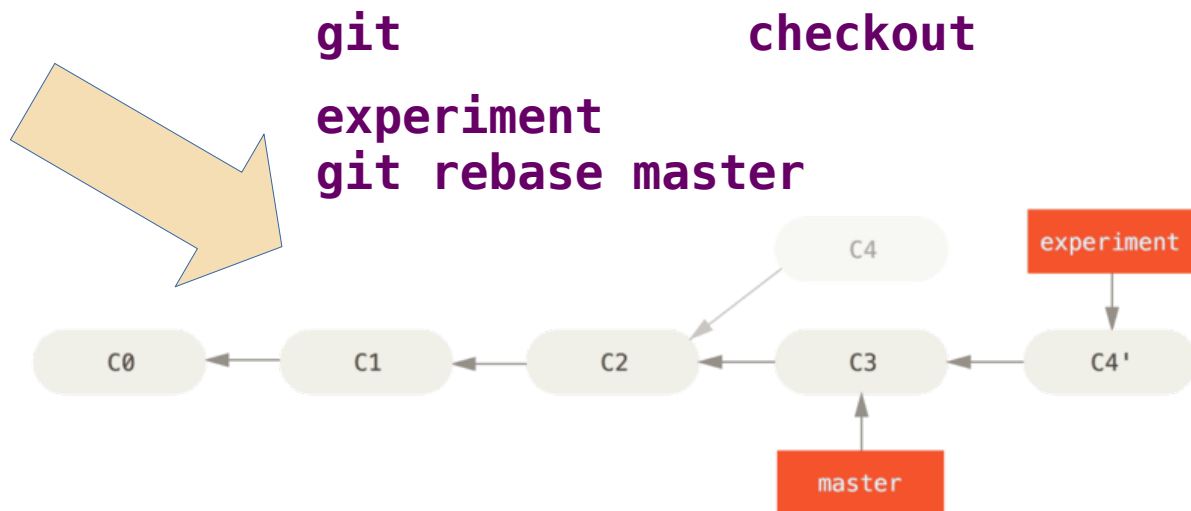
DESPLEGAMENT D'APLICACIONS WEB

TEMA 04: CONTROL DE VERSIONS: GIT (PART III): rebase

- **rebase.**



- Amb **rebase**, es prenen tots els **commits** que s'han fet en una branca i es tornen a reproduir en una altra.



DESPLEGAMENT D'APLICACIONS WEB

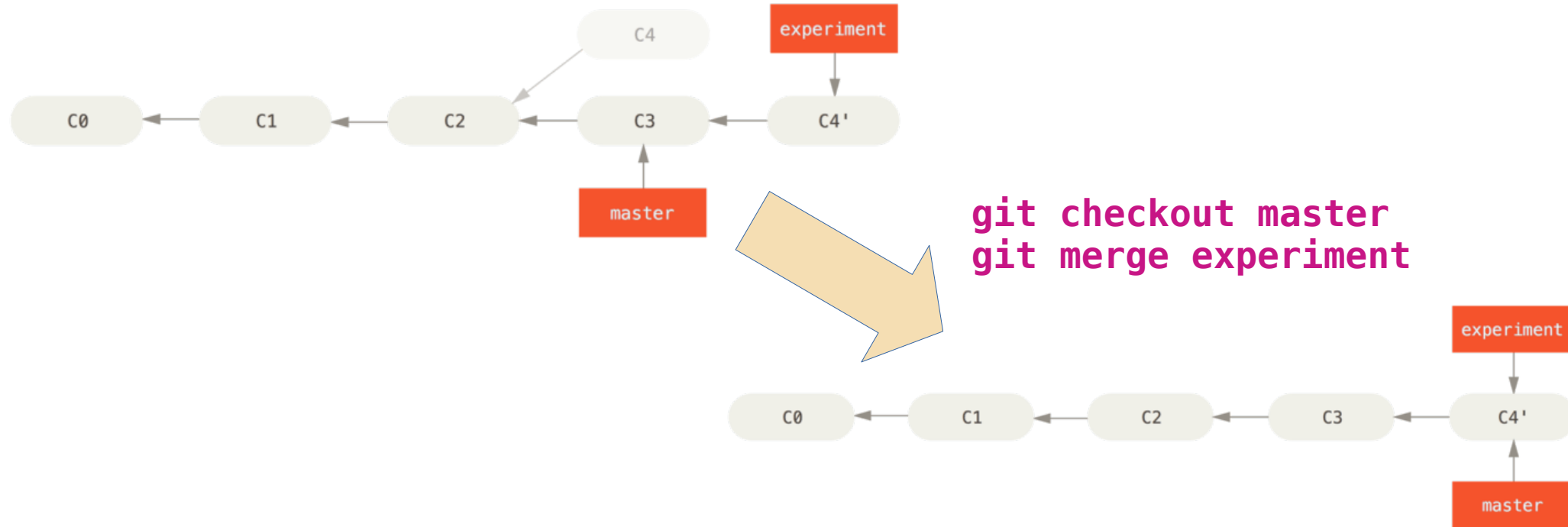
TEMA 04: CONTROL DE VERSIONS: GIT (PART III): rebase

