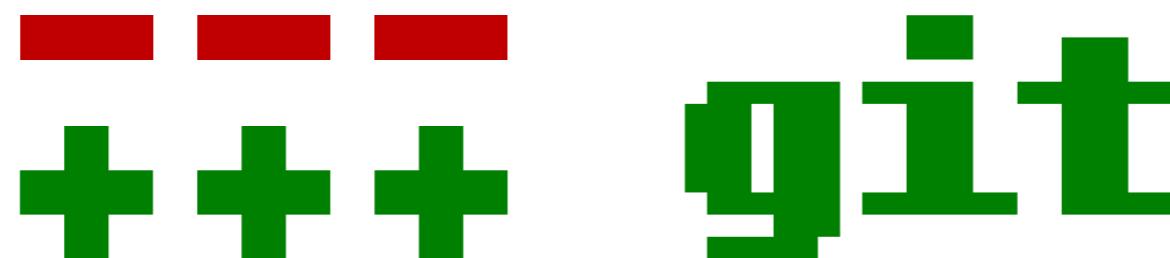


# Taller de



Autor original de las transparencias:  
Bart Trojanowski



<http://www.jukie.net/~bart/>  
[bart@jukie.net](mailto:bart@jukie.net)

Traducción y modificaciones:  
Antonio García Domínguez

aquí estoy :-)

<http://xmleye.wordpress.com>  
[antonio.garciadominguez@uca.es](mailto:antonio.garciadominguez@uca.es)

# Gestión del Código Fuente

# Gestión del Código Fuente

## - Sistema de Control de Revisiones

# Gestión del Código Fuente

- Sistema de Control de Revisiones
- Sistema de Control de Versiones

# ¿Por qué usar uno?

# ¿Por qué usar uno?

- historial de cambios

# ¿Por qué usar uno?

- historial de cambios
- desarrollo paralelo

# ¿Por qué usar uno?

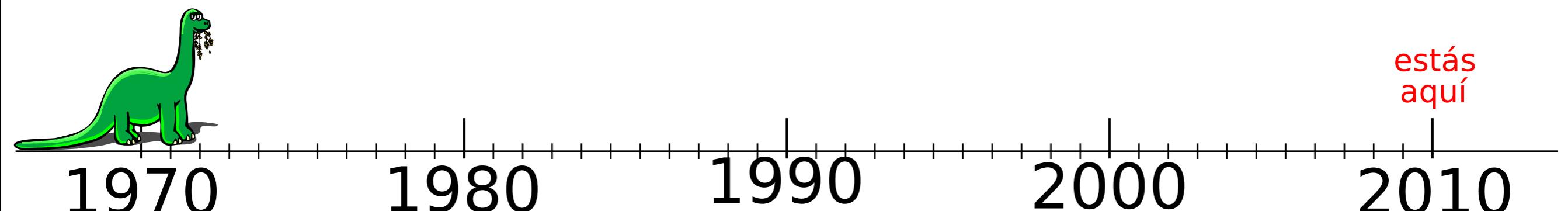
- historial de cambios
- desarrollo paralelo
- colaboración

# ¿Por qué usar uno?

- historial de cambios
- desarrollo paralelo
- colaboración
- copias de seguridad

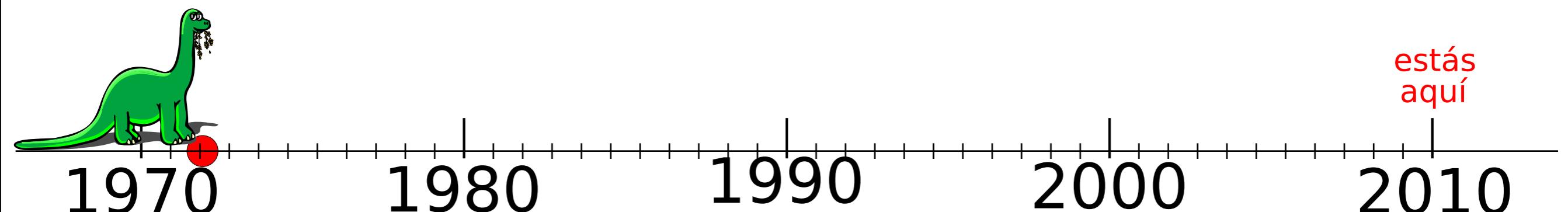
Conceptos

# Historia



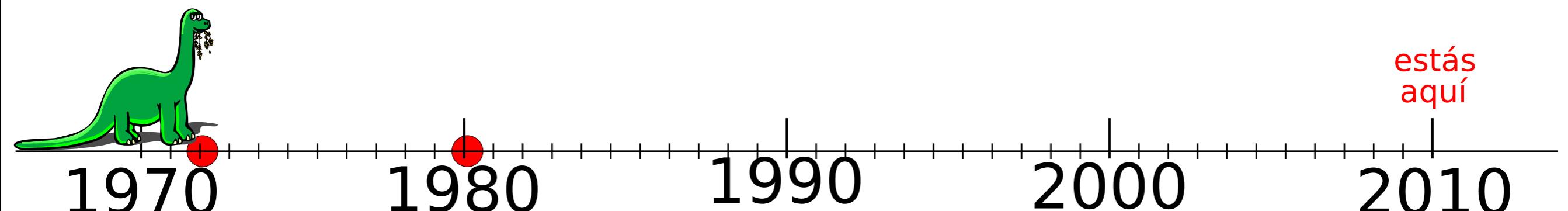
# Historia - SCCS

(1972)



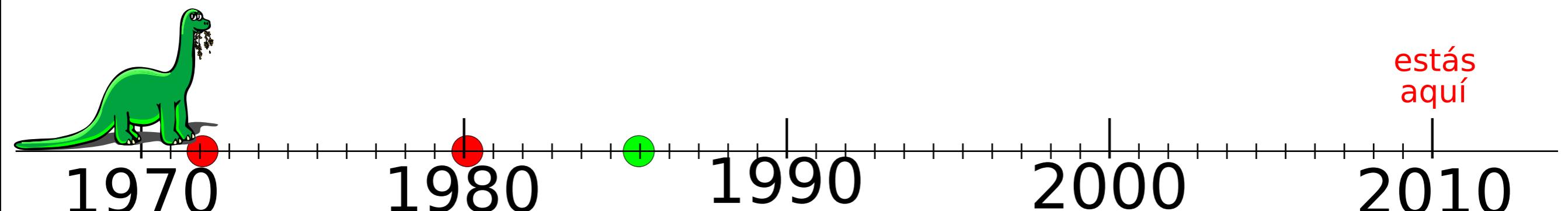
# Historia

- SCCS (1972)
- RCS (1980)



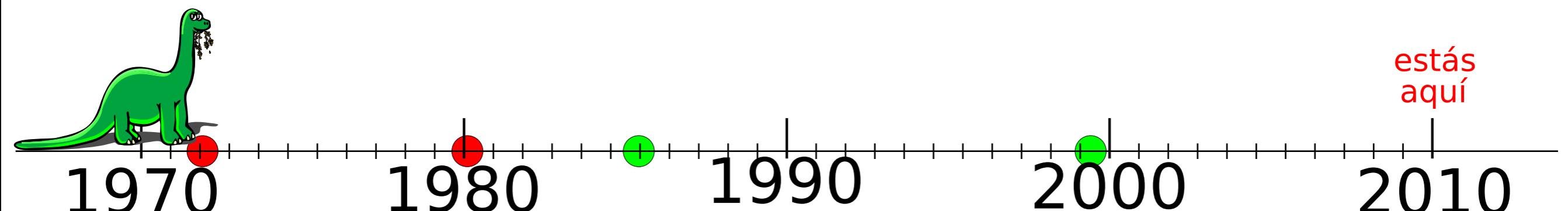
# Historia

- SCCS (1972)
- RCS (1980)
- CVS (1986)



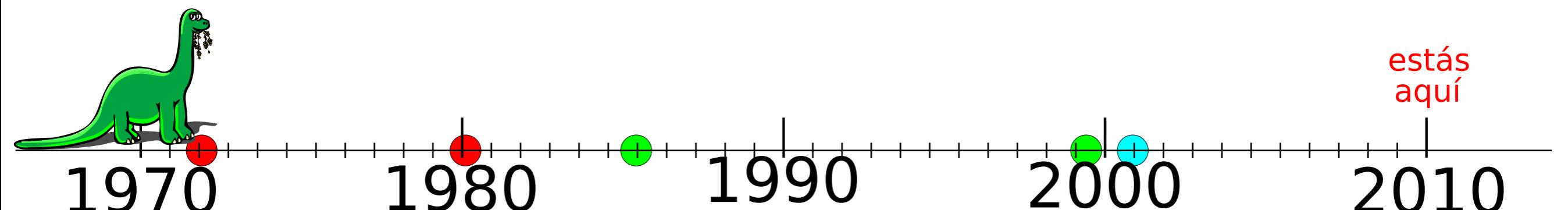
# Historia

- SCCS (1972)
- RCS (1980)
- CVS (1986)
- Subversion (SVN) (1999)



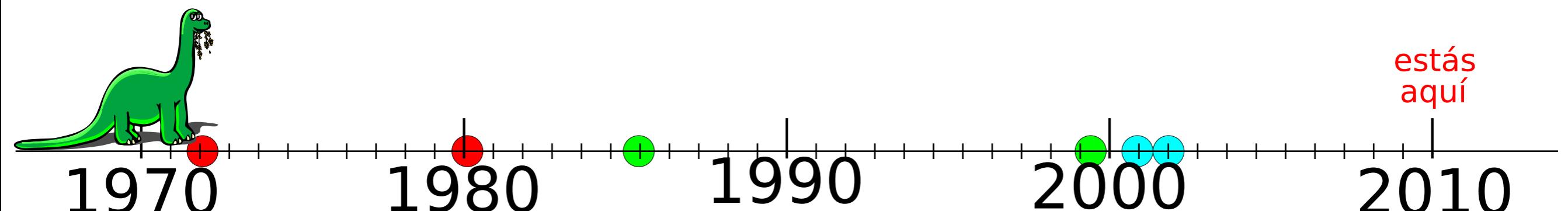
# Historia

- SCCS (1972)
- RCS (1980)
- CVS (1986)
- Subversion (SVN) (1999)
- Arch, monotone (2001)



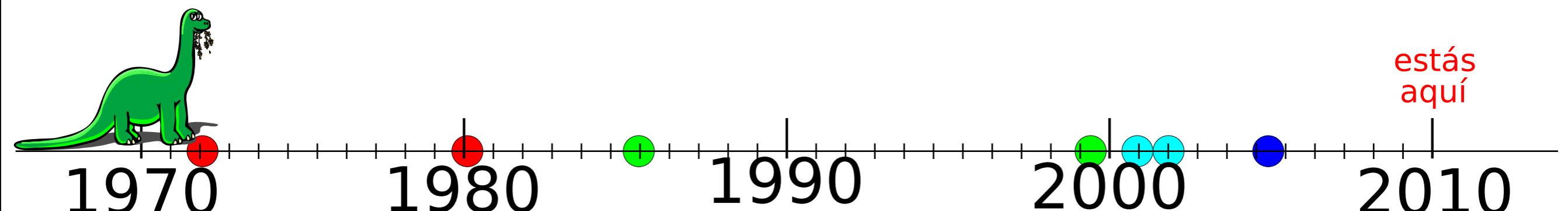
# Historia

- SCCS (1972)
- RCS (1980)
- CVS (1986)
- Subversion (SVN) (1999)
- Arch, monotone (2001)
- Darcs (2002)



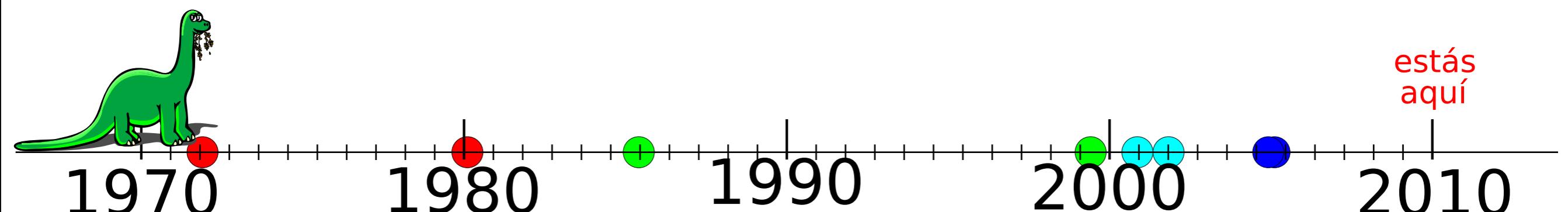
# Historia

- SCCS (1972)
- RCS (1980)
- CVS (1986)
- Subversion (SVN) (1999)
- Arch, monotone (2001)
- Darcs (2002)
- Git (2005)



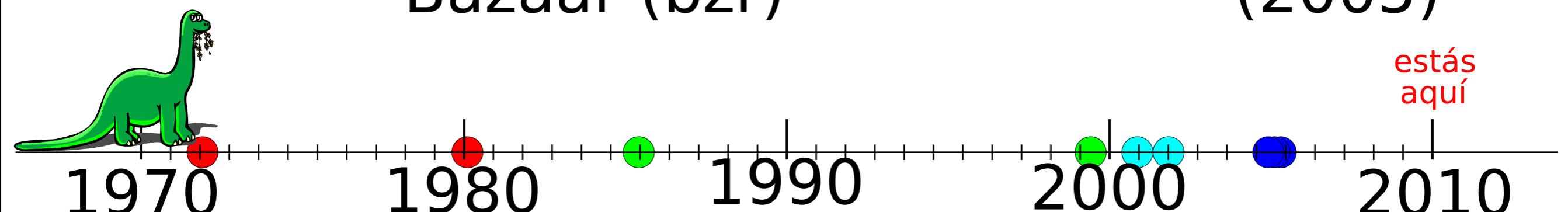
# Historia

- SCCS (1972)
- RCS (1980)
- CVS (1986)
- Subversion (SVN) (1999)
- Arch, monotone (2001)
- Darcs (2002)
- Git (2005)
- Mercurial (hg) (2005)



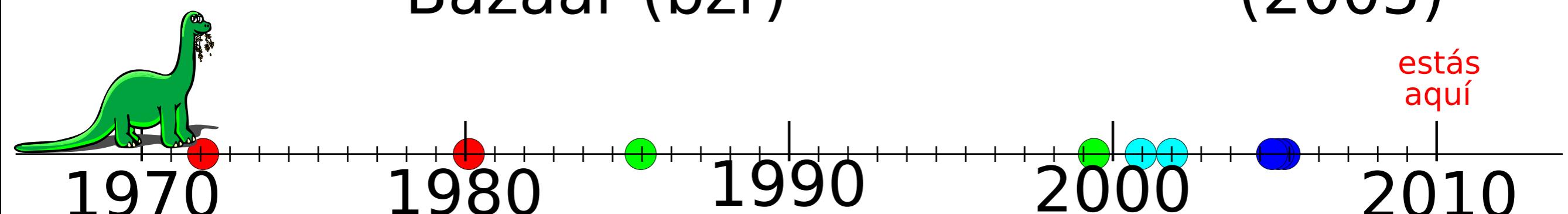
# Historia

- SCCS (1972)
- RCS (1980)
- CVS (1986)
- Subversion (SVN) (1999)
- Arch, monotone (2001)
- Darcs (2002)
- Git (2005)
- Mercurial (hg) (2005)
- Bazaar (bzr) (2005)



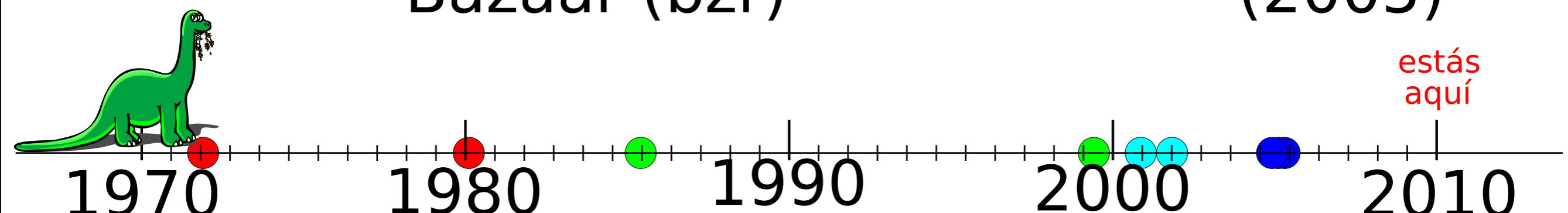
# Historia

- SCCS (1972)
- RCS (1980)
- CVS (1986)
- Subversion (SVN) (1999)
- Arch, monotone (2001)
- Darcs (2002)
- Git (2005)
- Mercurial (hg) (2005)
- Bazaar (bzr) (2005)



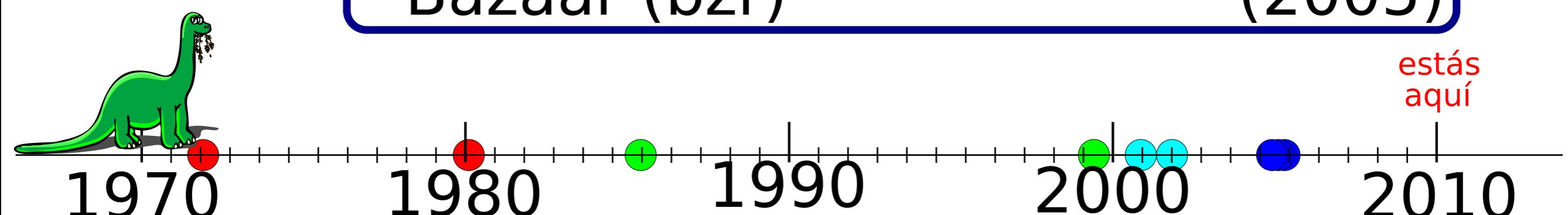
# Historia

- SCCS (1972)
- RCS (1980)
- CVS (1986)
- Subversion (SVN) (1999)
- Arch, monotone (2001)
- Darcs (2002)
- Git (2005)
- Mercurial (hg) (2005)
- Bazaar (bzr) (2005)



# Historia

- SCCS (1972)
- RCS (1980)
- CVS (1986)
- Subversion (SVN) (1999)
- Arch, monotone (2001)
- Darcs (2002)
- Git (2005)
- Mercurial (hg) (2005)
- Bazaar (bzr) (2005)