- **En aquest exemple**, **estant a la branca que volem fusionar**, l'acció consisteix a:
 - 1)Anar a l'ancestre comú de les dues branques:
 - ➔La que estem
 - ➔Aquella a la que volem fer el **rebase**.
 - 2)Obtenir la diferència introduïda per cada **commit** de la branca en la que estem.
 - 3)Guardar aquestes diferències en arxius temporals.
 - 4)Reiniciar la branca actual al **commit** de la branca a la qual es vol fer el **rebase**.
 - 5)Crear un nou i únic **commit** (perquè en la branca només n'hem fet un) en el que s'apliquen de manera incremental les diferències trobades anteriorment.

DESPLEGAMENT D'APLICACIONS WEB

TEMA 04: CONTROL DE VERSIONS: GIT (PART III): rebase

- En aquest punt, es pot tornar a la branca principal (**master**) i fer un **merge**:

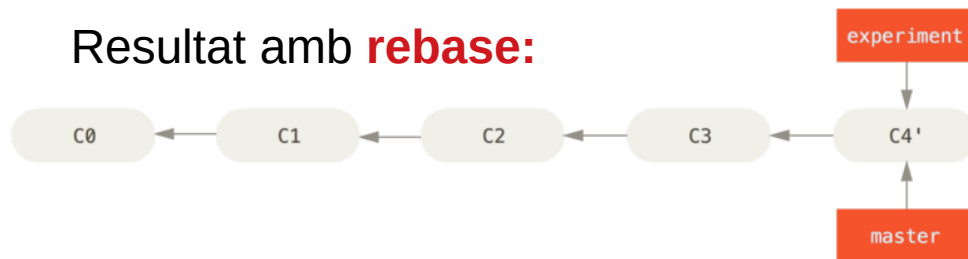


DESPLEGAMENT D'APLICACIONS WEB

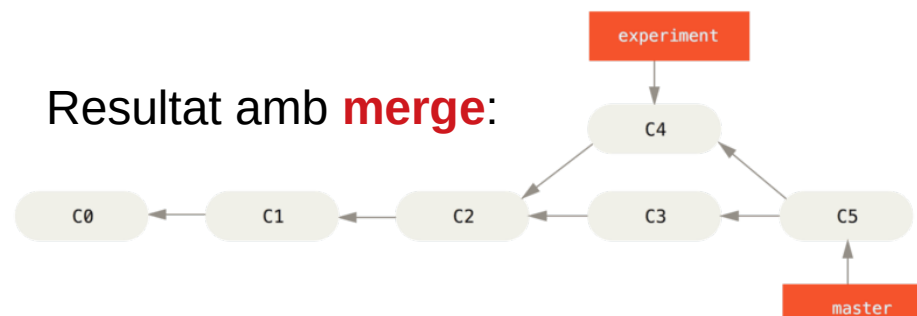
TEMA 04: CONTROL DE VERSIONS: GIT (PART III): rebase

- El resultat en **C4'** és exactament el mateix que l'obtingut en **C5** (exemple del **merge**). No hi ha diferència en el producte final de la integració,
- rebase** deixa un historial més net:
 - log d'una branca on s'ha utilitzat **rebase**: es veu com una història lineal
 - Sembla que tota la feina ha ocorregut en sèrie.
 - Encara que va succeir originalment en paral·lel.

Resultat amb **rebase**:



Resultat amb **merge**:



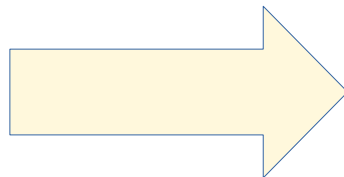
DESPLEGAMENT D'APLICACIONS WEB

TEMA 04: CONTROL DE VERSIONS: GIT (PART III): rebase

Altre exemple

- **No fa falta estar necessàriament en la branca a la que volem fer el rebase:**
 - En la mateixa instrucció podem indicar-ho amb un altre paràmetre.
 - Quan acabe el **rebase**, **experiment seguirà sent la branca activa**.