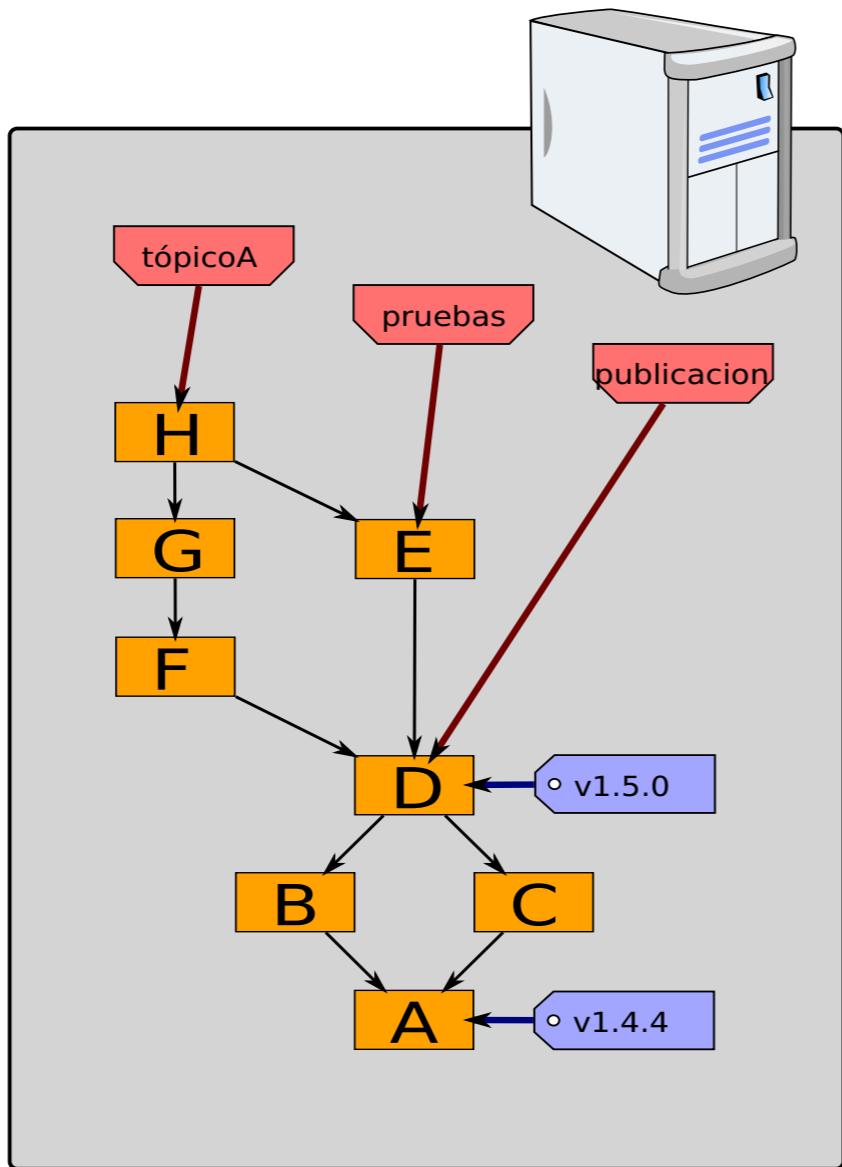


# SCV centralizado

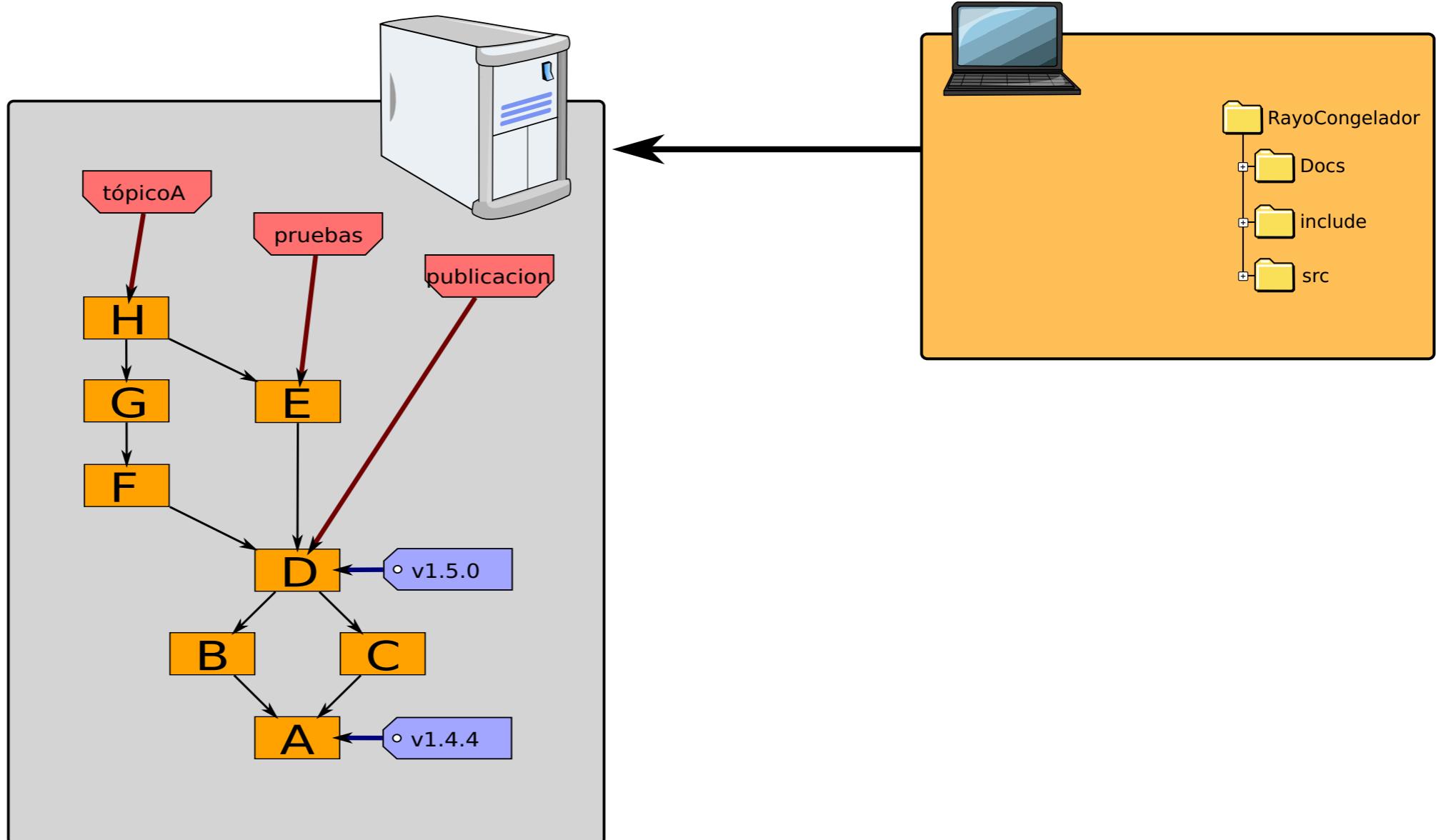
Centralizado



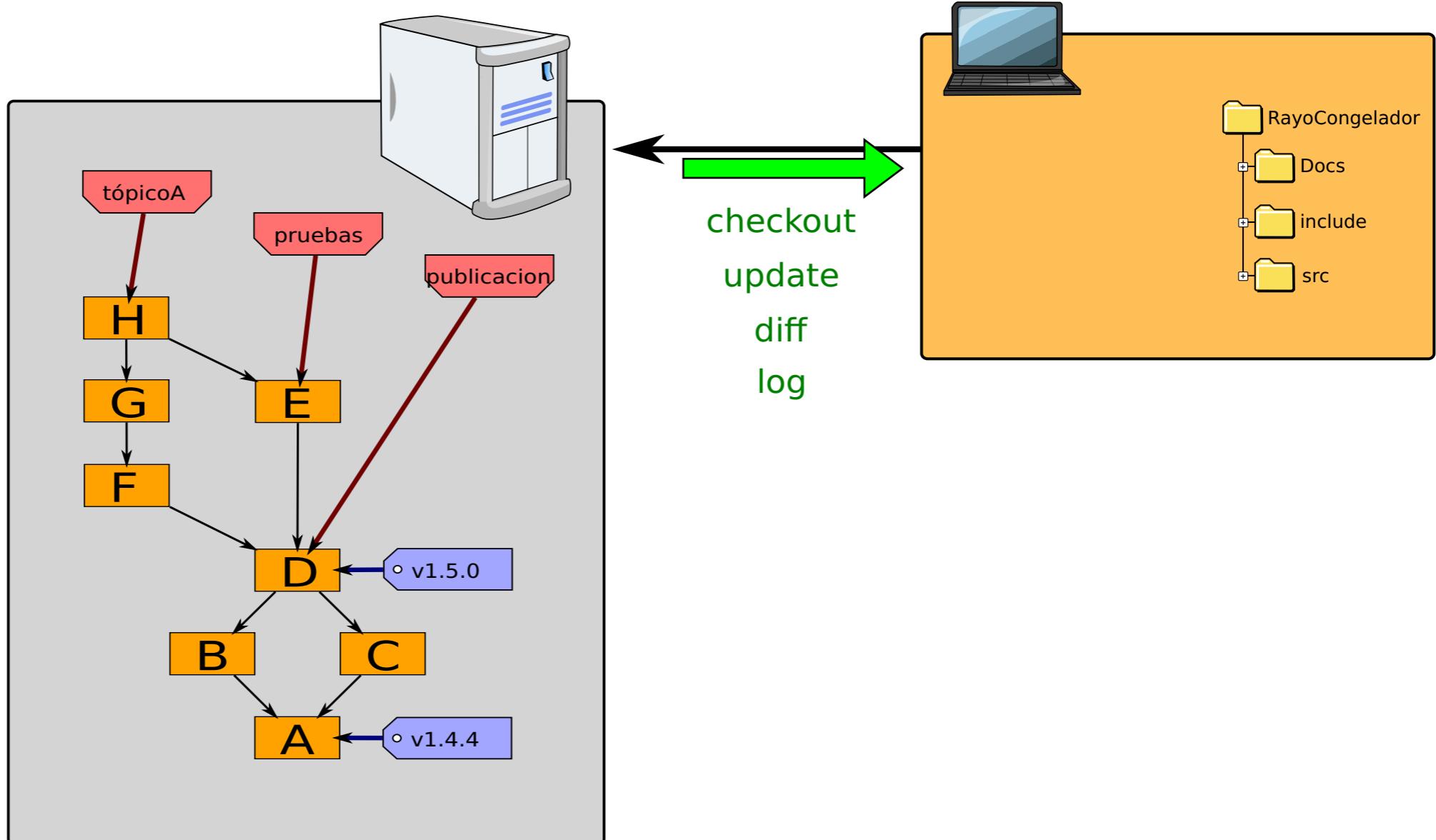
# Centralizado



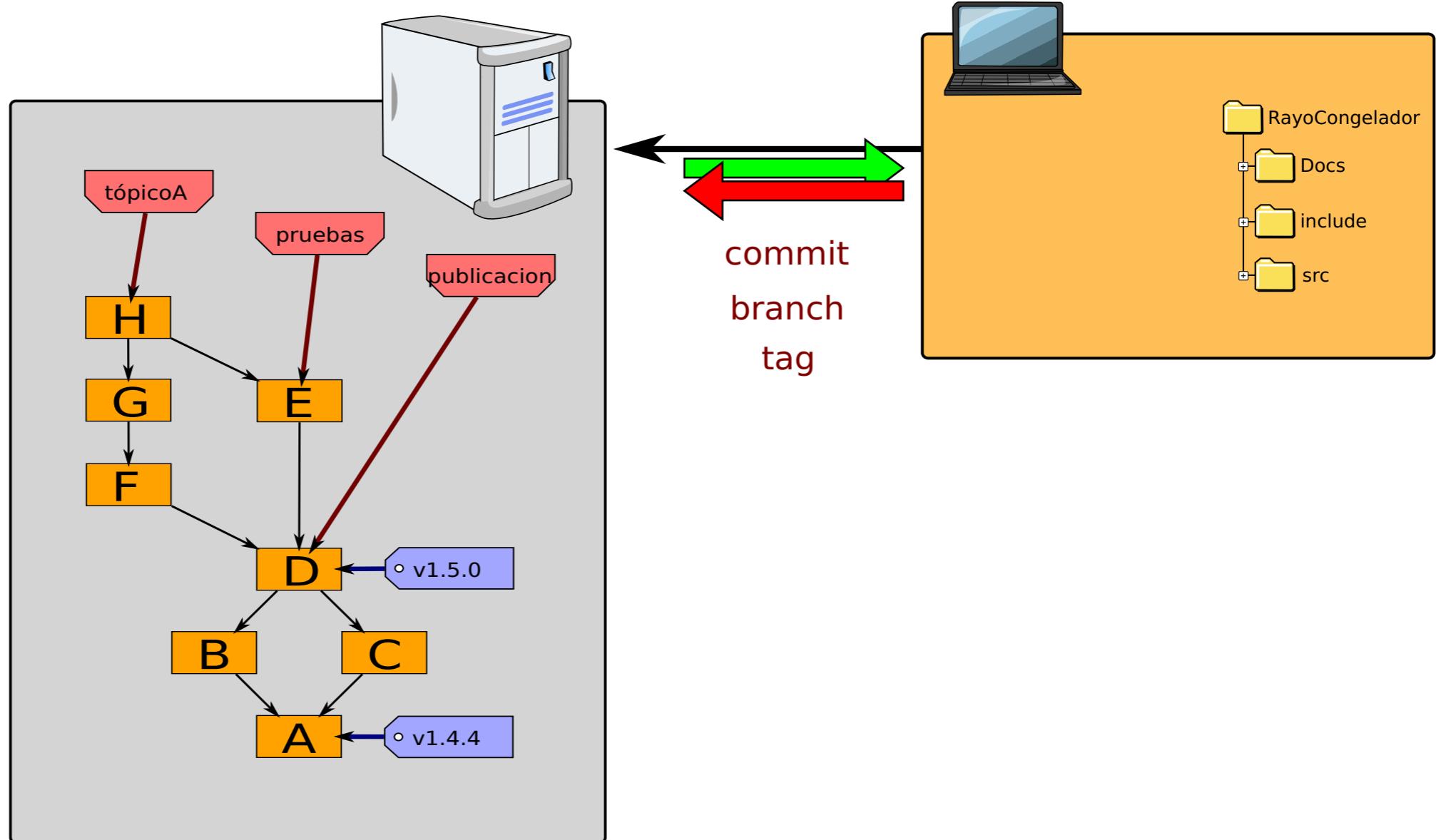
# Centralizado



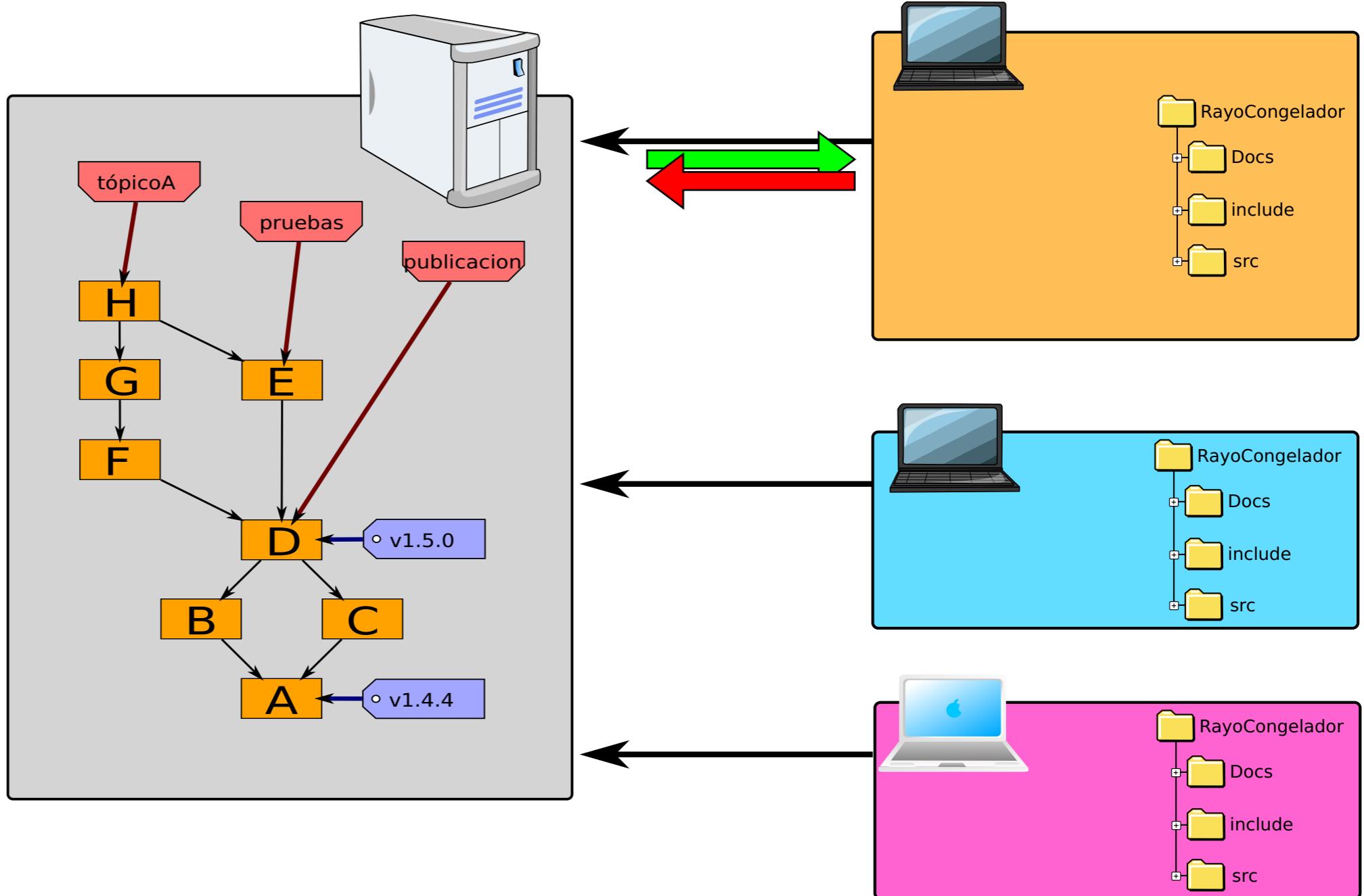
# Centralizado



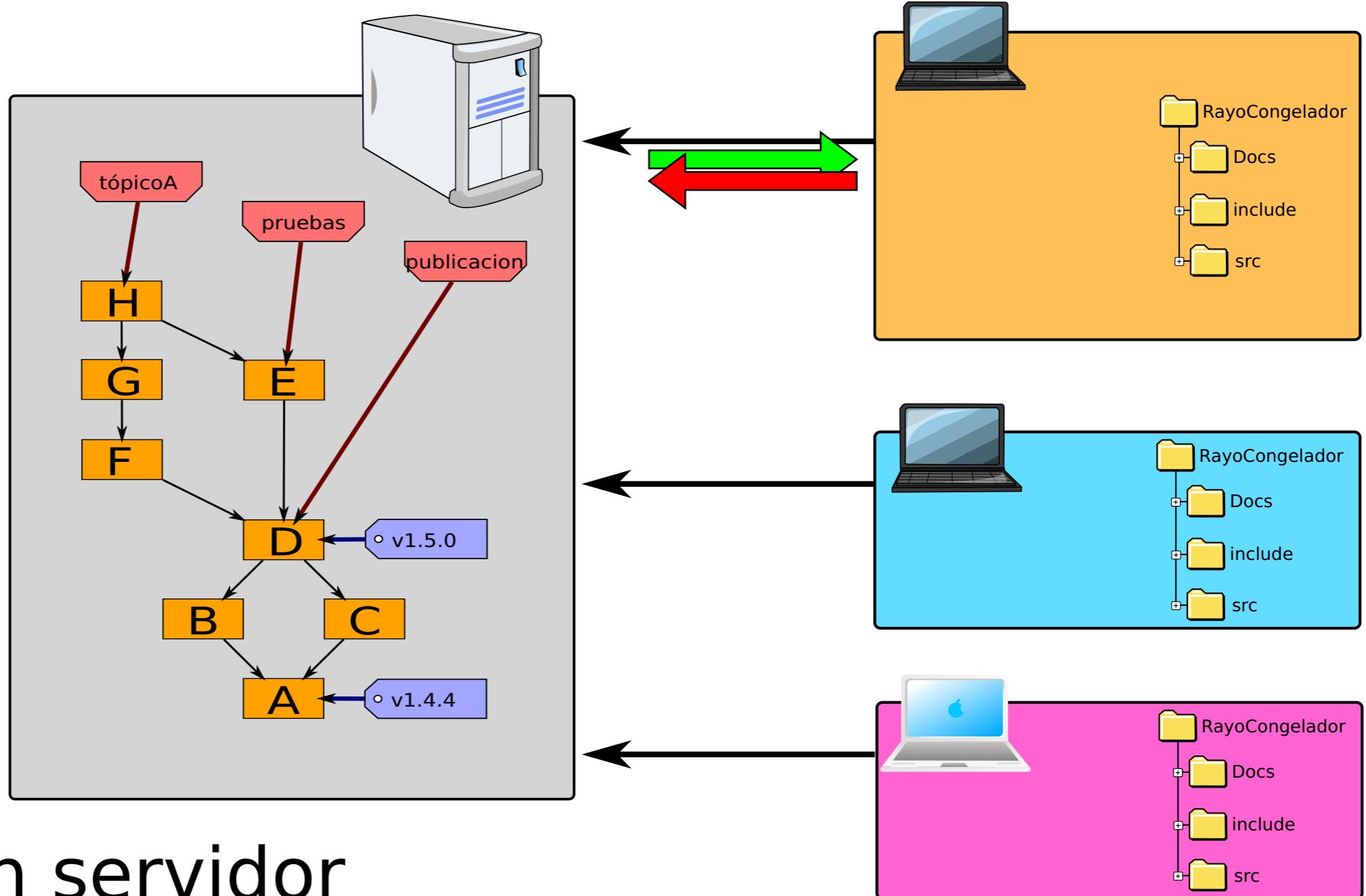