A---B---C *experiment
/
D---E---F---G master



A'--B'--C' *experiment
/
D---E---F---G master

experiment*: **git rebase master**

||

master*: **git rebase master experiment**

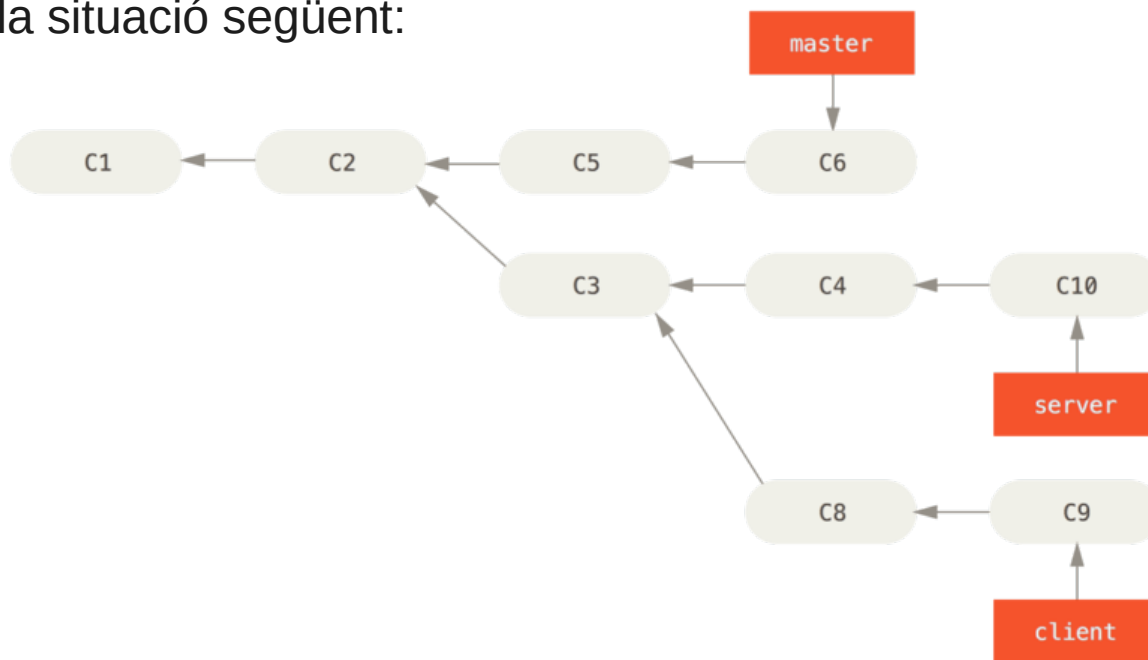
{ **git checkout experiment**
git rebase master

1) **rebase**

2) **Altres usos de rebase**

1) Rebase de branques no directament connectades

- **Exemple** a partir de la situació següent:

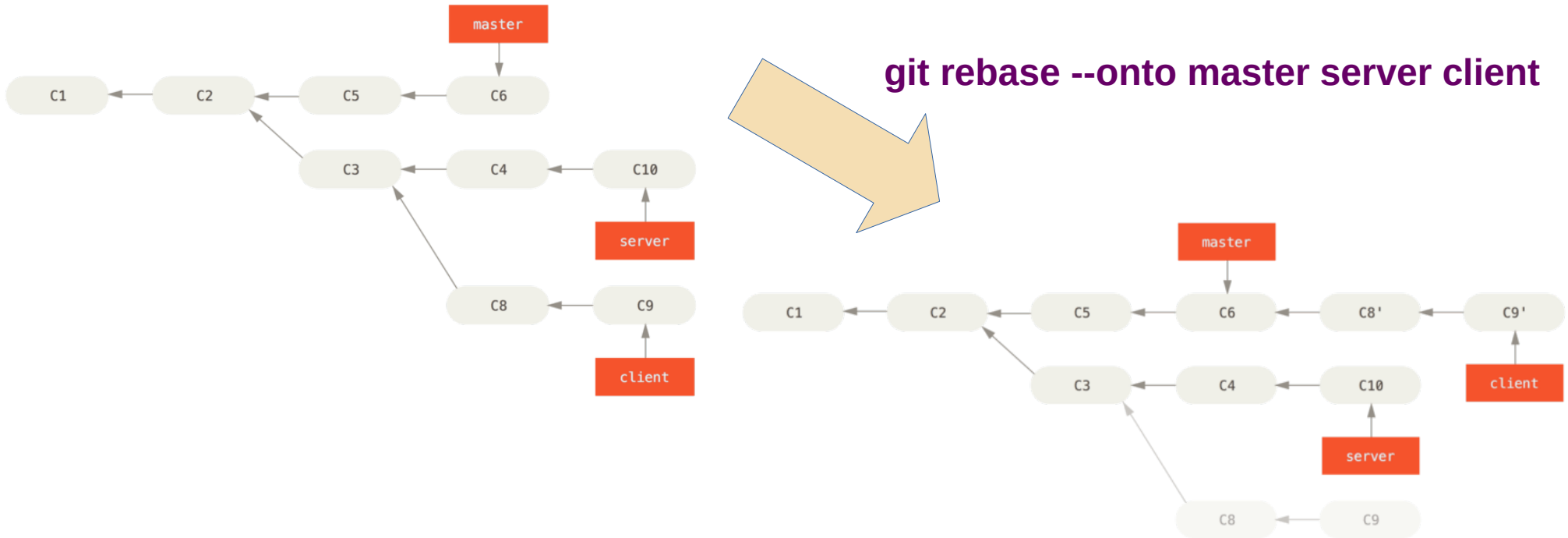


- Volem fusionar canvis de branca **client** en la línia **master** per a un llançament de codi.
- Volem mantindre sense aplicar els canvis del **server** fins que es facen més proves.
- Es pot prendre els canvis en el client que no estan al servidor (C8 i C9)
 - Reproduir-los en la branca principal mitjançant l'opció **--onto**:

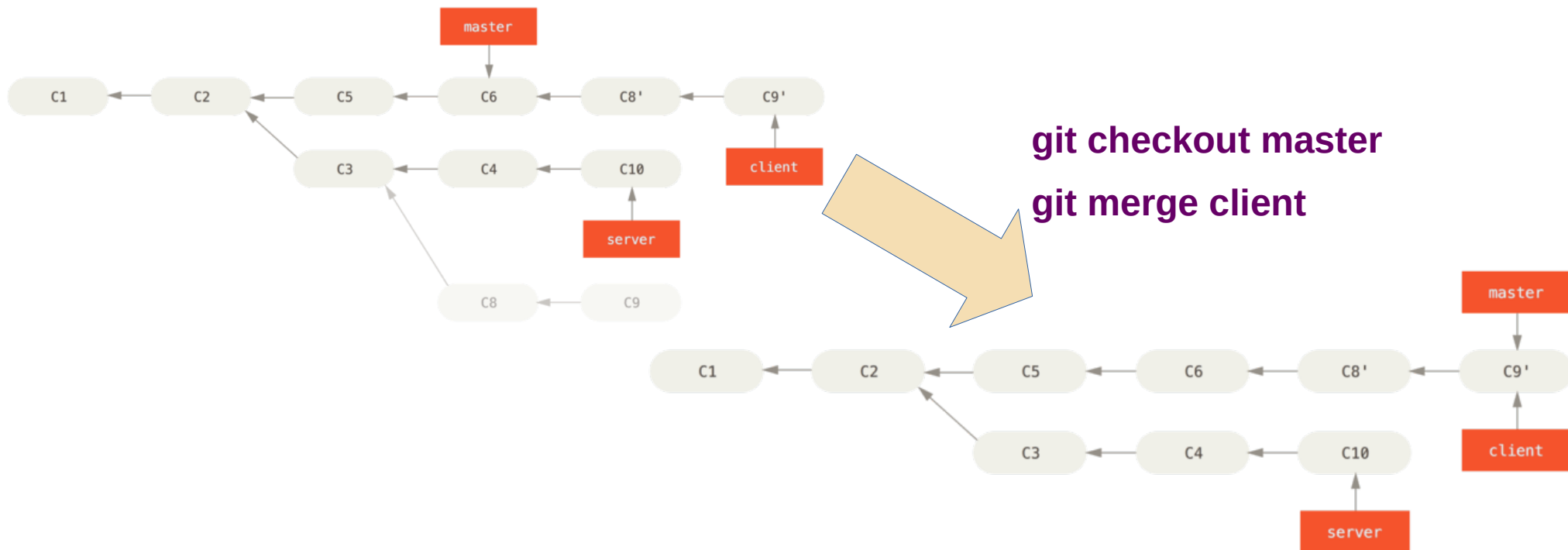
git rebase --onto master server client

- Aquesta instrucció vol dir:
 - 1) Agafa la branca client
 - 2) Esbrina els canvis des que es va separar de la branca del servidor
 - 3) Reprodueix aquests canvis en la branca master.