# Centralizado



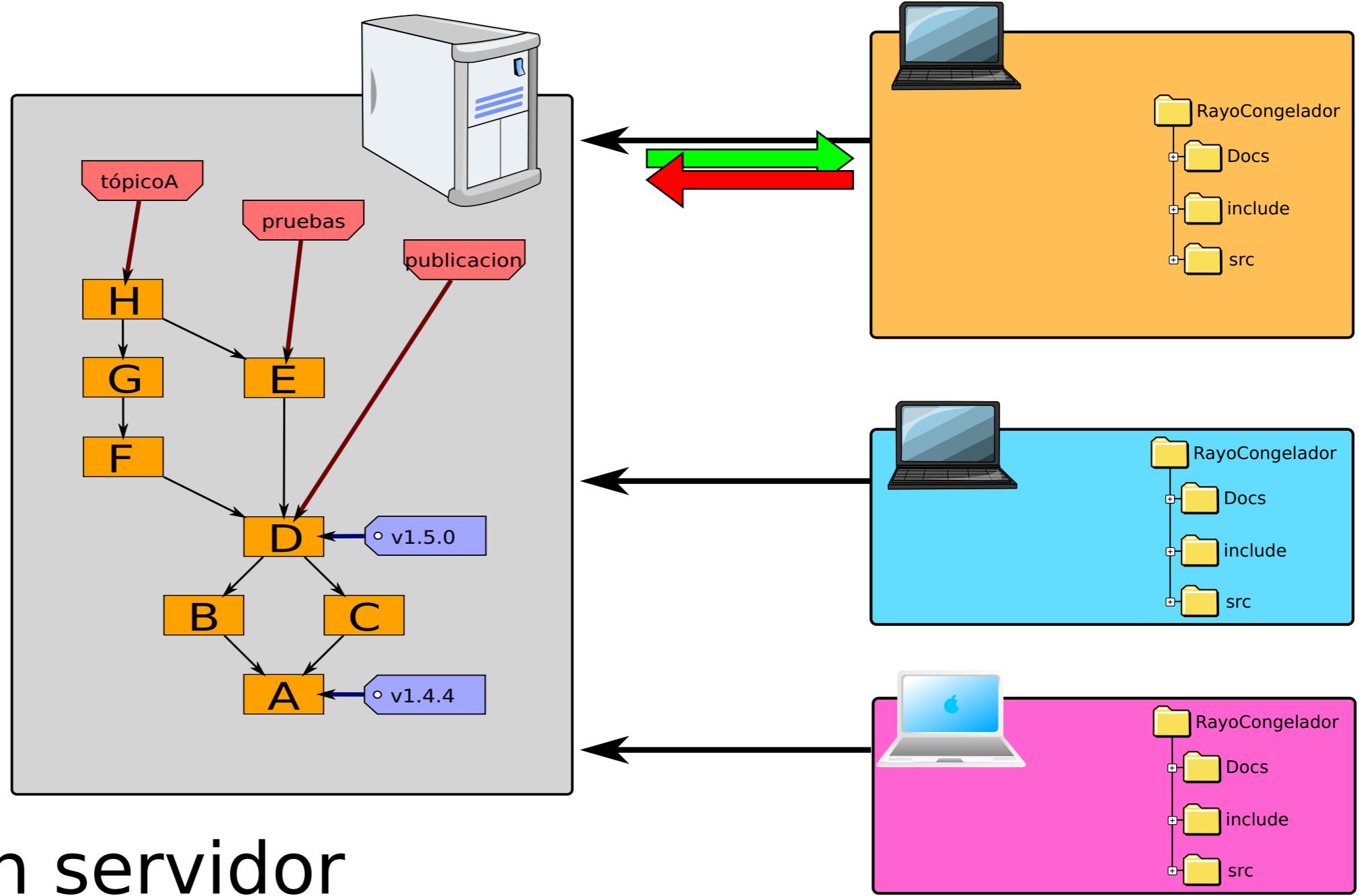
# Centralizado



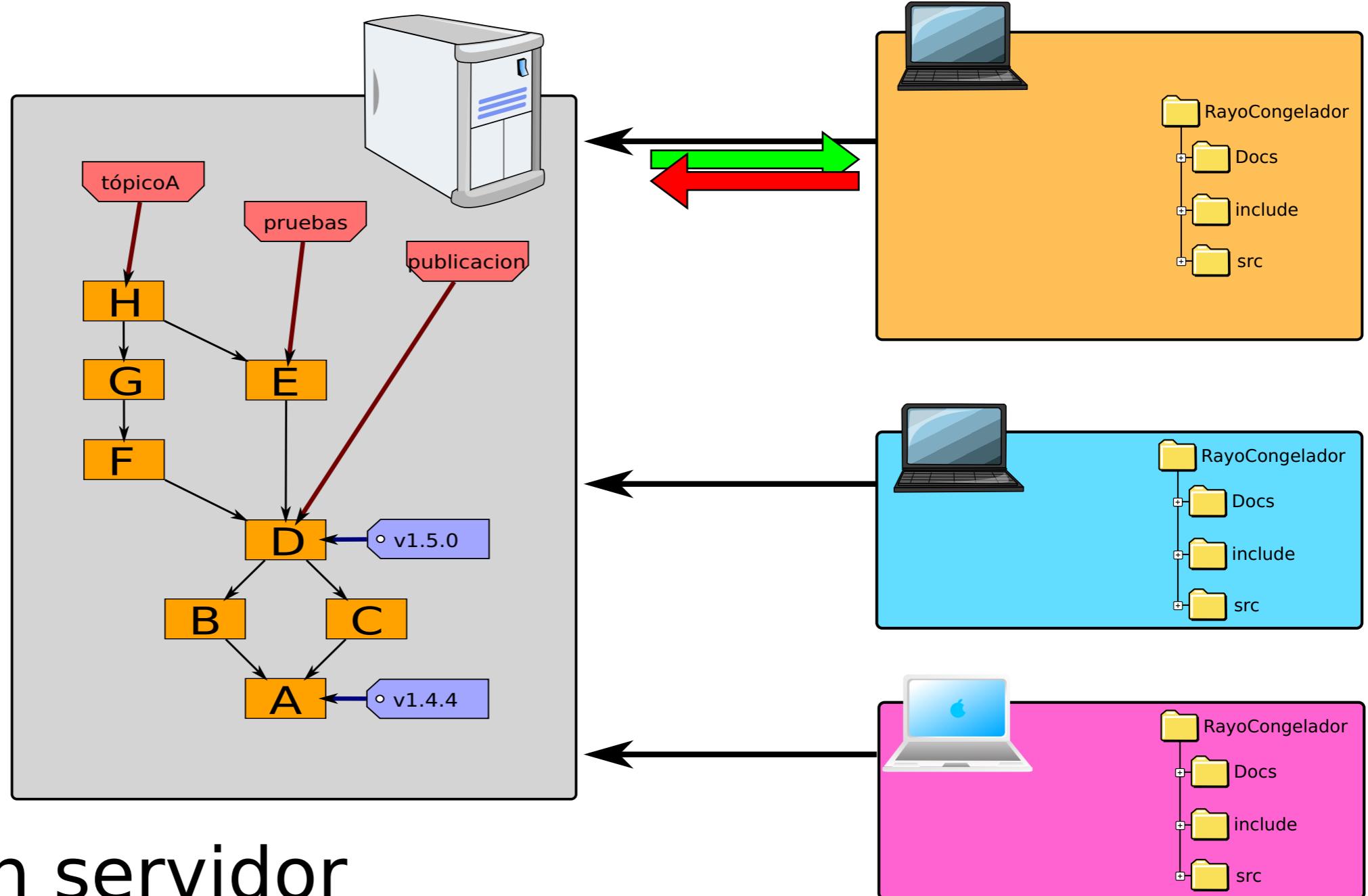
# Centralizado



- sólo un servidor

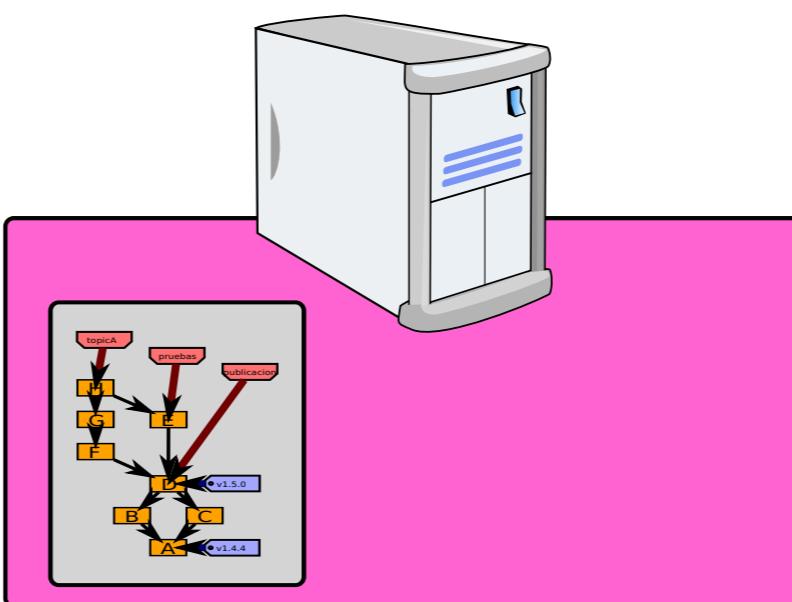


- sólo un servidor
  - todos dependen de él

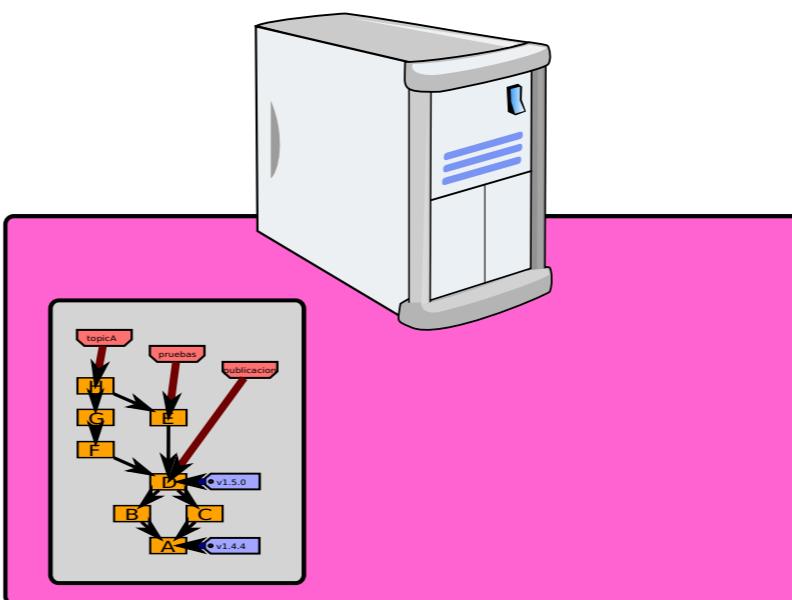
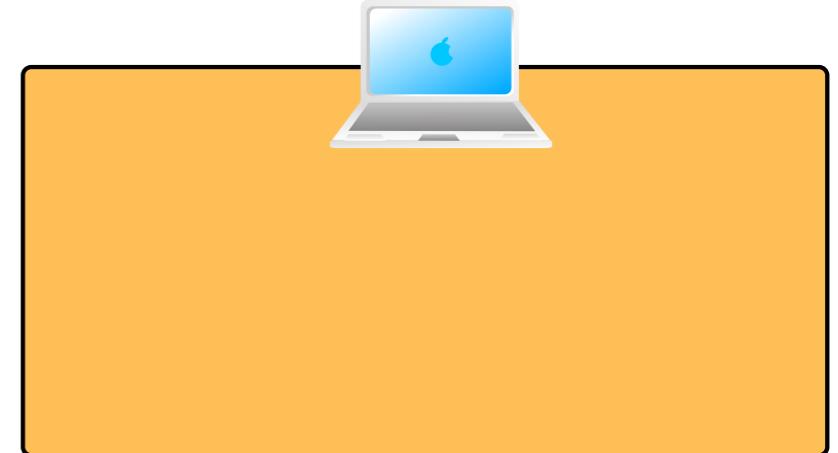