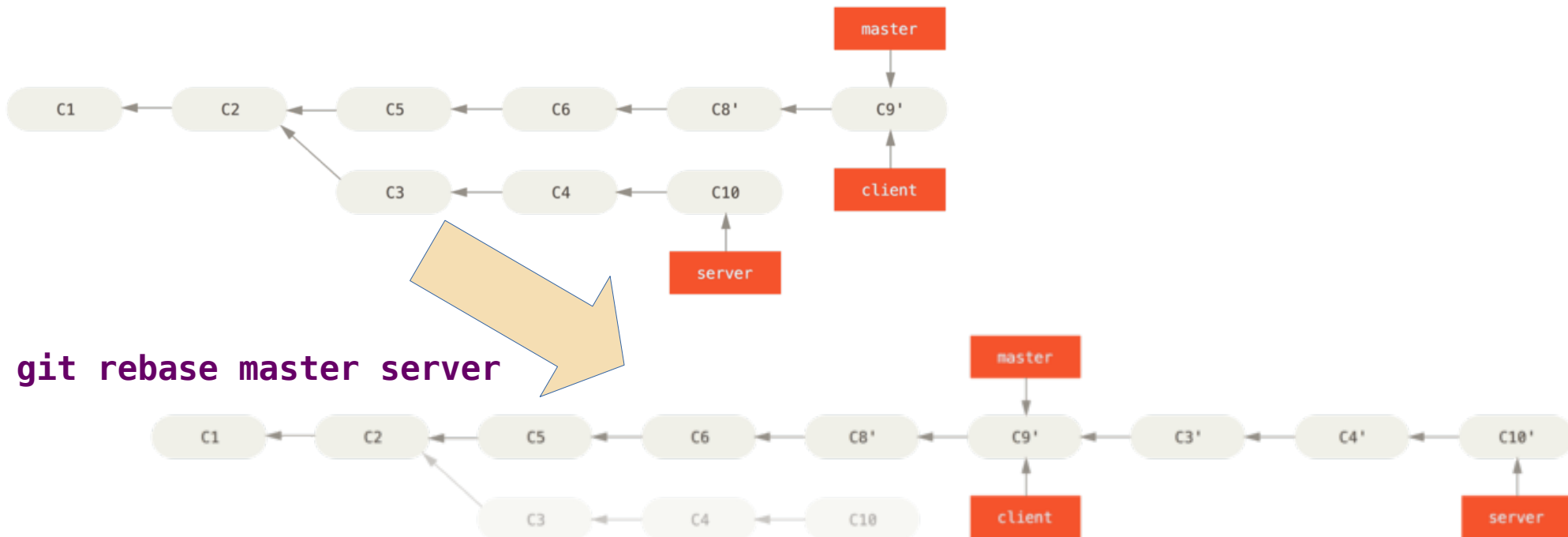
Rebase de branques no directament connectades



merge de la branca master i la client:

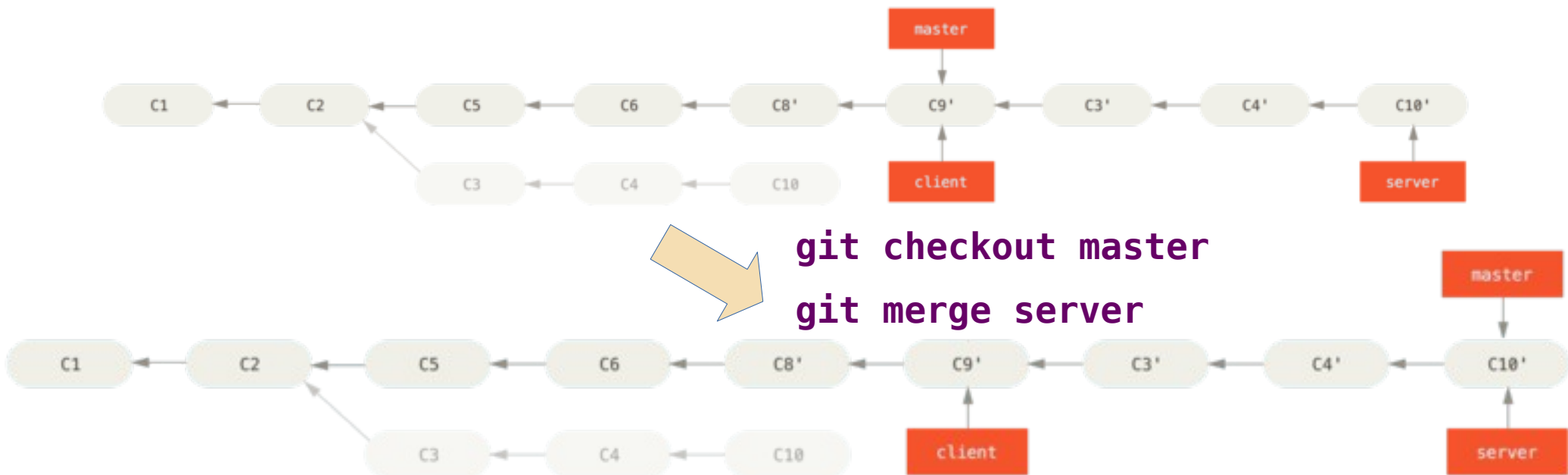


Integrar en la branca master els canvis fets en server



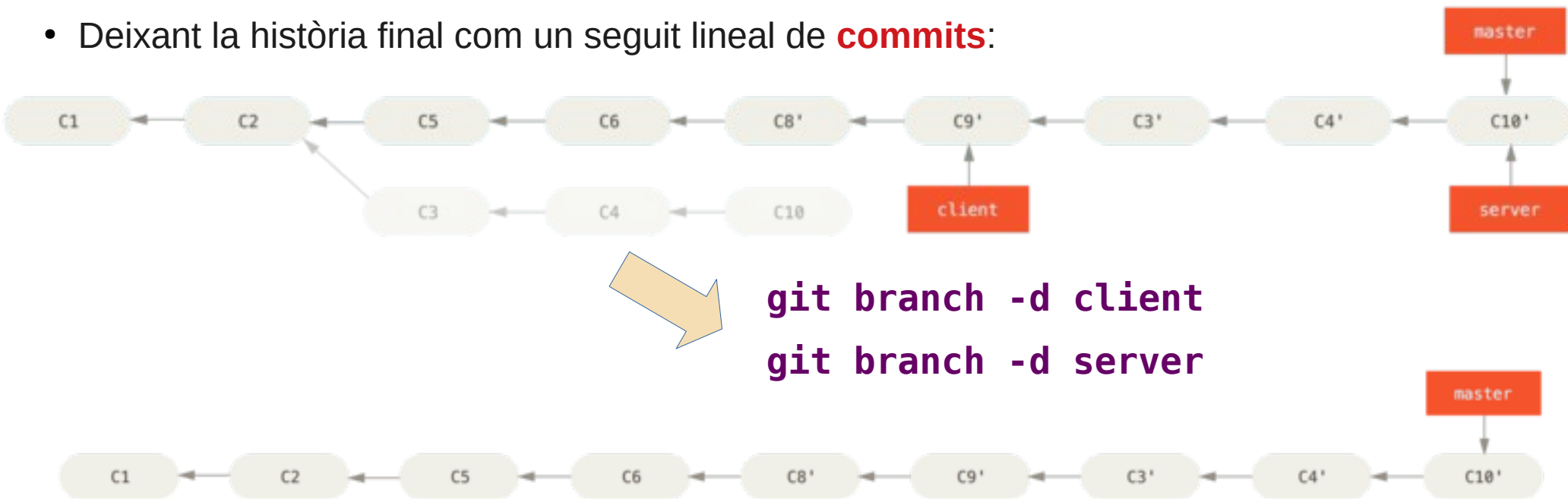
Unir la branca master amb la server

- HEAD del master estiga en el **commit** més avançat (fem un **fast forward**):



Eliminar les branques client i server

- Deixant la història final com un seguit lineal de **commits**:



Quina tècnica és millor utilitzar, rebase o merge?

- Si volem que:
 - La història reflectisca **tot el procés realitzat per arribar al resultat final**
 - Encara que s'hagen realitzat moltes proves, canvis i errades
 - S'ha d'utilitzar el **merge**.
- Si el que volem és:
 - Deixar una **història neta i fàcil de seguir per comprendre tots els canvis que s'han fet**,
 - És millor utilitzar **rebase**.

- Ús comú :
 - Utilitzar **rebase** en canvis locals que encara no han sigut compartits amb un **push** per deixar més neta la història.
 - No fer el **rebase** d'alguna cosa que ja s'ha pujat.

2) Utilitzant rebase per a diversos commits en un mateix branch

- **rebase** permet combinar diversos **commits** en un d'únic.
- Permet a l'usuari **reescriure una mica la història dels commits** (fent una mica de neteja) **abans de fer un push** dels canvis a un repositori remot.
- **Exemple** : Crearem diversos **commits** que seran combinats més endavant.

- Creem un nou arxiu

touch rebase.txt

- Afegim a l'índex i fem el **commit**

git add . && git commit -m "rebase.txt afegit a l'índex"

```
professor@pcprofessor:~/repo01$ touch rebase.txt
professor@pcprofessor:~/repo01$ git add . && git commit -m "rebase.txt afegit a
l'índex"
[master cdf42c9] rebase.txt afegit a l'índex
 2 files changed, 5 insertions(+)
 create mode 100644 conflictiu.txt.orig
 create mode 100644 rebase.txt
professor@pcprofessor:~/repo01$ git log
commit cdf42c9995998080e1c47ffdfb141e4d75214c66 (HEAD -> master)
Author: Professor Entorns <professor@entorns.com>
Date:   Tue Apr 9 07:30:55 2019 +0200

    rebase.txt afegit a l'índex
```