- sólo un servidor
  - todos dependen de él
  - cuello de botella

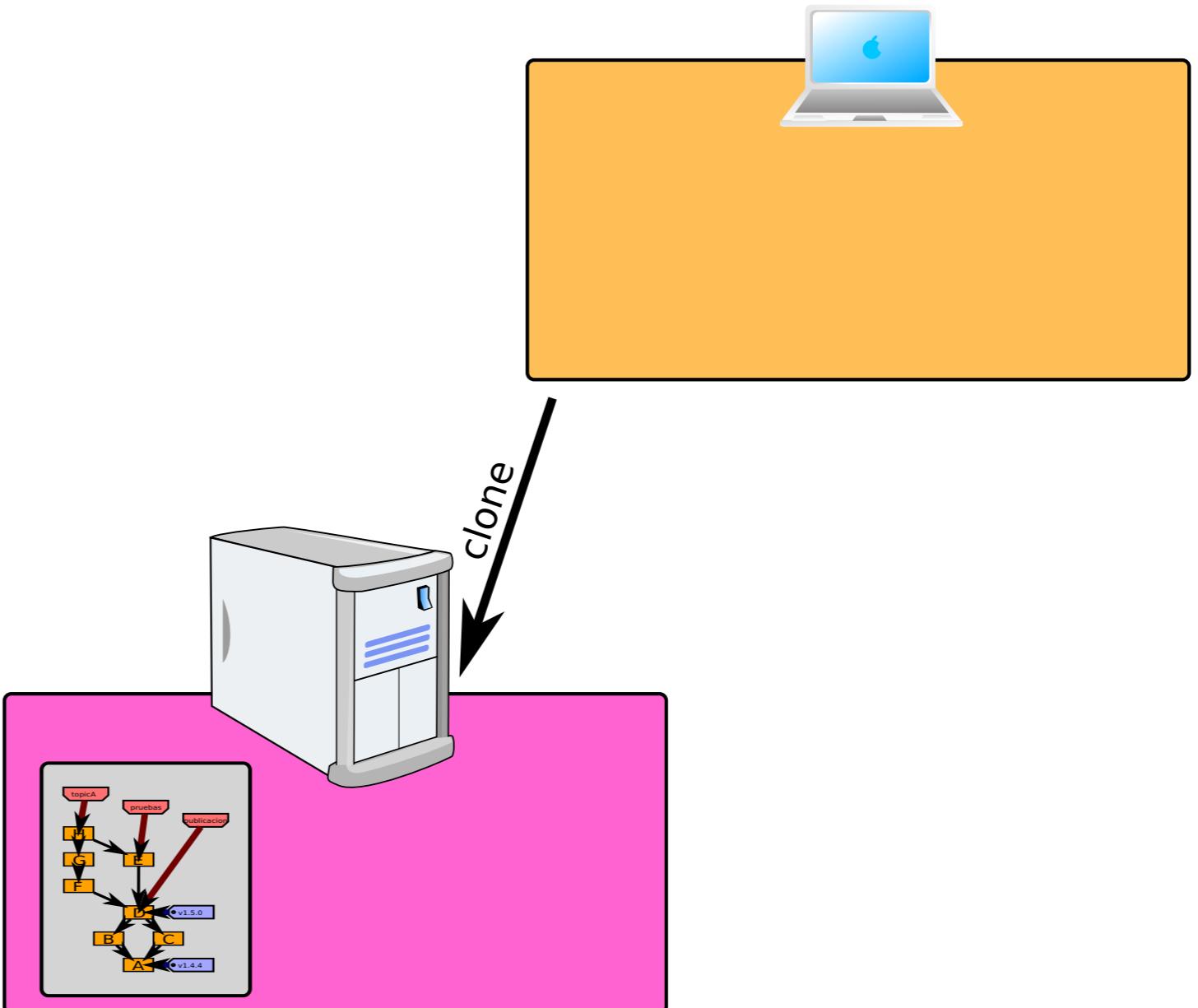
# SCV descentralizado o distribuido



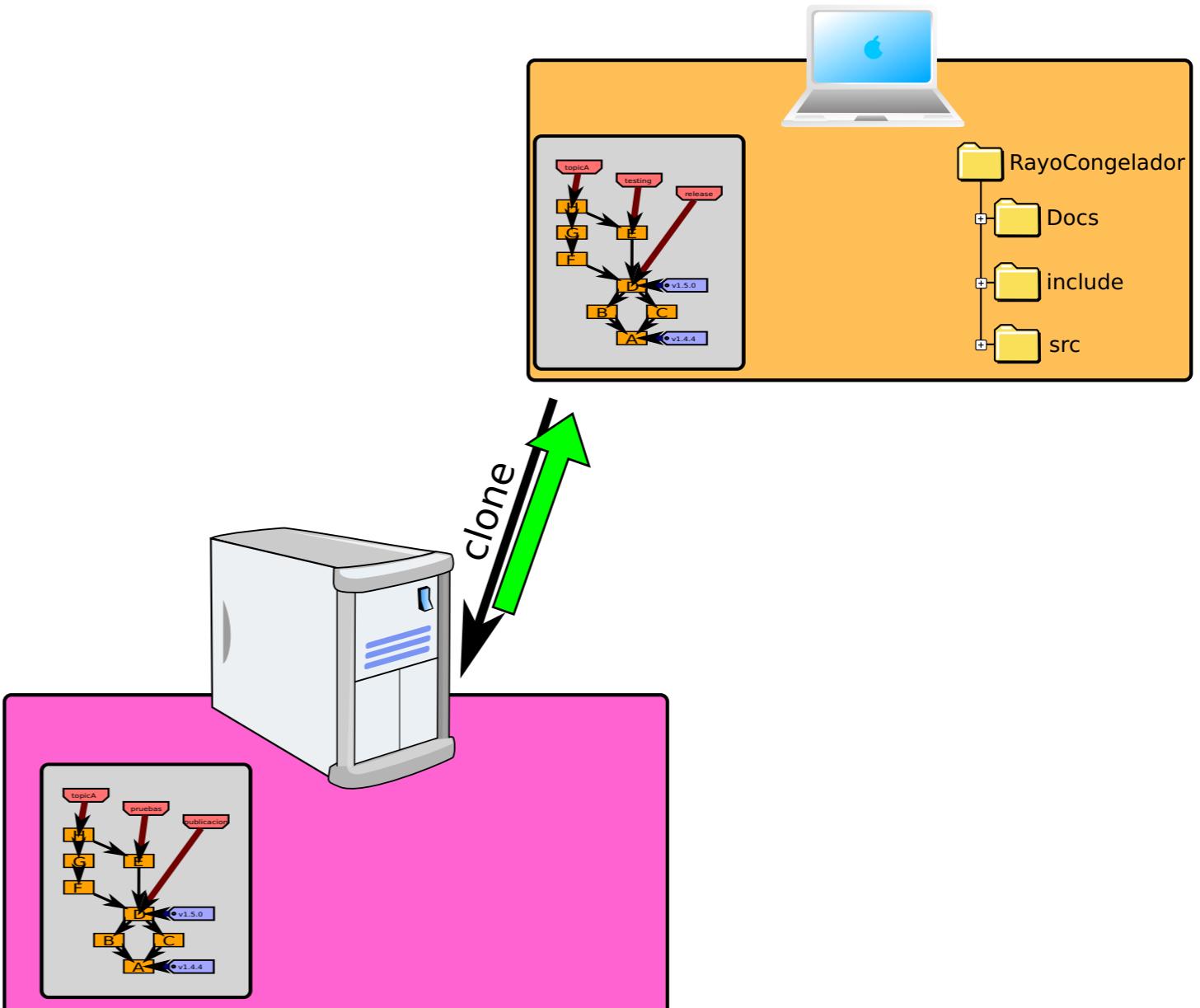
Distribuido



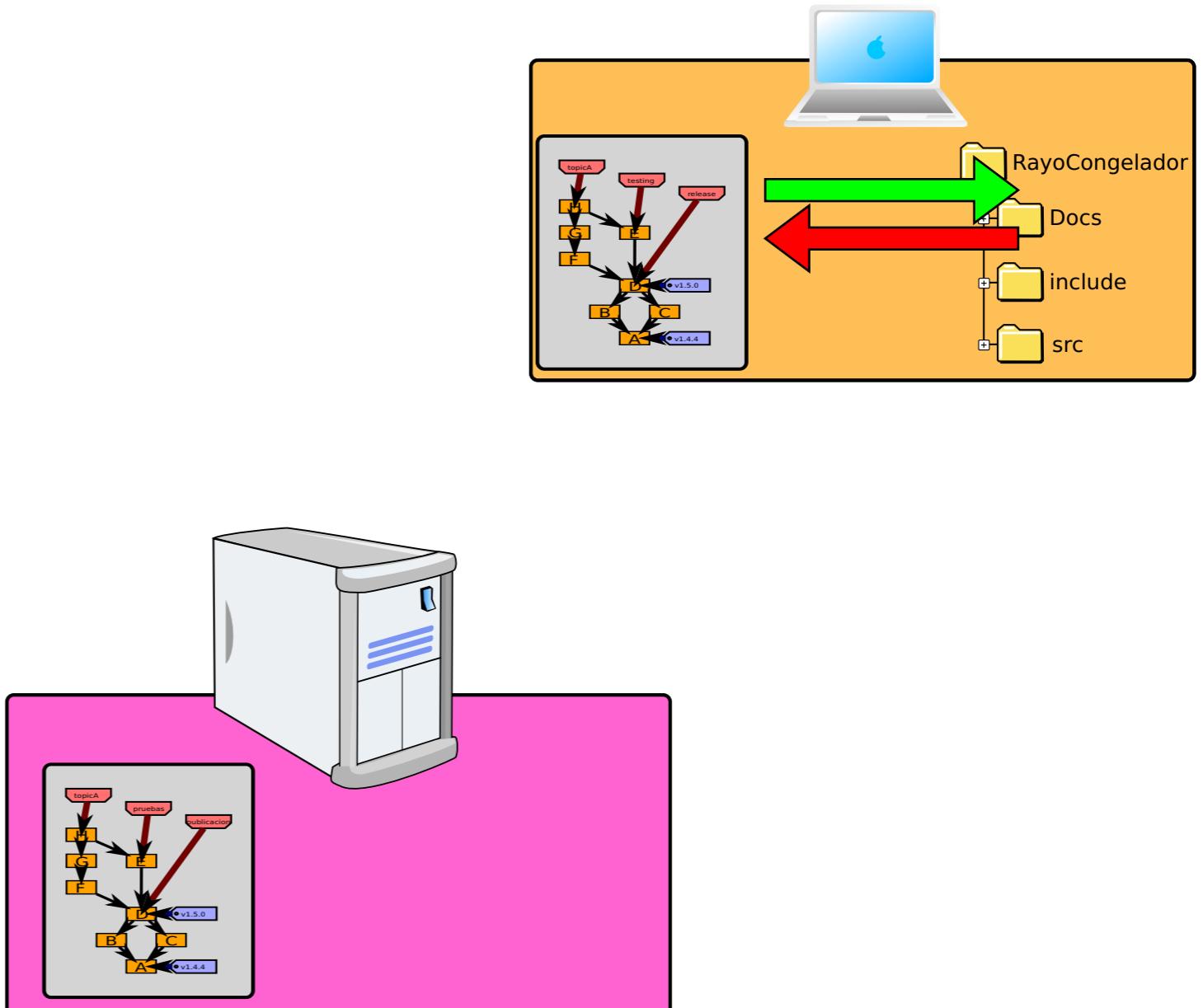
# Distribuido



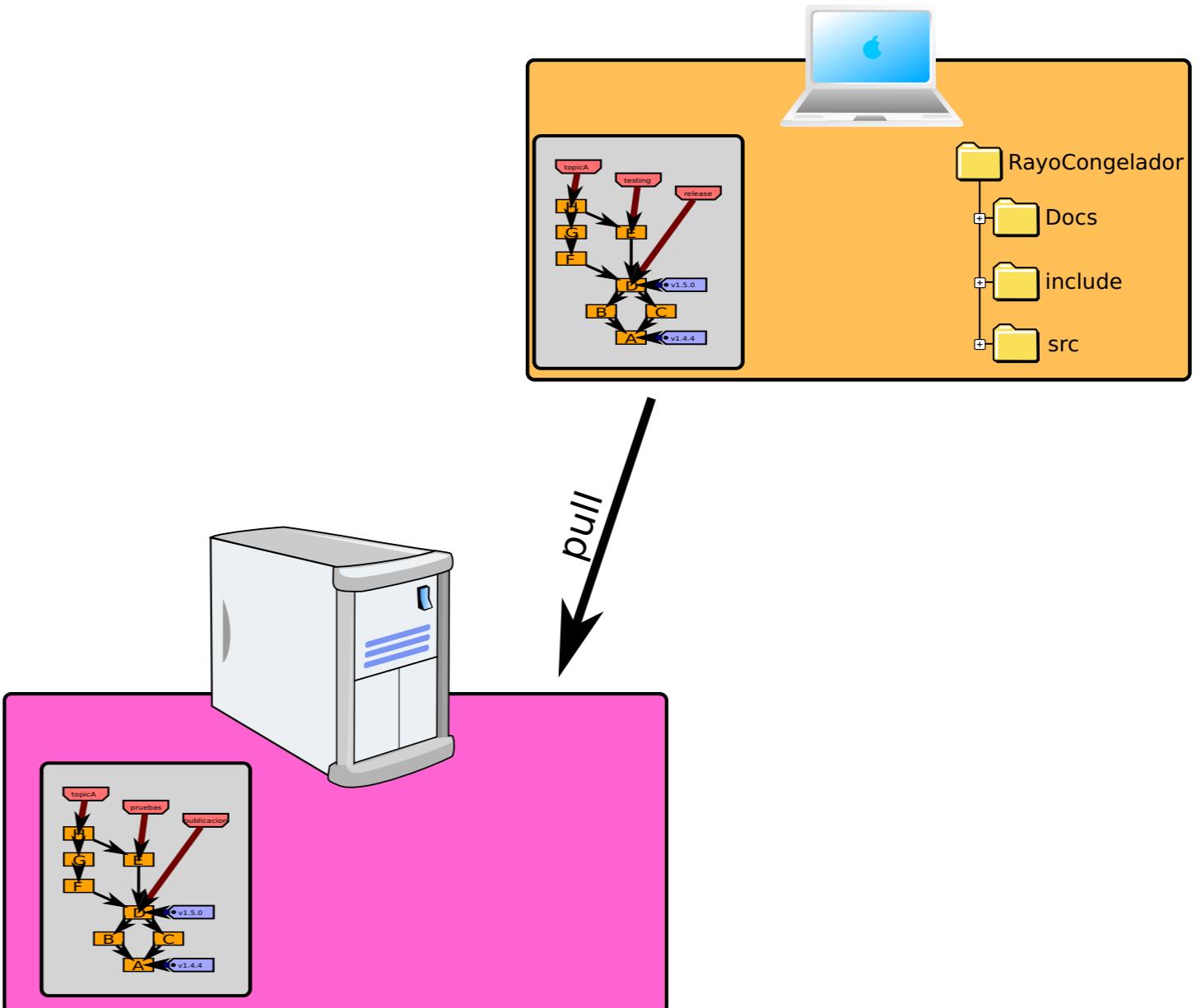
# Distribuido



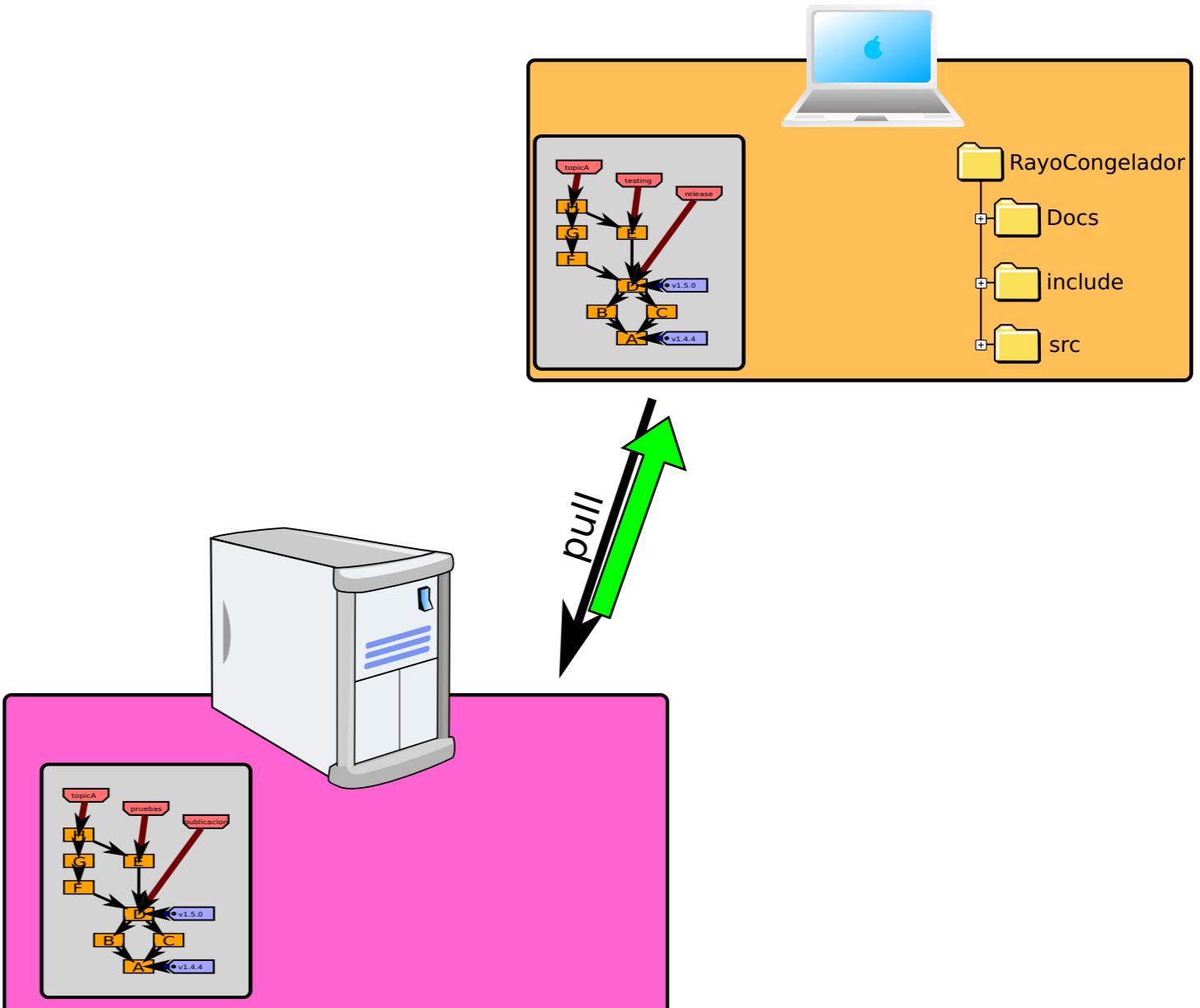
# Distribuido



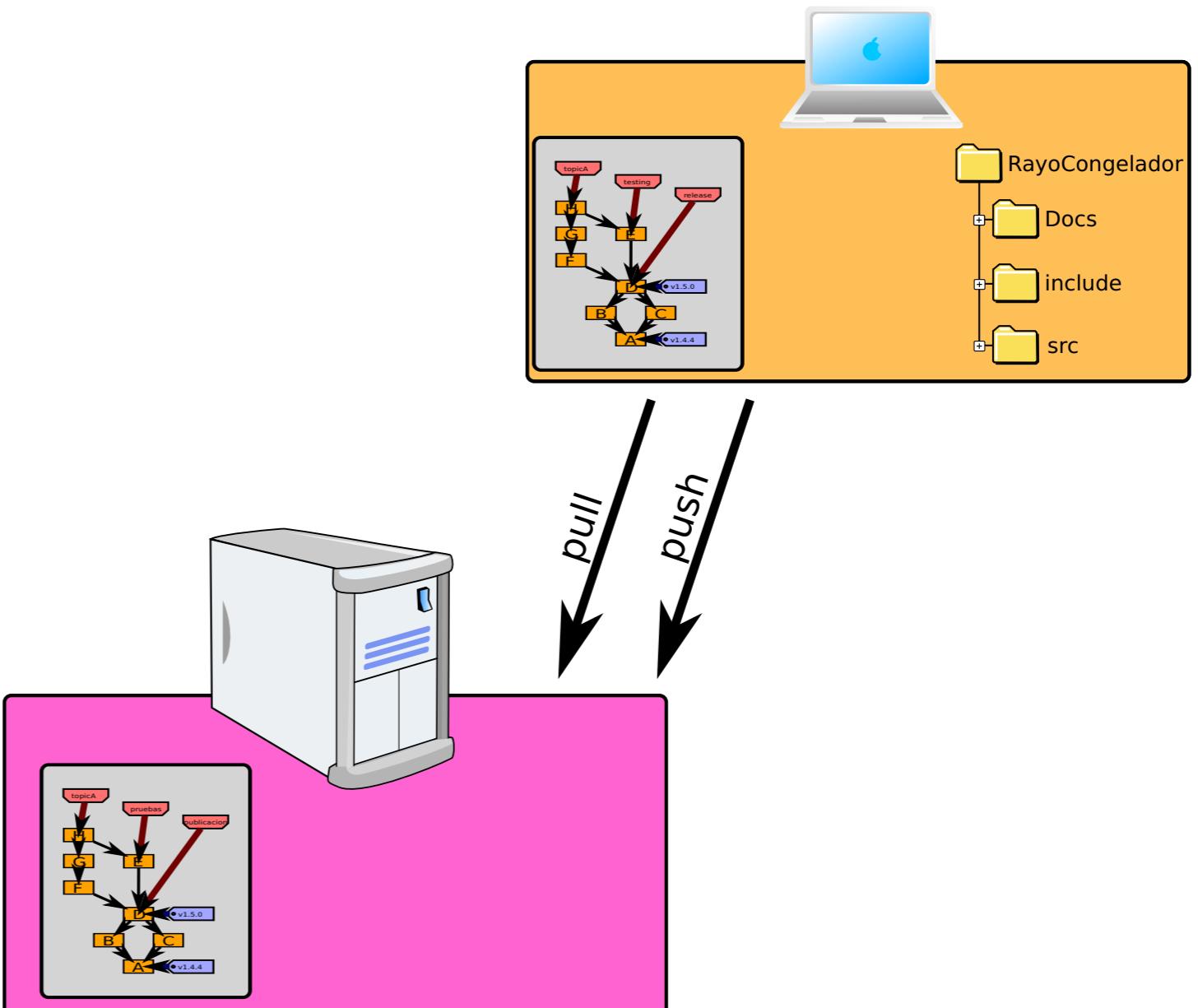
# Distribuido



# Distribuido

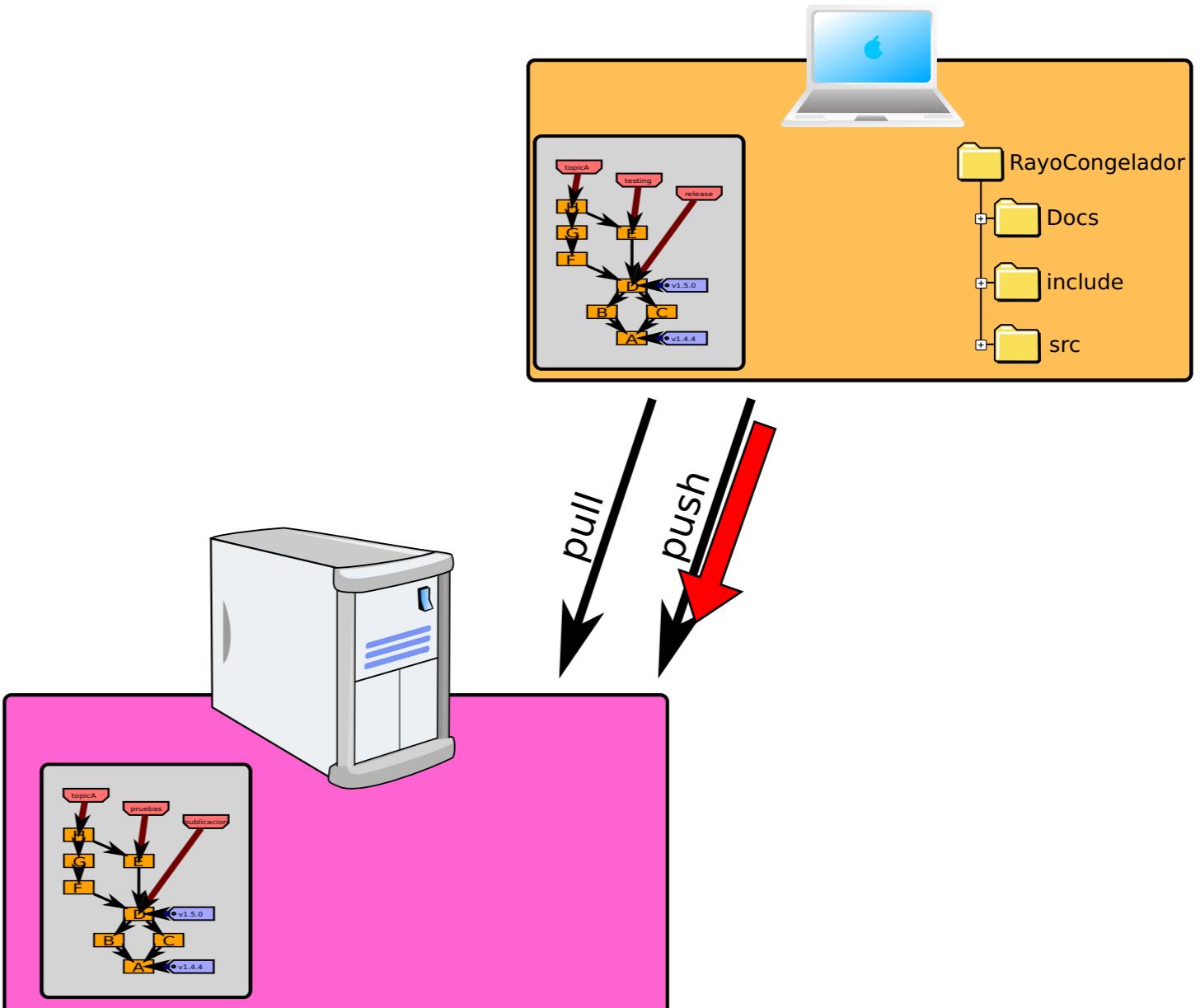


# Distribuido

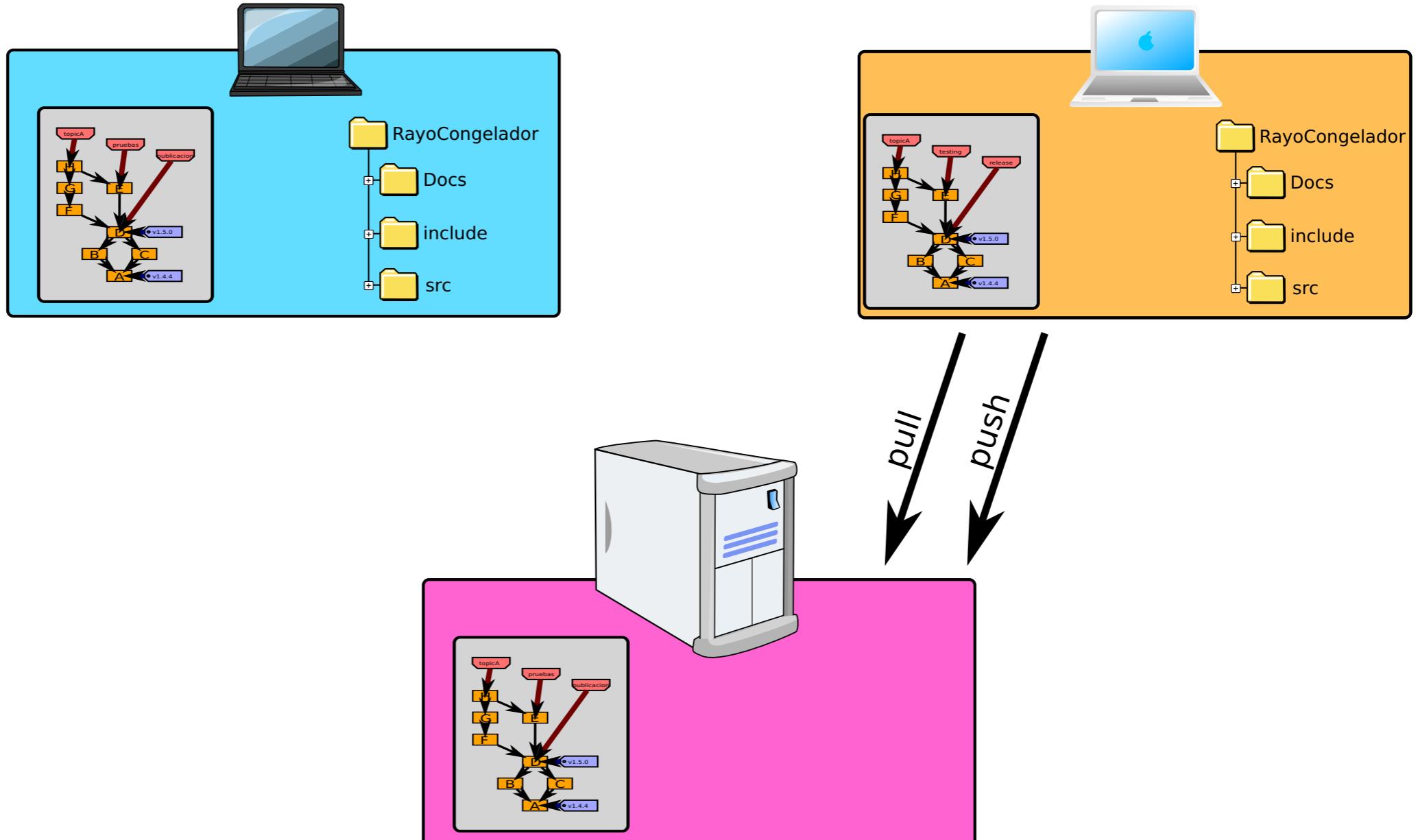


*pull*      *push*

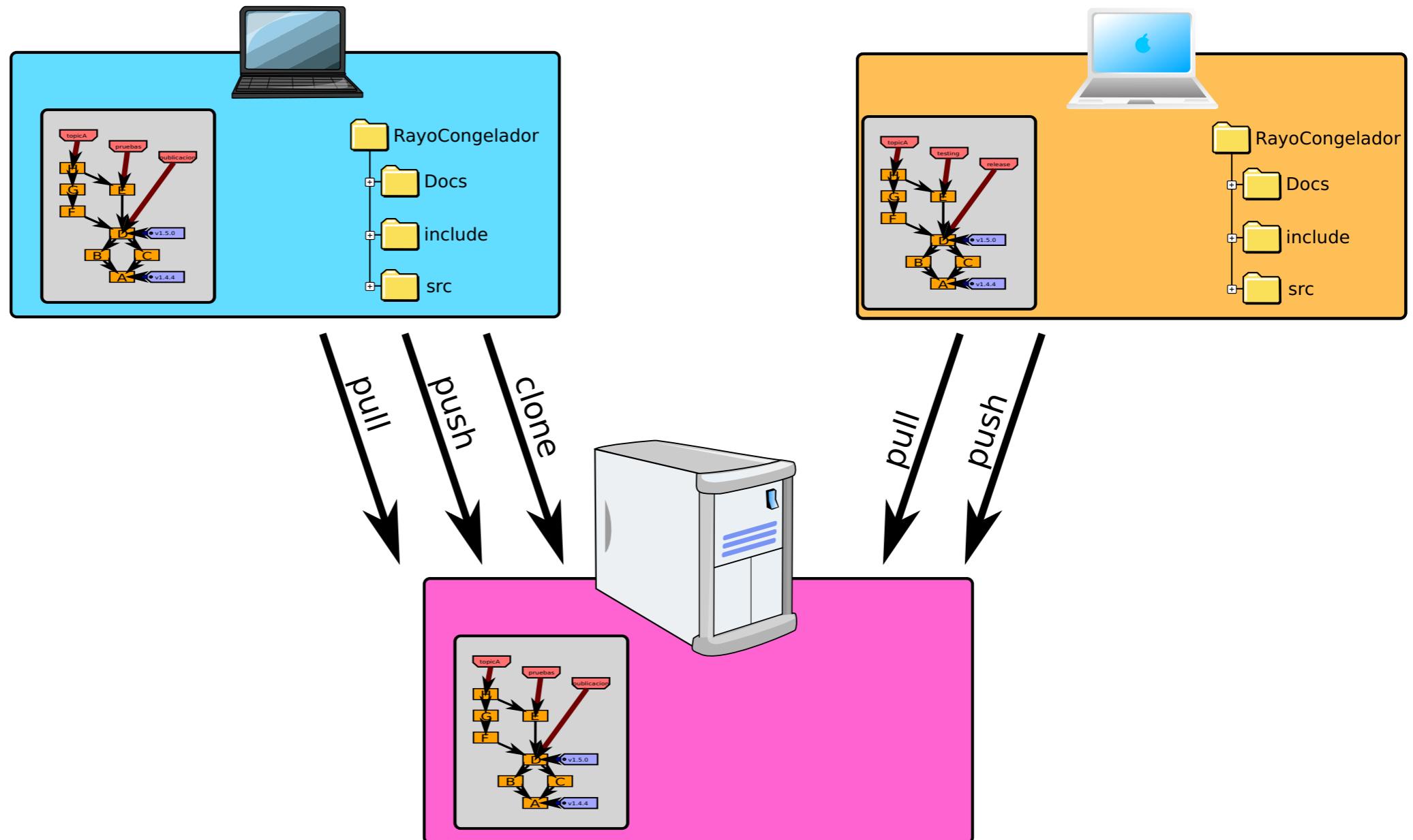
# Distribuido



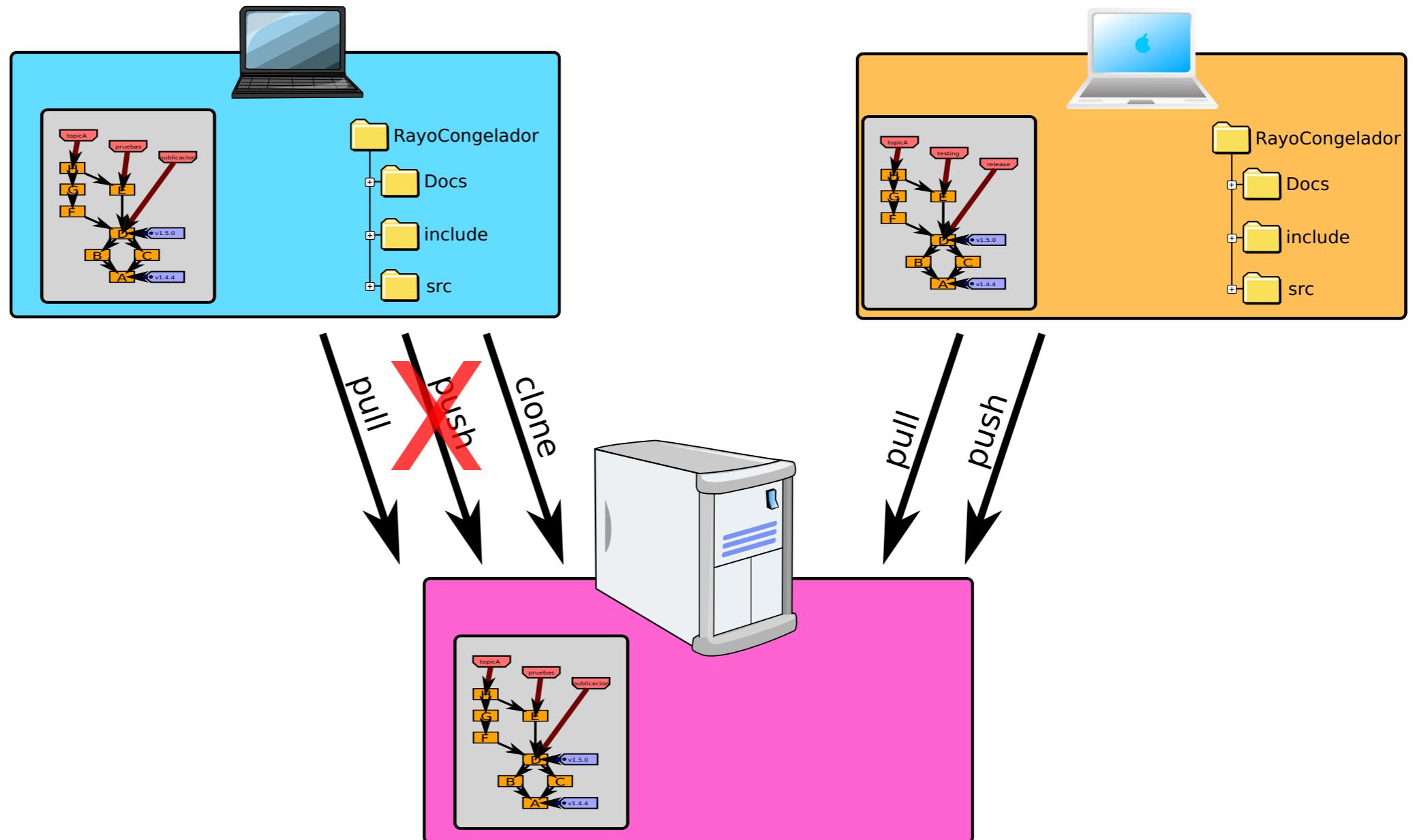
# Distribuido



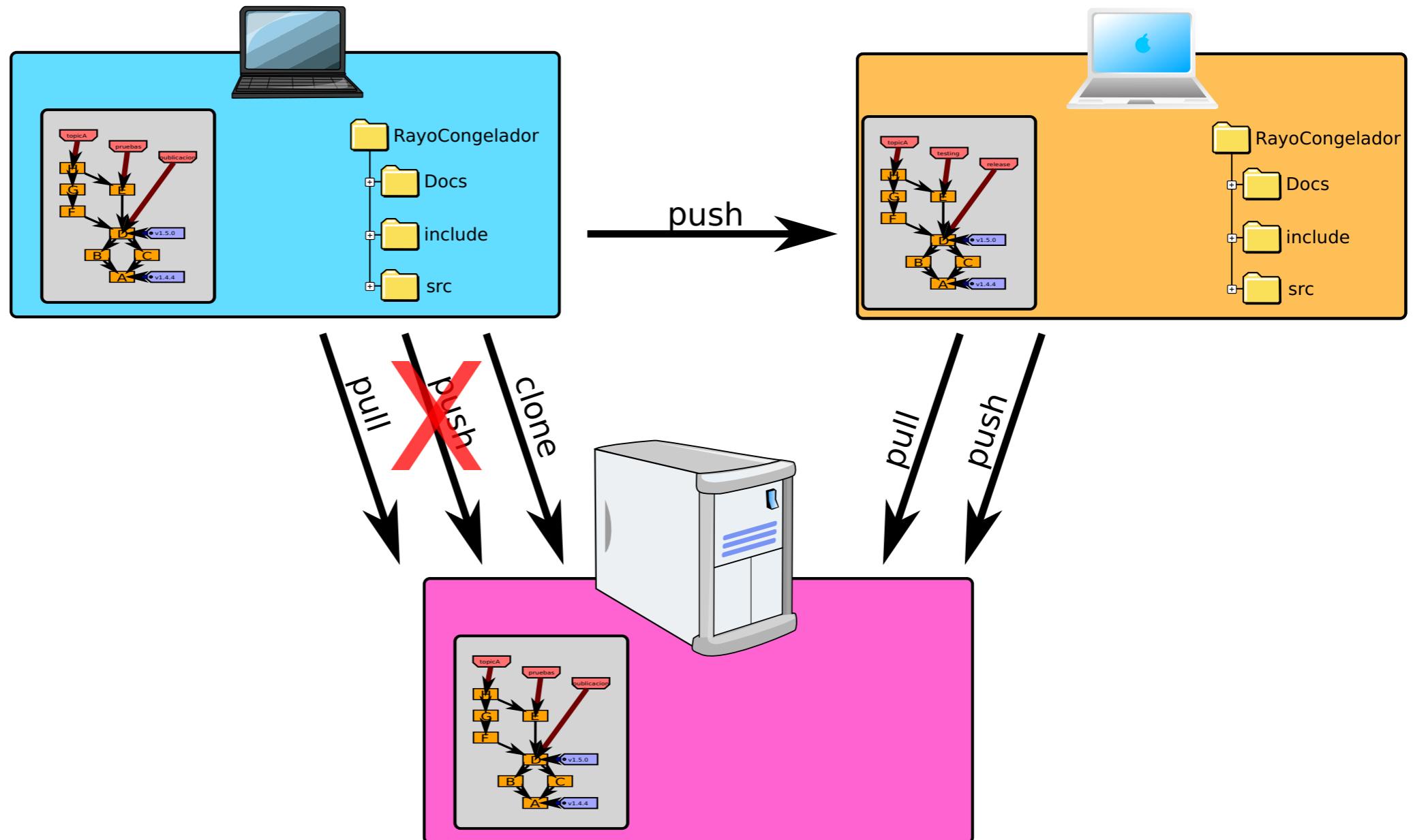
# Distribuido



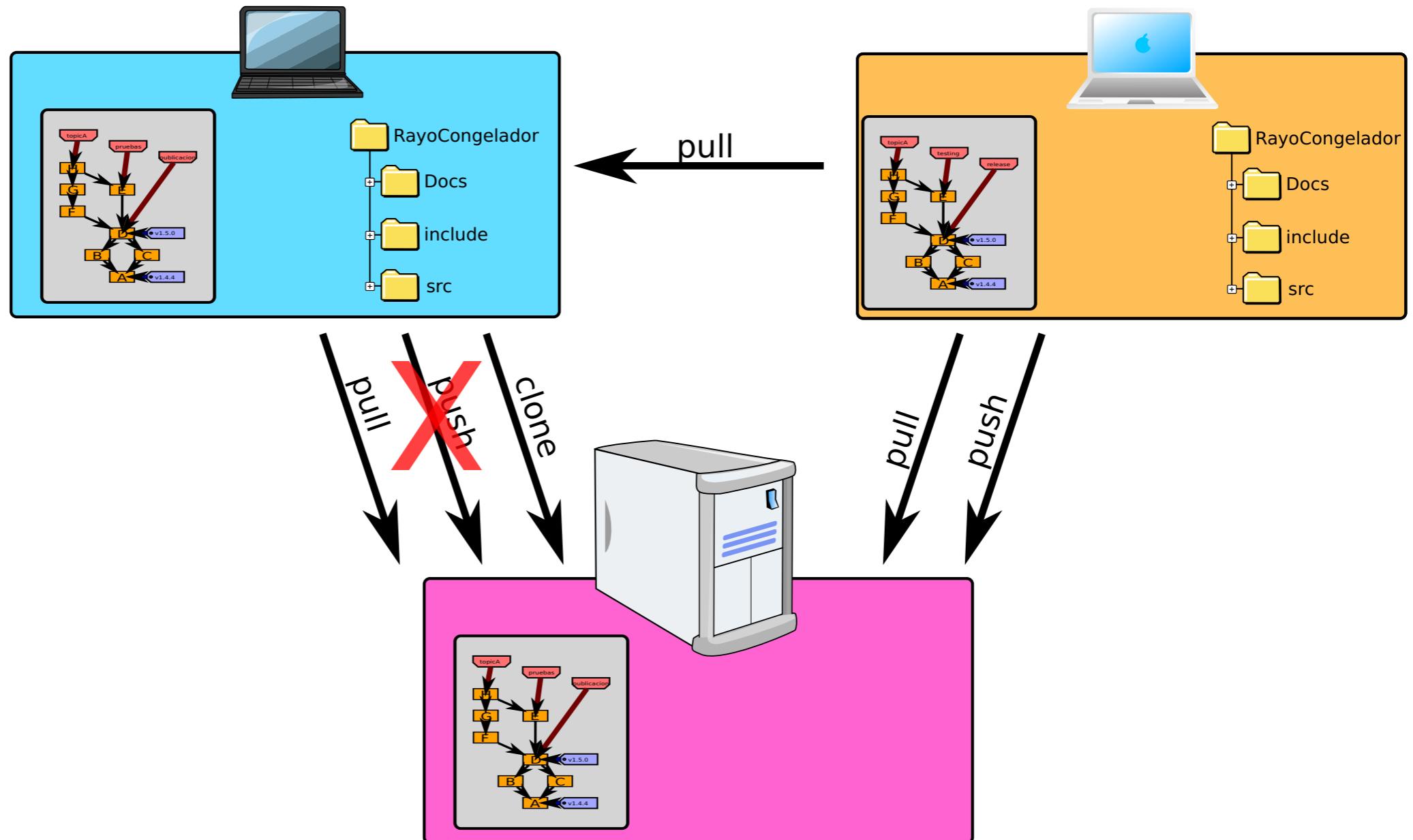
# Distribuido



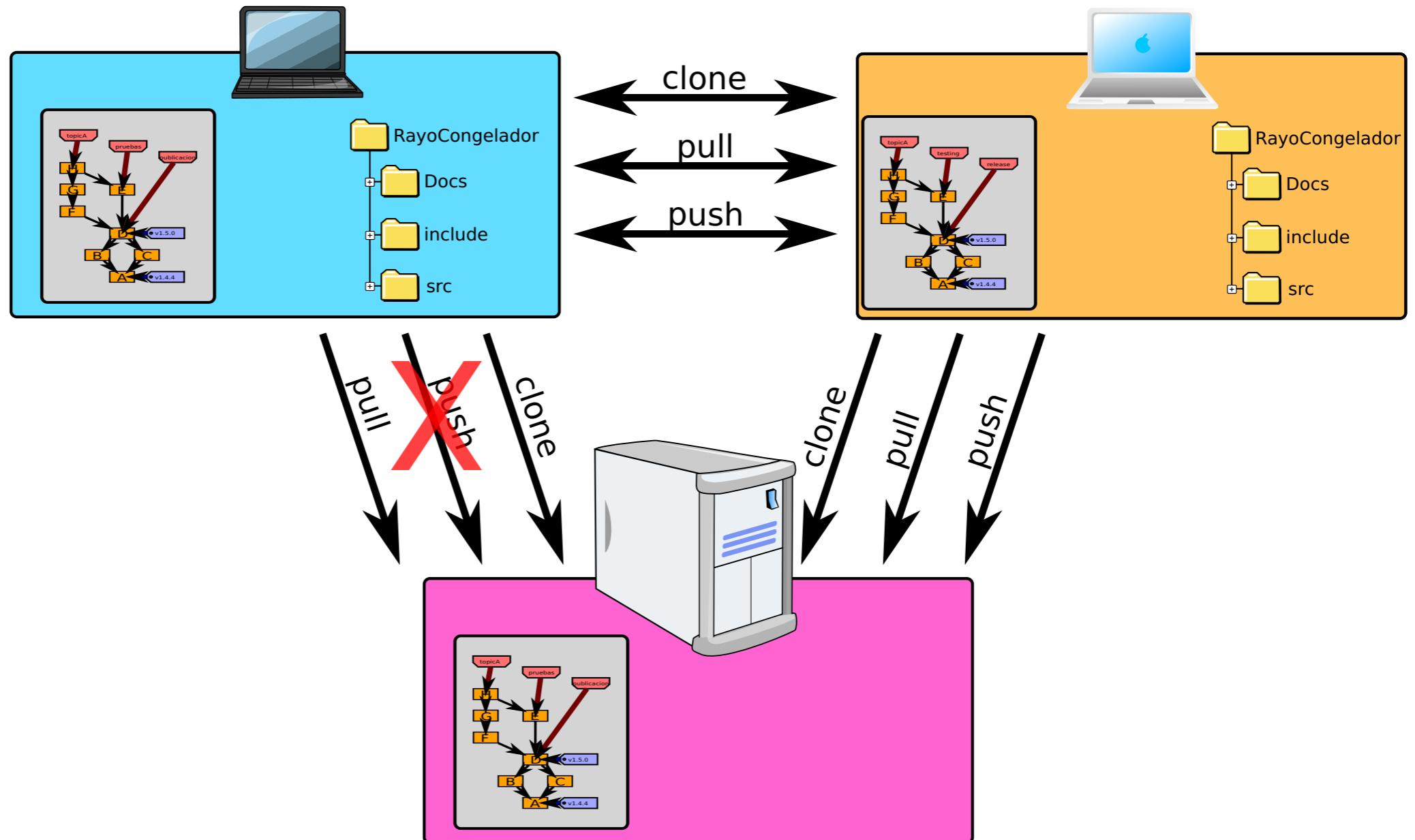
# Distribuido



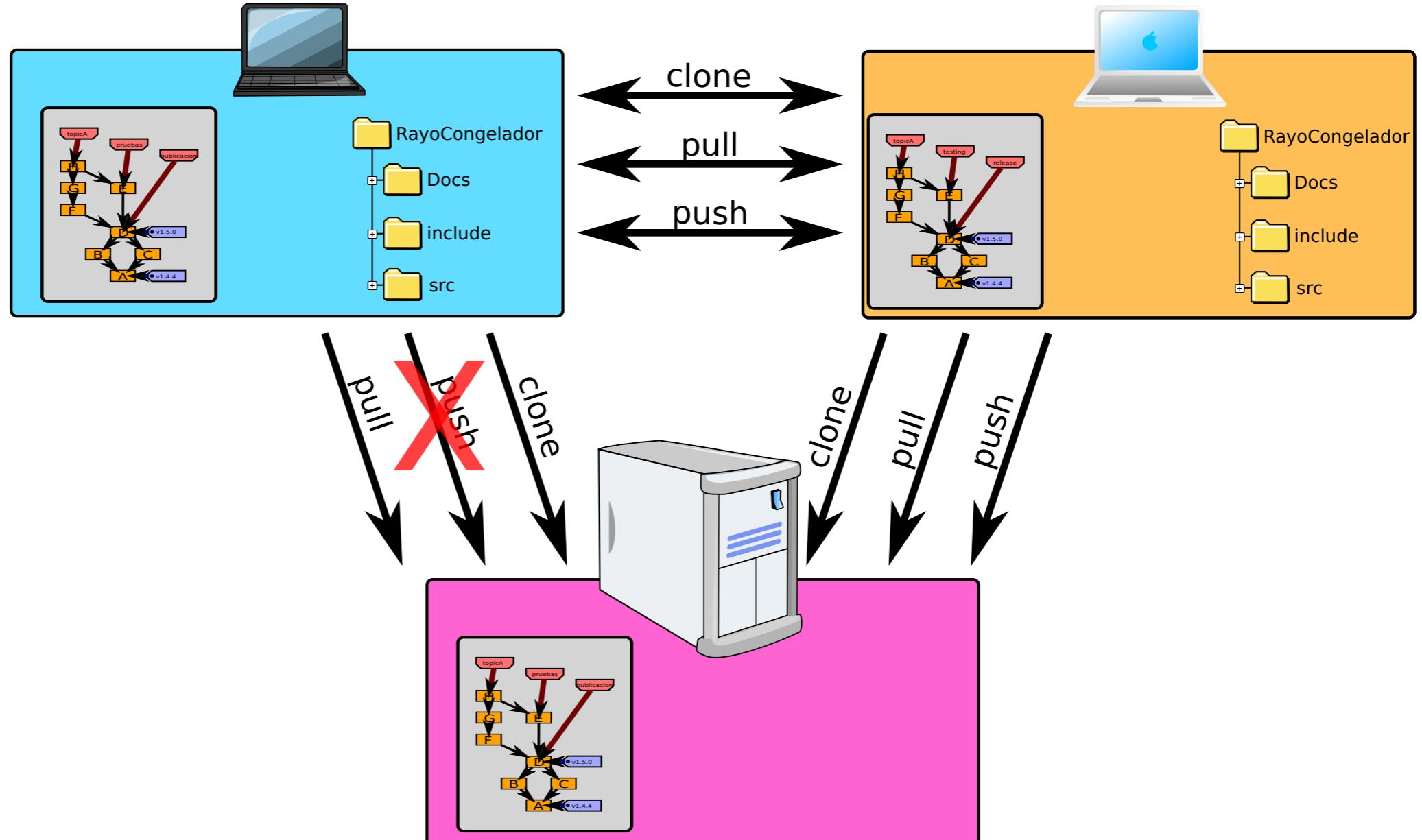
# Distribuido



# Distribuido



# Distribuido



cualquiera puede tener su servidor:  
la diferencia es de tipo social, no técnica

# Beneficios

Distribuido  
Beneficios

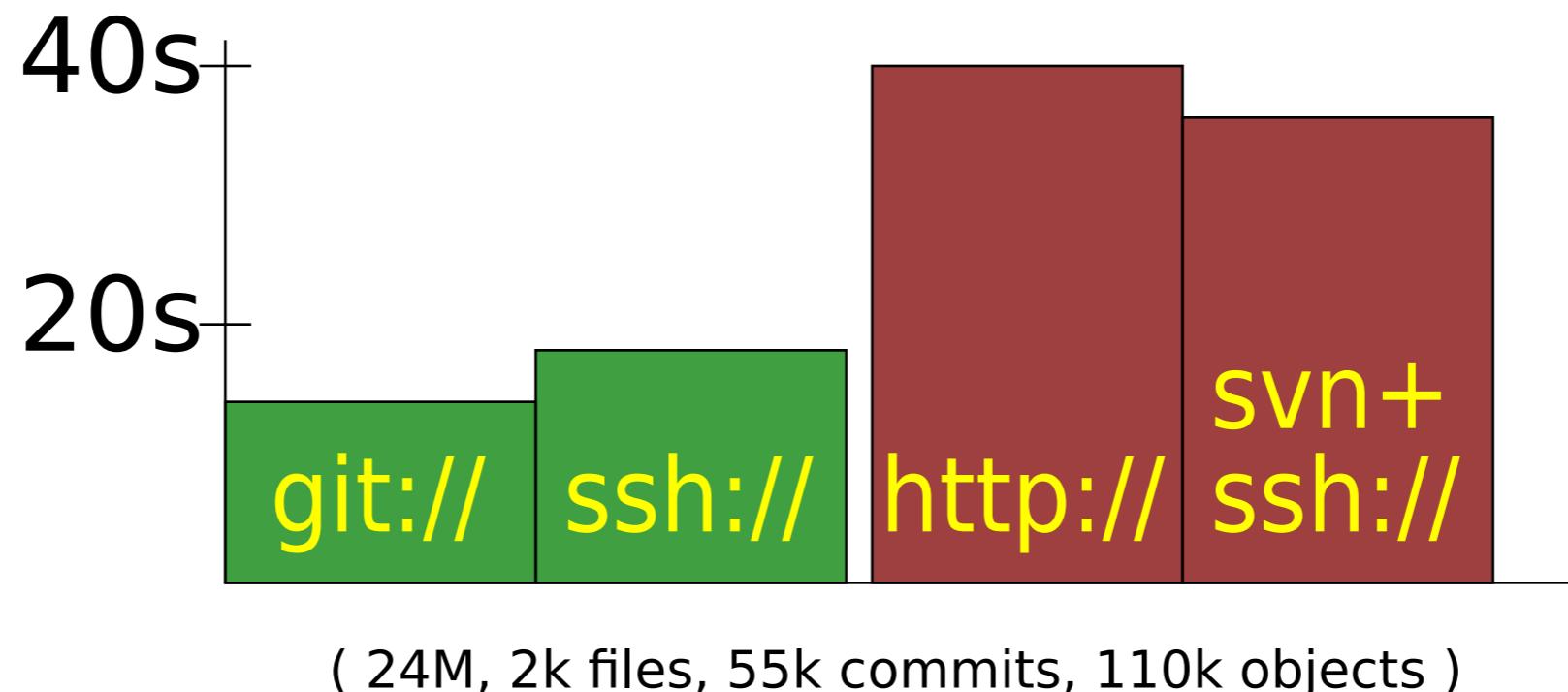
# Rapidez

# Rapidez

- repositorio local
  - no hay latencia por la red
  - E/S de disco bajo caché del SO:  
*cold/warm cache*

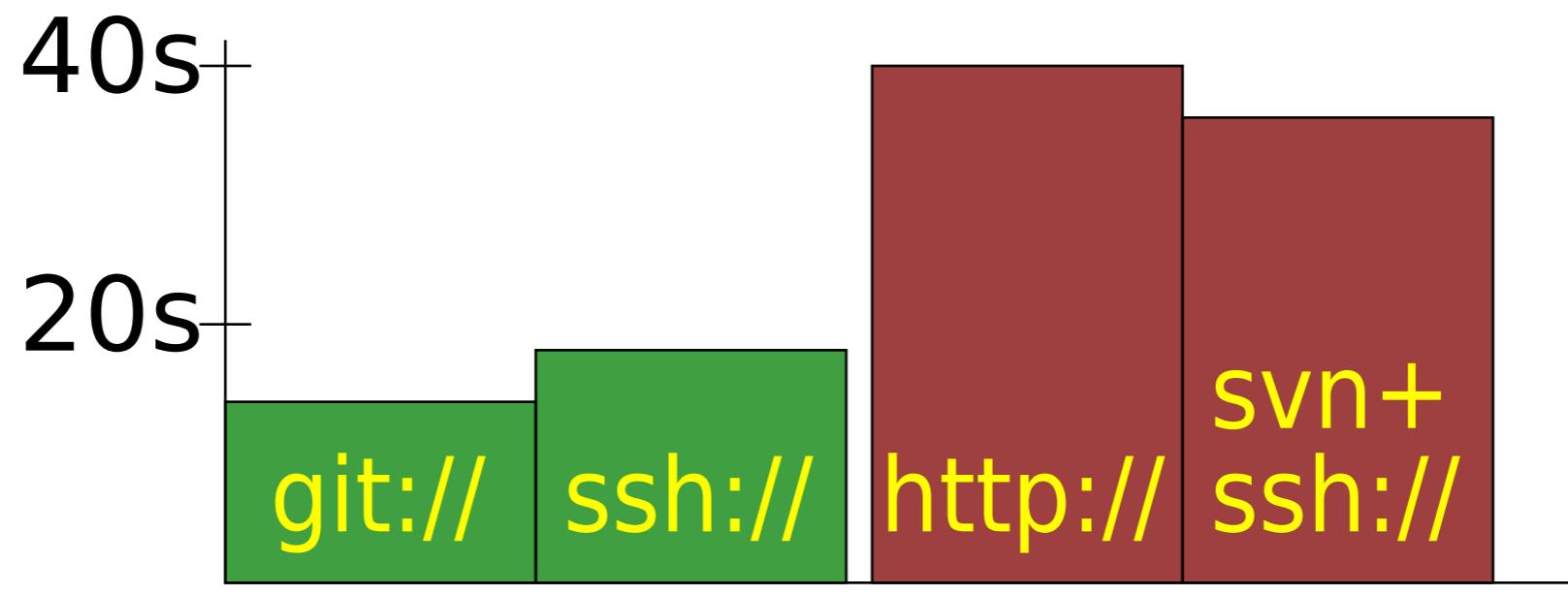
# Rapidez

- **git clone** a menudo más rápido que **svn co**



# Rapidez

- **git clone** a menudo más rápido que  
**svn co**



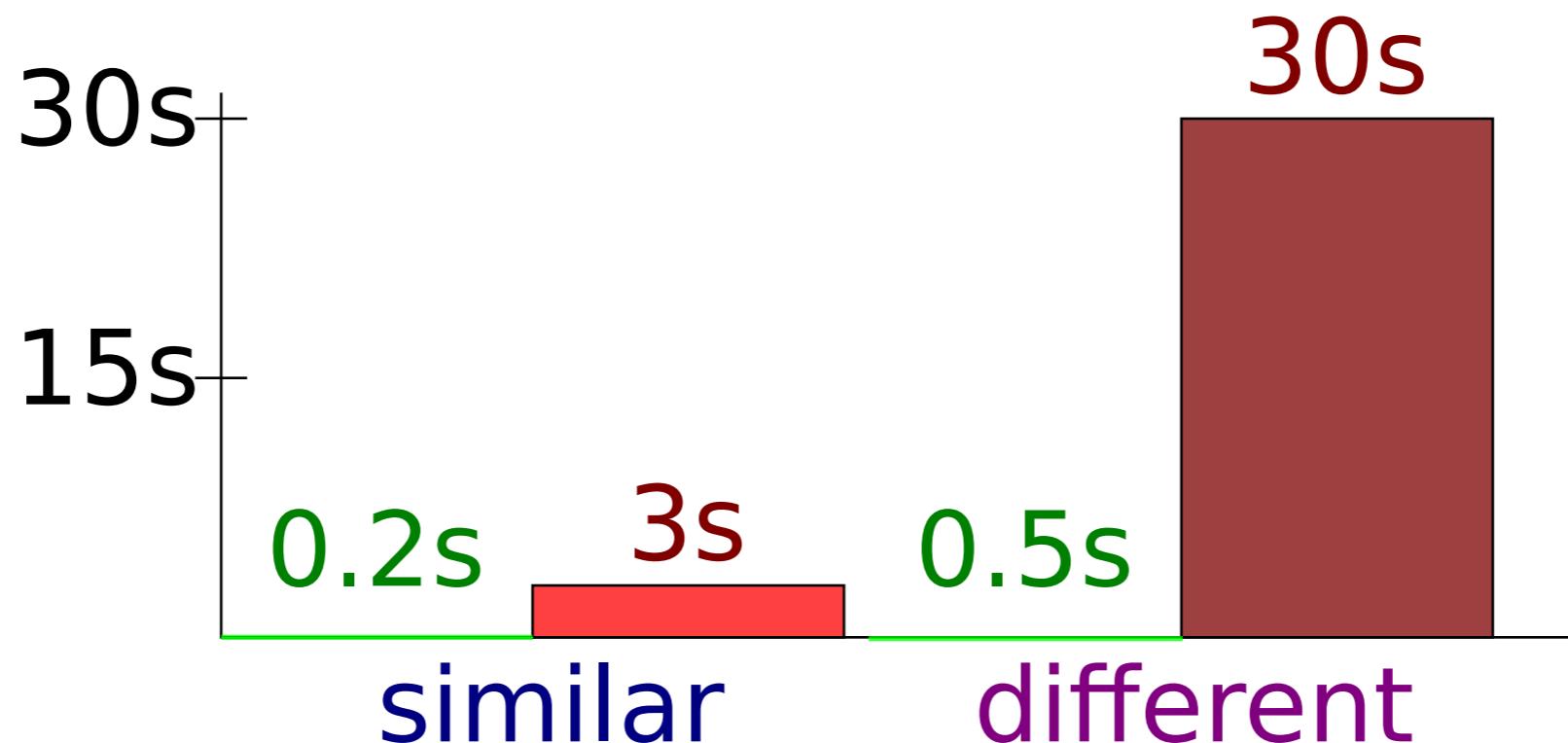
( 24M, 2k files, 55k commits, 110k objects )

( disk usage: 92M & 81M )

# Rapidez

- cambio de ramas

**git vs svn**



( 135 files changed, 1978 insertions, 3289 deletions )  
( 1588 files changed, 53145 insertions, 105176 deletions )

Distribuido  
Beneficios

# Micro-revisiones no intrusivas

# Micro-revisiones no intrusivas

- se crean a revisiones a menudo
- revisiones pequeñas y cohesivas

# Micro-revisiones no intrusivas

- se crean a revisiones a menudo
- revisiones pequeñas y cohesivas
- nunca se deja el código a medias para el resto de desarrolladores

# Micro-revisiones no intrusivas

- se crean a revisiones a menudo
  - revisiones pequeñas y cohesivas
  - nunca se deja el código a medias para el resto de desarrolladores
- ... y además ...
- permiten llevar experimentos aparte

# Micro-revisiones no intrusivas

- se crean a revisiones a menudo
- revisiones pequeñas y cohesivas
- nunca se deja el código a medias para el resto de desarrolladores

... y además ...

- permiten llevar experimentos aparte
- los que salgan mal pueden tirarse sin consecuencias

Distribuido  
Beneficios

# Trabajo sin conexión

# Trabajo sin conexión

- en el tren, avión, autobús, etc.

# Trabajo sin conexión

- en el tren, avión, autobús, etc.
- sobre ramas "congeladas" o sin permisos de escritura

# Trabajo sin conexión

- en el tren, avión, autobús, etc.
- sobre ramas "congeladas" o sin permisos de escritura
- aunque la VPN o la red no funcionen

# Trabajo sin conexión

- en el tren, avión, autobús, etc.
- sobre ramas "congeladas" o sin permisos de escritura
- aunque la VPN o la red no funcionen

...¡incluso se puede colaborar!

Distribuido  
Beneficios

# Robustez

# Robustez

¿Y si hay un fallo de disco duro en el servidor, o deja de funcionar por otro motivo?

- Centralizados: todo se para en seco hasta que se recupere la copia de seguridad (si la hay) y se levanta el servidor.
- Distribuidos: se colabora rodeando el problema sin más.