DESPLEGAMENT D'APLICACIONS WEB

TEMA 04: CONTROL DE VERSIONS: GIT (PART III): Altres usos de rebase

- Realitzem alguns xicotets canvis al fitxer **rebase.txt**.
- Fem un **commit** en cada cas.

```
professor@pcprofessor:~/repo01$ echo "més contingut" >> rebase.txt
professor@pcprofessor:~/repo01$ git add . && git commit -m "agregue més contingut"
[master 8168dcf] agregue més contingut
1 file changed, 1 insertion(+)
professor@pcprofessor:~/repo01$ echo "més contingut" >> rebase.txt
professor@pcprofessor:~/repo01$ git add . && git commit -m "agregue més contingut"
[master 300f840] agregue més contingut
1 file changed, 1 insertion(+)

professor@pcprofessor:~/repo01$ git log
commit 300f84071ea55a733aaa9b14b25f154c50169082 (HEAD -> master)
Author: Professor Entorns <professor@entorns.com>
Date: Tue Apr 9 07:33:52 2019 +0200

    agregue més contingut

commit 8168dcfe587120fd4fb55791b9476ccbb3bc987e
Author: Professor Entorns <professor@entorns.com>
Date: Tue Apr 9 07:33:51 2019 +0200

    agregue més contingut

commit 1c0c3b141ab9d3aa69e1d6593699fc264e22e52e
Author: Professor Entorns <professor@entorns.com>
Date: Tue Apr 9 07:33:51 2019 +0200

    agregue més contingut
```


- **Combinarem els últims set commits** mitjançant la següent ordre:

git rebase -i HEAD~7

- Paràmetre -i (interactive) obre editor de text
- Opcions per editar cada missatge de **commit**
 - Mantenir (**pick**),
 - Mantenir i canviar el missatge (**reword**)
 - Compactar amb el **commit** anterior (**squash**)
 - Arreglar (**fixup**)
 - Eliminar (**drop**).

```
professor@pcprofessor: ~/repo01
Fitxer  Edita  Visualitza  Cerca  Terminal  Ajuda
/home/professor/repo01/.git/rebase-merge/git-rebase-todo

pick cdf42c9 rebase.txt afegit a l'index
pick b9c128c agregue contingut
pick 635e517 agregue més contingut
pick e8119f3 agregue més contingut
pick 1c0c3b1 agregue més contingut
pick 8168dcf agregue més contingut
pick 300f840 agregue més contingut

# Rebase 1ed017f..300f840 sobre 1ed017f (7 ordres)
#
# Ordres:
# p, pick = usa la comissió
# r, reword = usa la comissió, però edita el missatge de comissió
# e, edit = usa la comissió, però atura't per a esmenar
# s, squash = usa la comissió, però fusiona'l a la comissió prèvia
# f, fixup = com "squash", però descarta el missatge de registre d'aquesta comissió
# x, exec = executa l'ordre (la resta de la línia) usant l'interpret d'ordres
# d, drop = elimina la comissió
#

[ 26 línies llegides ]
^G Ajuda    ^O Desa    ^W On és    ^K Retalla  ^J Justifica ^C Pos Act
^X Surt     ^R Llegeix  ^E Reemplaça ^U Enganxa  ^T Ortografia ^_ Vés a línia
```

- Si volem fusionar tots els **commits**:
 - Utilitzem en totes les branques l'opció **squash**:
 - Excepte en la primera, que utilitzarem com a **commit** per a reunir-les.

```
/home/professor/repo01/.git/rebase-merge/git-rebase-todo  Modificat  
pick cdf42c9 rebase.txt afegit a l'index  
squash b9c128c agregue contingut  
squash 635e517 agregue més contingut  
squash e8119f3 agregue més contingut  
squash 1c0c3b1 agregue més contingut  
squash 8168dcf agregue més contingut  
squash 300f840 agregue més contingut
```

- Després de guardar i tancar aquesta part,
- S'obre de nou un editor de text per modificar el missatge corresponent al **commit** resultant:

```
GNU nano 2.9.3 /home/professor/repo01/.git/COMMIT_EDITMSG
# Això és una combinació de 7 comissions.
# Aquest és el 1r missatge de comissió:

rebase.txt afegit a l'index

# Aquest és el missatge de comissió #2:

agregue contingut

# Aquest és el missatge de comissió #3:

agregue més contingut

# Aquest és el missatge de comissió #4:

agregue més contingut

# Aquest és el missatge de comissió #5:

agregue més contingut
```

- Deixem només un missatge indicant que s'ha fet un **rebase**:

```
GNU nano 2.9.3      /home/professor/repo01/.git/COMMIT_EDITMSG      Modificat

# Això és una combinació de 7 comissions.
# Aquest és el 1r missatge de comissió:

rebase.txt afegit a l'índex, CANVIS UNITS AMB rebase

# Introduïu el missatge de comissió dels vostres canvis.
# S'ignoraran les línies començant amb '#', i un missatge de
# comissió buit avorta la comissió.
```