- Introducción
- Órdenes
- Uso general
- Uso avanzado

- Introducción
- Órdenes
- Uso general
- Uso avanzado

- Introducción
- Órdenes
- Uso general
- Uso avanzado

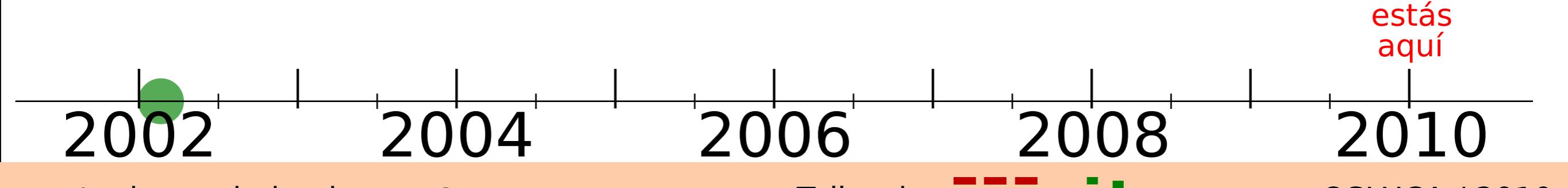
- Introducción
- Órdenes
- Uso general
- Uso avanzado

- Introducción
- Órdenes
- Uso general
- Uso avanzado

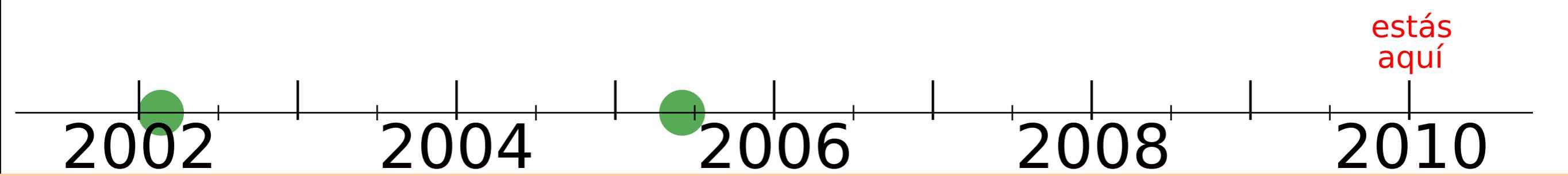
# Historia

## - BitKeeper

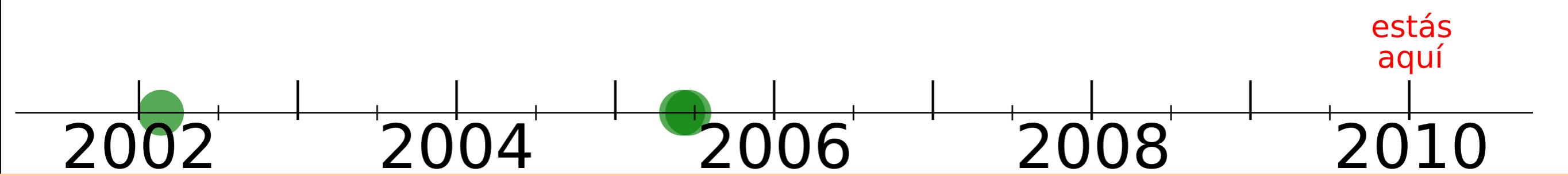
Febrero 2002



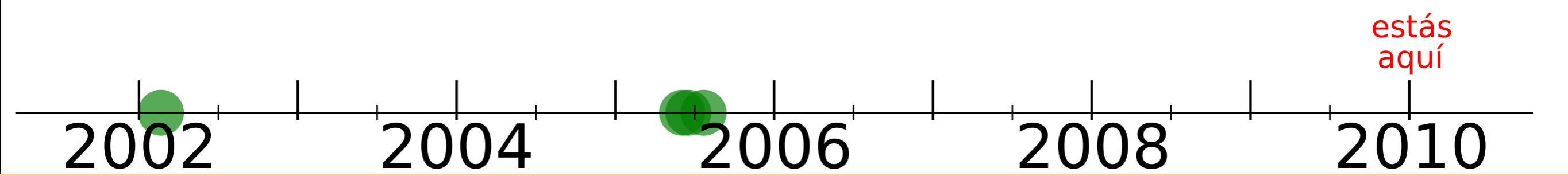
- BitKeeper  
Febrero 2002
- se deja de usar BK  
6 abril, 2005



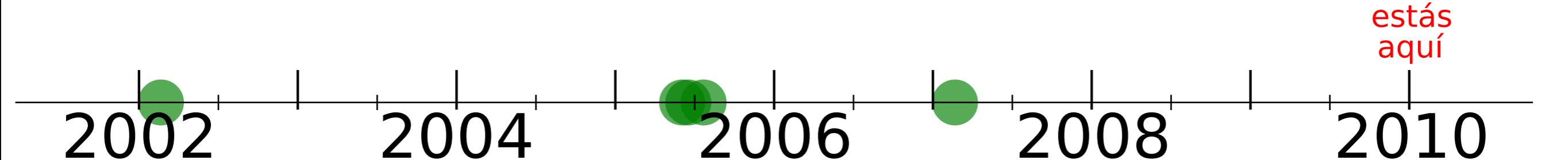
- BitKeeper  
Febrero 2002
- se deja de usar BK  
6 abril, 2005
- Git reúne ramas  
18 abril, 2005



- BitKeeper  
Febrero 2002
- se deja de usar BK  
6 abril, 2005
- Git reúne ramas  
18 abril, 2005
- Git se usa para desarrollar Linux  
16 junio, 2005



- BitKeeper  
Febrero 2002
- se deja de usar BK  
6 abril, 2005
- Git reúne ramas  
18 abril, 2005
- Git se usa para desarrollar Linux  
16 junio, 2005
- Git 1.5.0  
14 febrero, 2007



- BitKeeper  
Febrero 2002
- se deja de usar BK  
6 abril, 2005
- Git reúne ramas  
18 abril, 2005
- Git se usa para desarrollar Linux  
16 junio, 2005
- Git 1.5.0  
14 febrero, 2007
- Git 1.7.0.2  
7 marzo, 2010

estás  
aquí

2002

2004

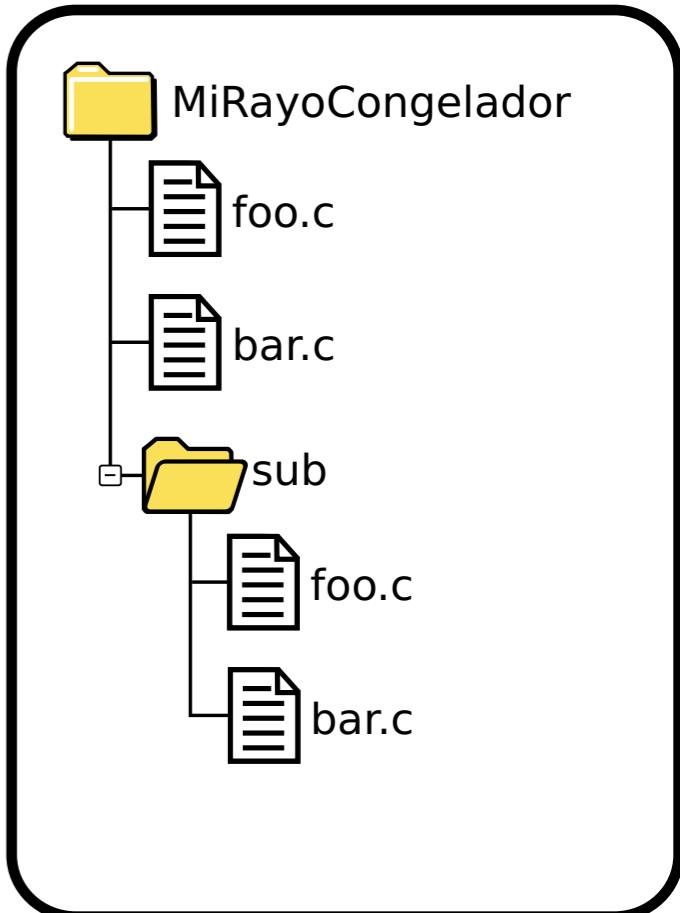
2006

2008

2010

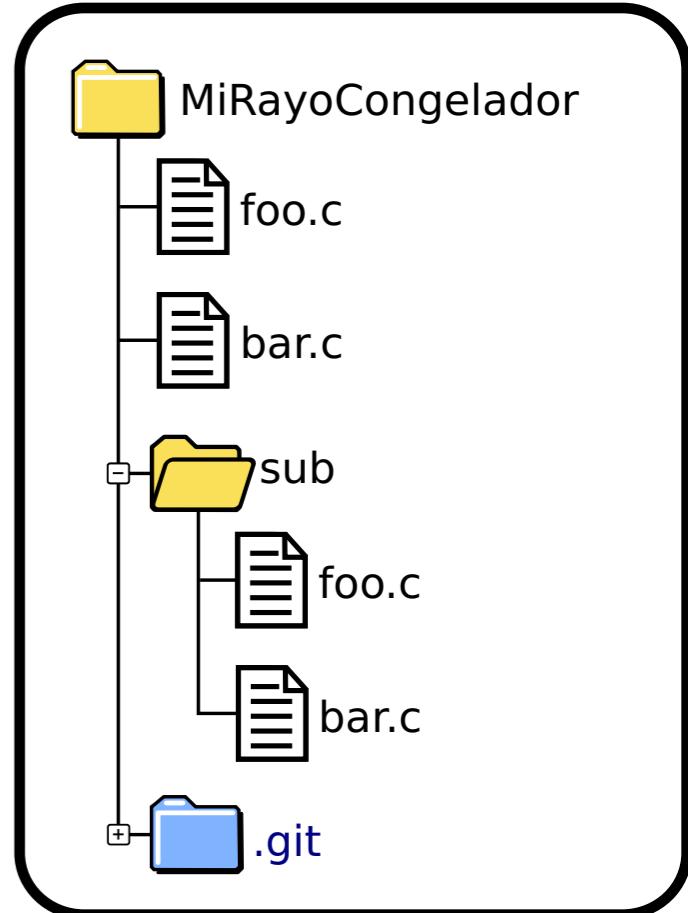
# Preparación de revisiones ("staging")