DESPLEGAMENT D'APLICACIONS WEB

TEMA 04: CONTROL DE VERSIONS: GIT (PART III): Altres usos de rebase

- Té lloc la unió dels diferents **commits** en un de sol, canviant el missatge:

```
professor@pcprofessor:~/repo01$ git rebase -i HEAD~7
[HEAD separat f52ef20] rebase.txt afegit a l'índex, CANVIS UNITS AMB rebase
Date: Tue Apr 9 07:30:55 2019 +0200
2 files changed, 11 insertions(+)
create mode 100644 conflictiu.txt.orig
create mode 100644 rebase.txt
Successfully rebased and updated refs/heads/master.
professor@pcprofessor:~/repo01$ git log
commit f52ef203cc7212e549d2fb72bf9fe942aa9e1e31 (HEAD -> master)
Author: Professor Entorns <professor@entorns.com>
Date: Tue Apr 9 07:30:55 2019 +0200
```

```
rebase.txt afegit a l'índex, CANVIS UNITS AMB rebase
```

```
professor@pcprofessor:~/repo01$ cat rebase.txt
contingut
més contingut
més contingut
més contingut
més contingut
més contingut
```

- Podem veure com els continguts introduïts en **rebase.txt**, també es mantenen:

3) rebase entre branches

- **Com ja hem explicat més amunt:**
 - **merge** combina els canvis de dues branques.
 - **rebase** pren els canvis d'una branca, crea un patch i l'aplica a l'altra branca
 - El resultat final del codi font és el mateix en ambdós casos.
 - Amb **merge** la història de **commit** és més neta: sembla ser lineal.
- Veiem ara les instruccions del que hem vist prèviament de manera gràfica, en la consola.

- Creem una **nova branca** i ens hi canviem:

git branch provarebase

git checkout provarebase

git branch

```
professor@pcprofessor:~/repo01$ git branch
* master
professor@pcprofessor:~/repo01$ git branch provarebase
professor@pcprofessor:~/repo01$ git checkout provarebase
S'ha canviat a la branca «provarbase»
professor@pcprofessor:~/repo01$ git branch
  master
* provarebase
```

- Fem alguns **canvis** i el **commit** corresponent, comprovant la història després:

`echo "Farem un rebase amb el màster"> test01`

`git commit -a -m "Alguna cosa nova en el branch"`

`git log`

```
professor@pcprofessor:~/repo01$ echo "Farem un rebase amb el màster"> test01
professor@pcprofessor:~/repo01$ git commit -a -m "Alguna cosa nova en el branch"
[provarebase 4b3a36f] Alguna cosa nova en el branch
1 file changed, 1 insertion(+), 1 deletion(-)
professor@pcprofessor:~/repo01$ git log
commit 4b3a36ff1ca0234c9b4230bfe697f9ab403d702c (HEAD -> provarebase)
Author: Professor Entorns <professor@entorns.com>
Date: Tue Apr 9 08:10:29 2019 +0200

    Alguna cosa nova en el branch
```


- Ara tornem a la branca master i realitzem algun **commit**:

git checkout master

echo "Farem un canvi amb el màster"> test02

git commit -a -m "Alguna cosa nova en el master"

```
professor@pcprofessor:~/repo01$ git checkout master
S'ha canviat a la branca «master»
professor@pcprofessor:~/repo01$ echo "Farem un canvi amb el màster"> test02
professor@pcprofessor:~/repo01$ git commit -a -m "Alguna cosa nova en el master"
[master 408520d] Alguna cosa nova en el master
 1 file changed, 1 insertion(+), 1 deletion(-)
professor@pcprofessor:~/repo01$ git log
commit 408520d29c9302c7a33f3b6cae9776c68a3b5a39 (HEAD -> master)
Author: Professor Entorns <professor@entorns.co
Date: Tue Apr 9 08:12:45 2019 +0200

    Alguna cosa nova en el master
```

- Tornem a la branca provarebase, fem un **rebase** amb el màster i comprovem el resultat amb el log:

git checkout provarebase

git rebase master

git log

```
professor@pcprofessor:~/repo01$ git checkout provarebase
S'ha canviat a la branca «provarabase»
professor@pcprofessor:~/repo01$ git rebase master
Primer, s'està rebobinant HEAD per a reproduir el vostre treball al damunt...
S'està aplicant: Alguna cosa nova en el branch
professor@pcprofessor:~/repo01$ git log
commit df2fd46ea0dd78aac013ff083271e861145038ef (HEAD -> provarebase)
Author: Professor Entorns <professor@entorns.co>
Date: Tue Apr 9 08:10:29 2019 +0200

    Alguna cosa nova en el branch

commit 408520d29c9302c7a33f3b6cae9776c68a3b5a39 (master)
Author: Professor Entorns <professor@entorns.co>
Date: Tue Apr 9 08:12:45 2019 +0200

    Alguna cosa nova en el master
```