# Árbol de trabajo



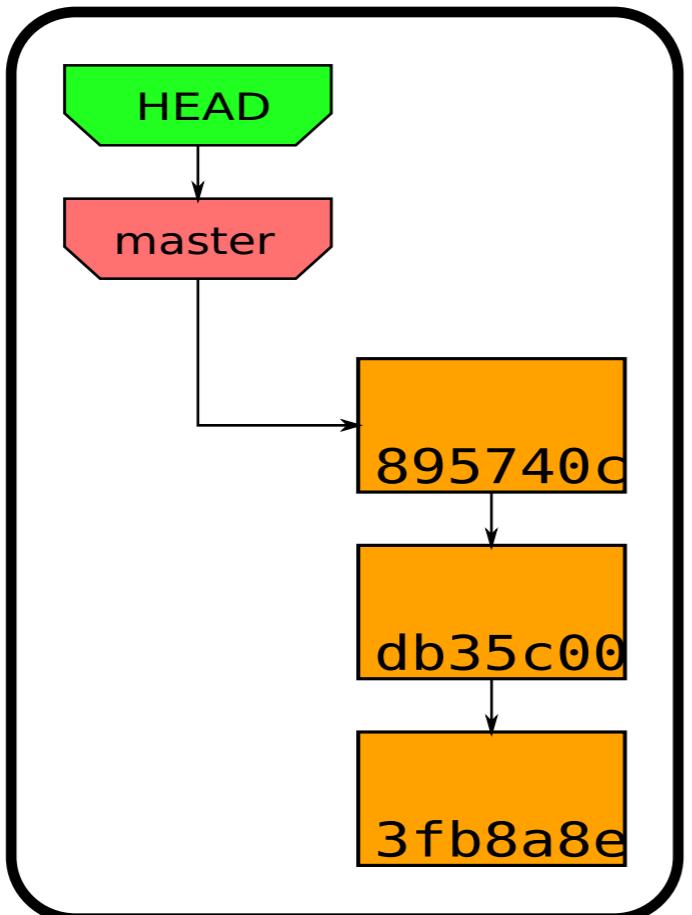
## ficheros editados

# Árbol de trabajo

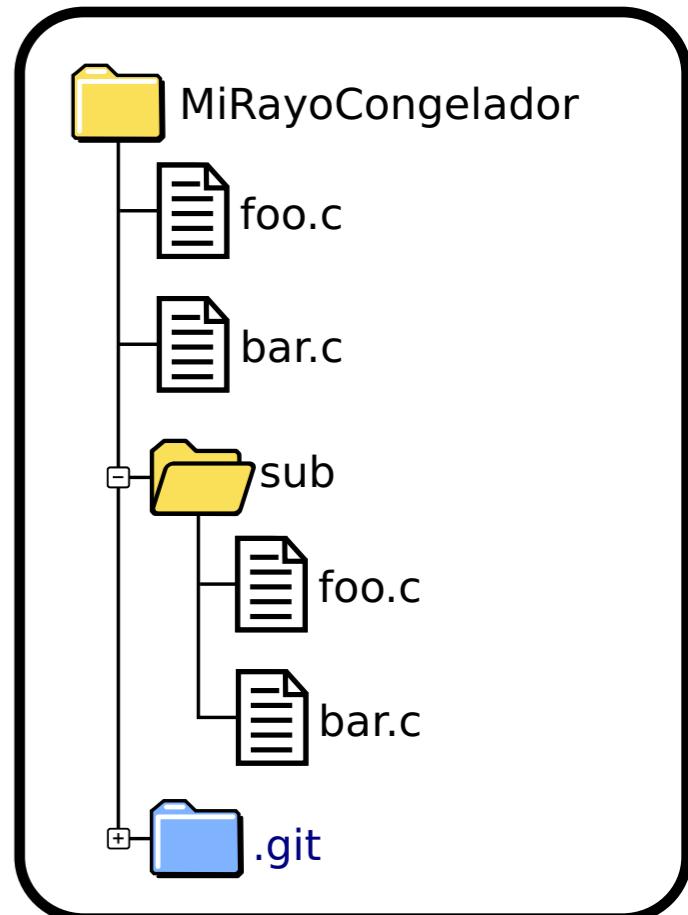


## ficheros de Git

# Repositorio

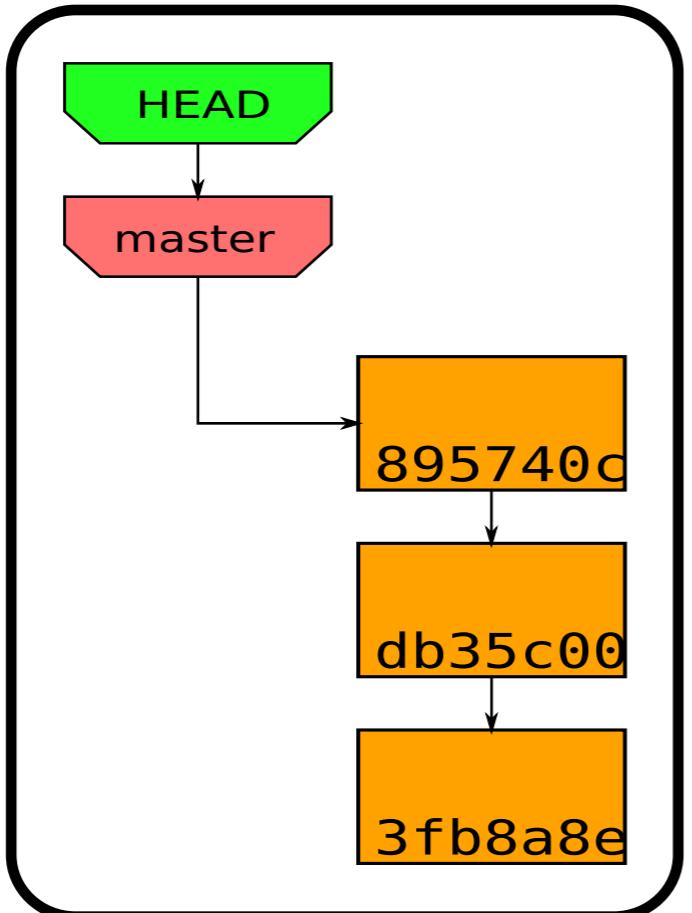


# Árbol de trabajo

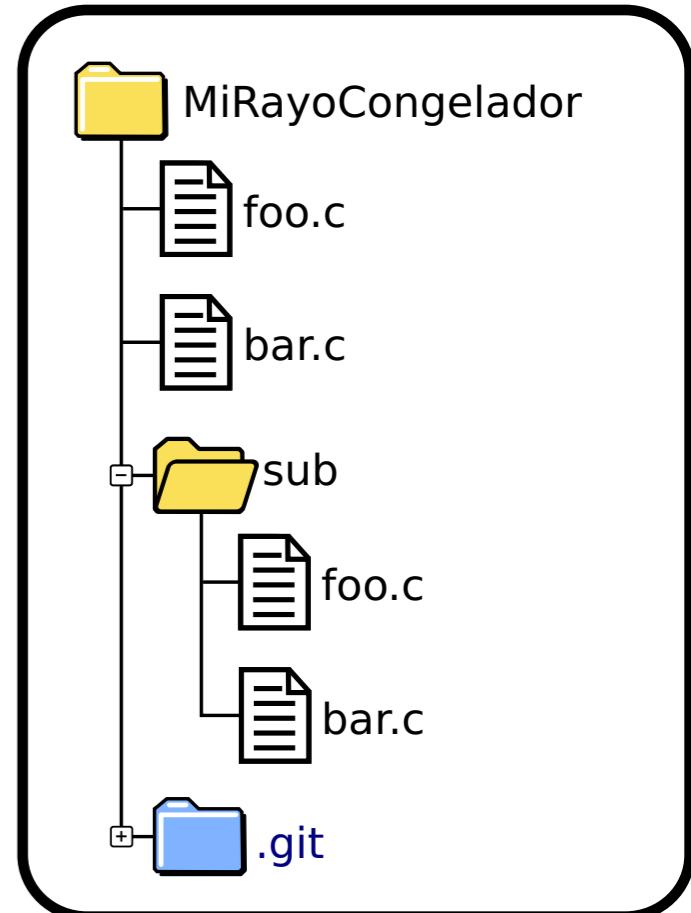


el *GOA*  
Grafo Orientado Acíclico

# Repositorio



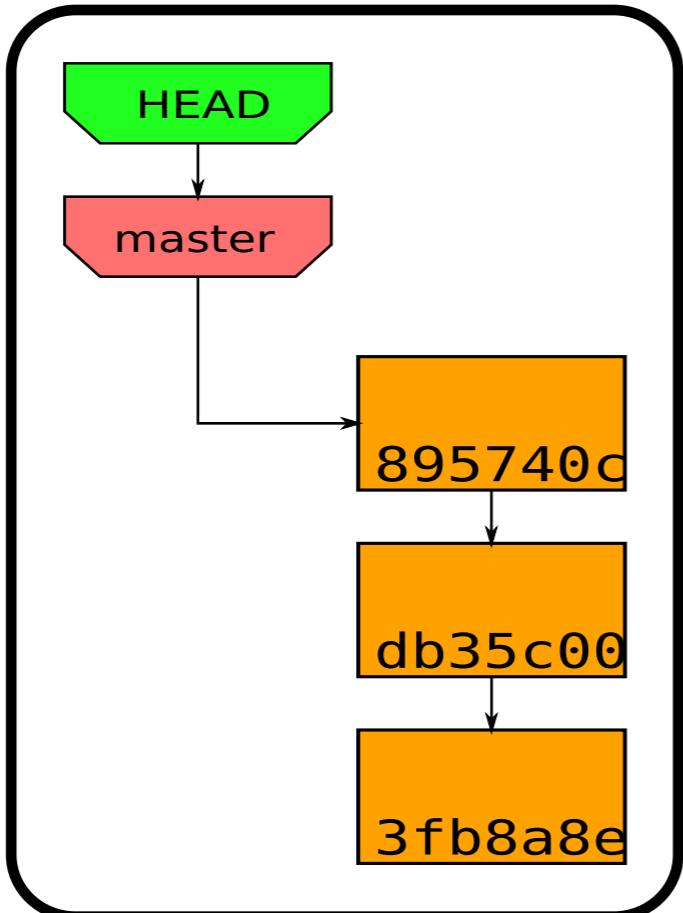
# Árbol de trabajo



contiene el historial

# Intro Preparación

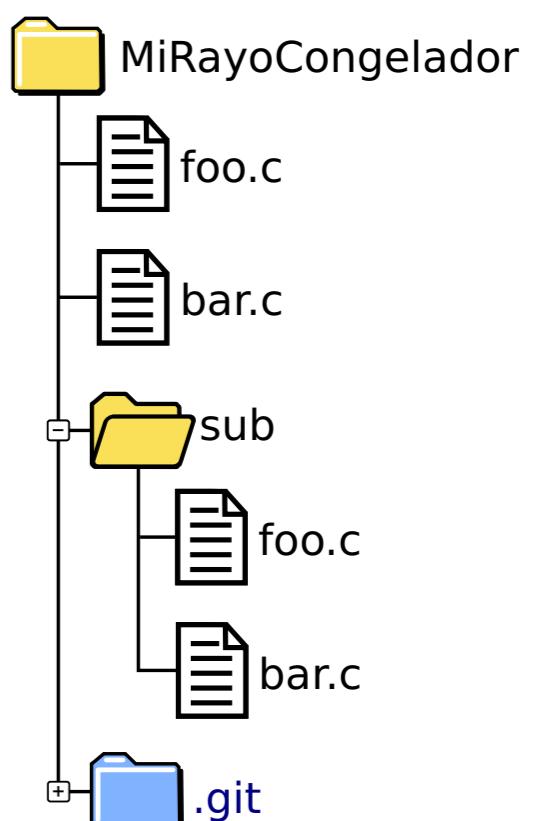
## Repositorio



## Área de preparación

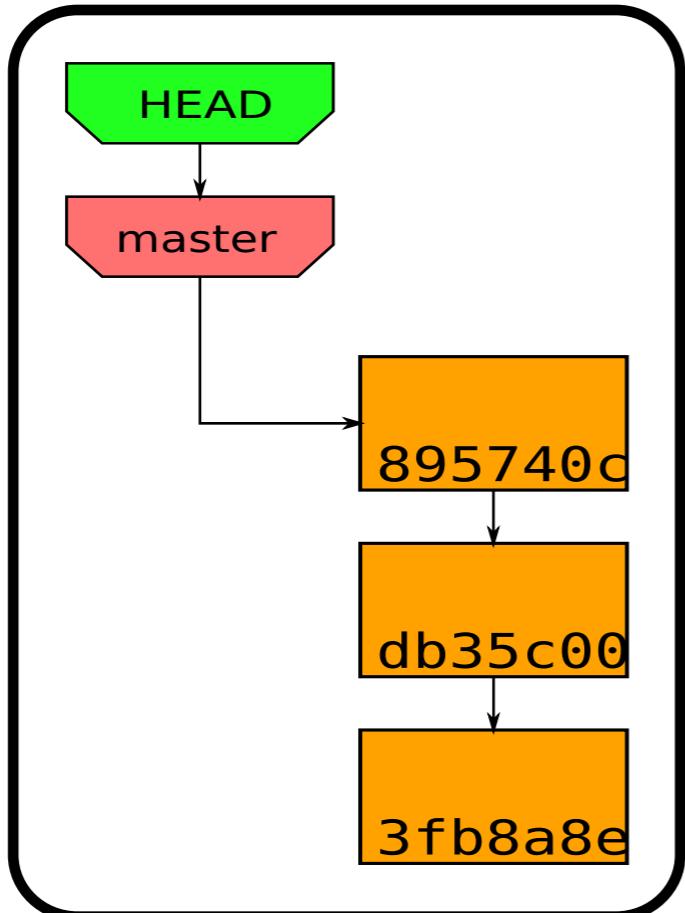
```
644 20b024 0 bar
644 1d52a6 0 baz
644 20b024 0 sub/fi
644 43dbe0 0 sub/foo
```

## Árbol de trabajo



## el Índice

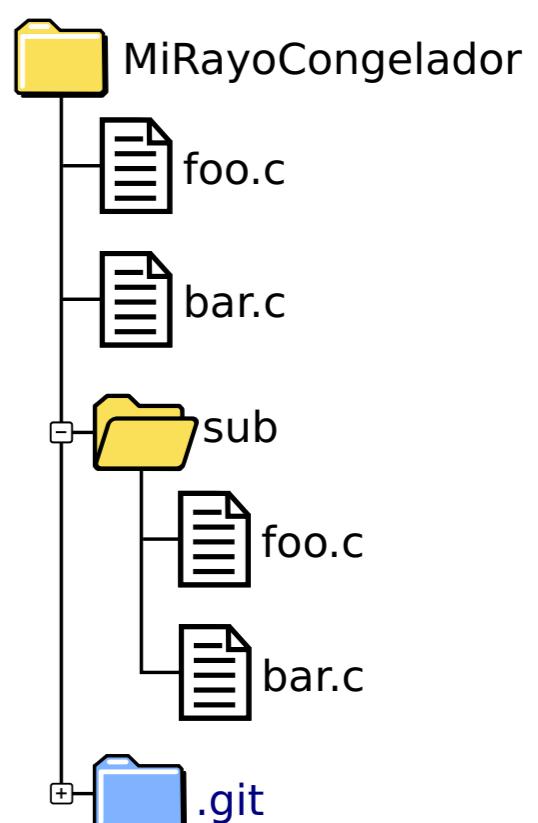
# Repositorio



# Área de preparación

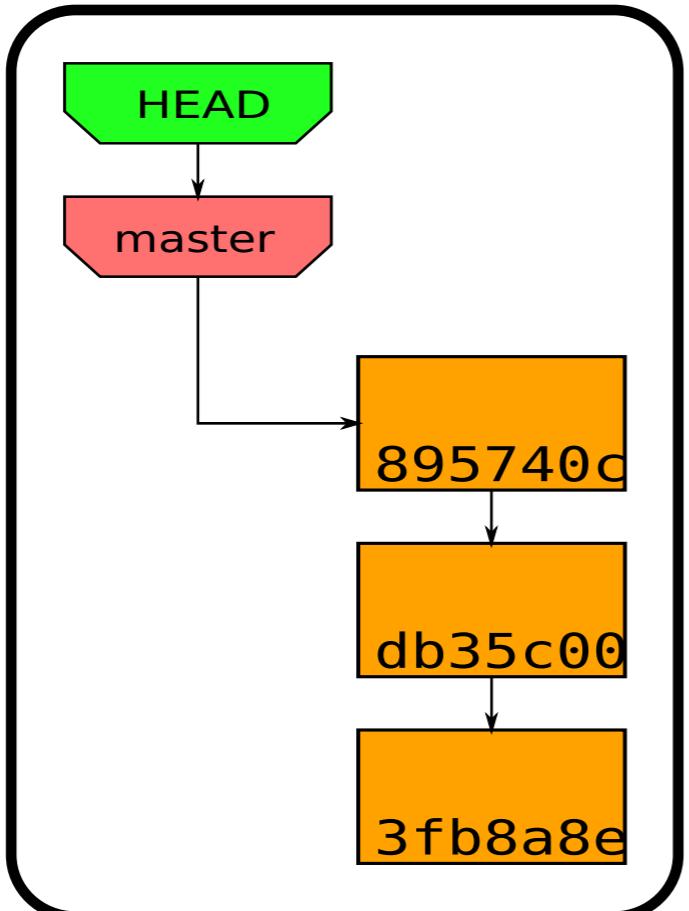
```
644 20b024 0 bar  
644 1d52a6 0 baz  
644 20b024 0 sub/fi  
644 43dbe0 0 sub/foo
```

# Árbol de trabajo



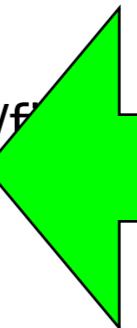
contiene la próxima revisión

# Repositorio

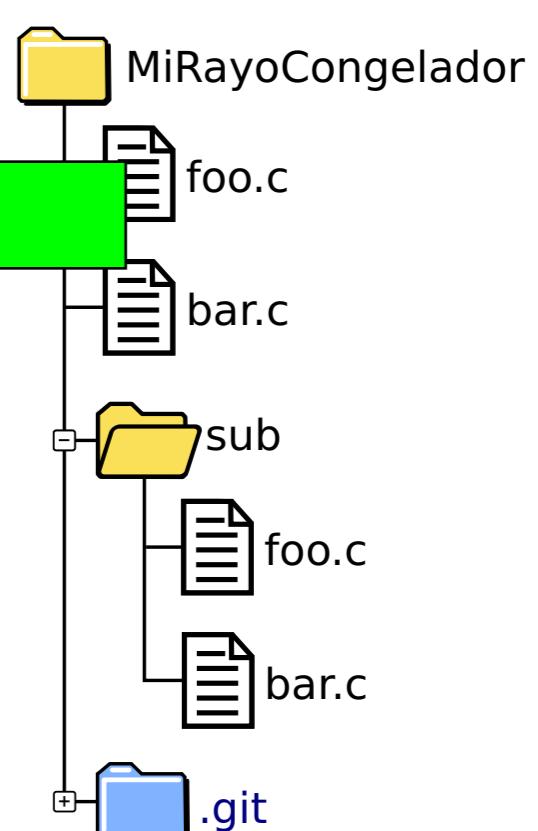


# Área de preparación

```
644 20b024 0 bar  
644 1d52a6 0 baz  
644 20b024 0 sub/f  
644 43dbe0 0 sub/
```



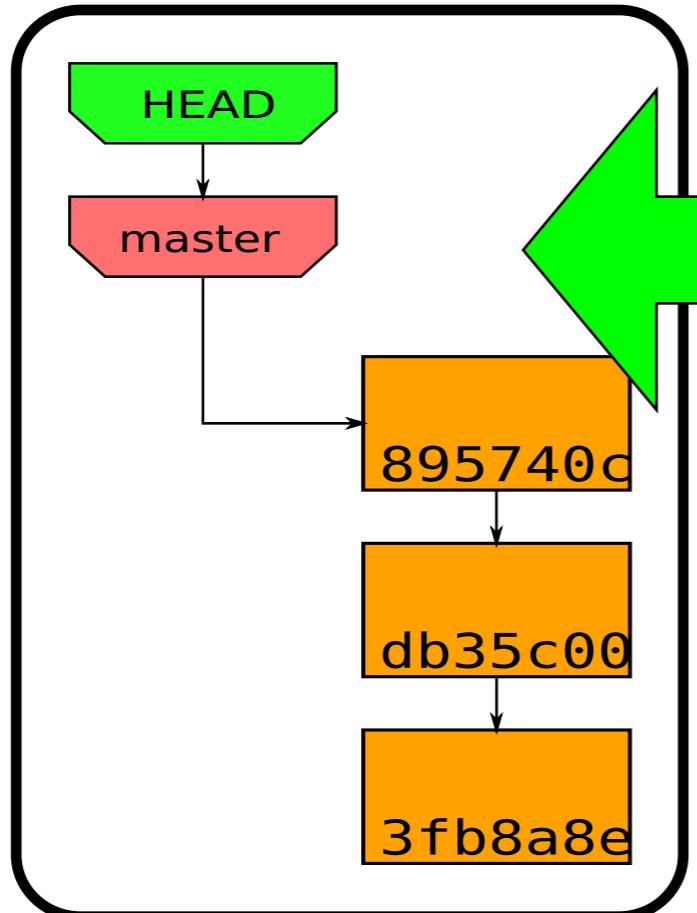
# Árbol de trabajo



"preparación"  
**git add**  
**git rm**  
**git mv**

# Intro Preparación

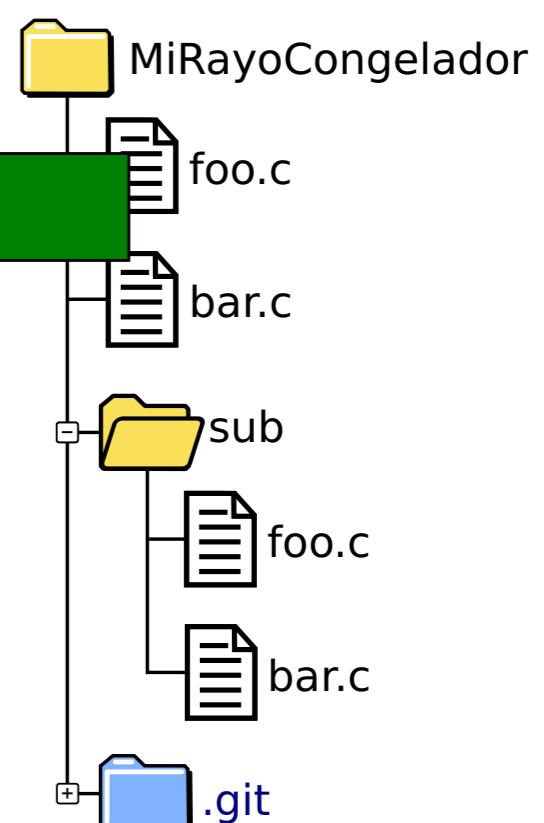
## Repositorio



## Área de preparación

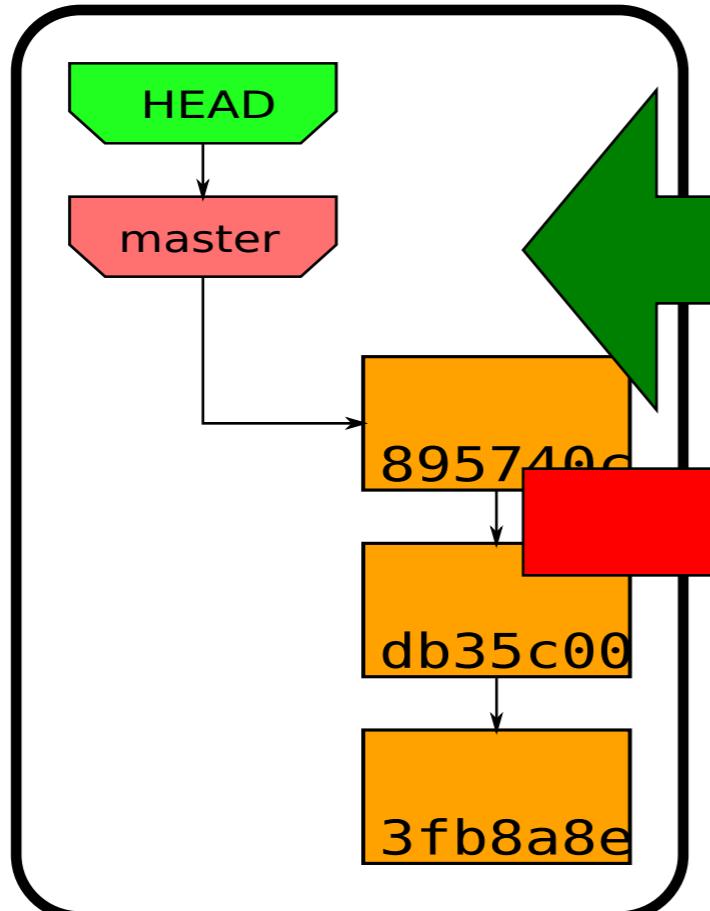
```
644 20b024 0 bar  
644 1d52a6 0 baz  
644 20b024 0 sub/f  
43dbe0 0 sub/
```

## Árbol de trabajo



"envío"  
**git commit**

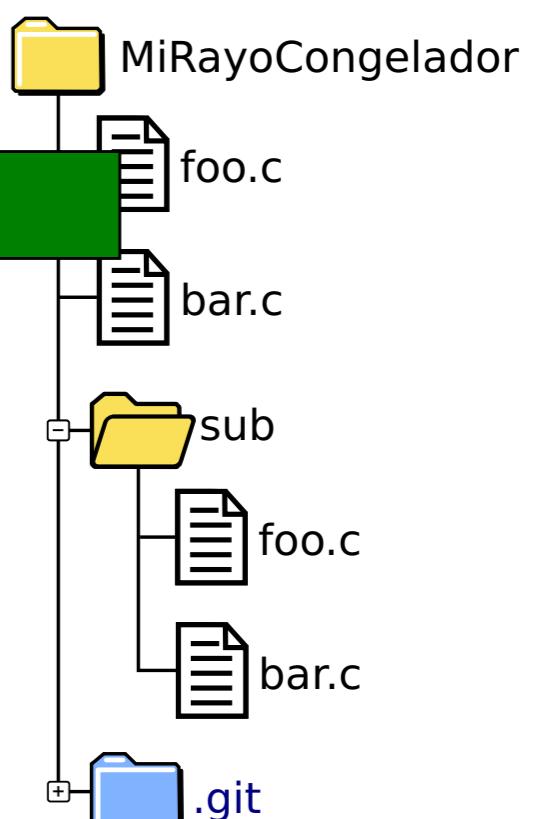
## Repositorio



## Área de preparación

```
644 20b024 0 bar  
644 1d52a6 0 baz  
644 20b024 0 sub/f  
43dbe0 0 sub/
```

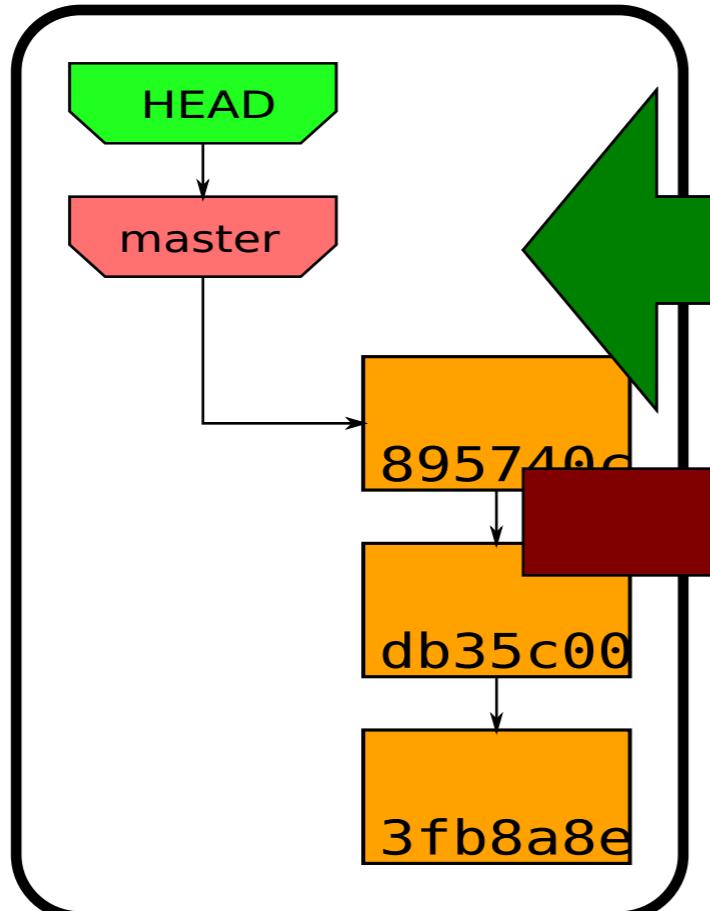
## Árbol de trabajo



"lectura de un árbol"

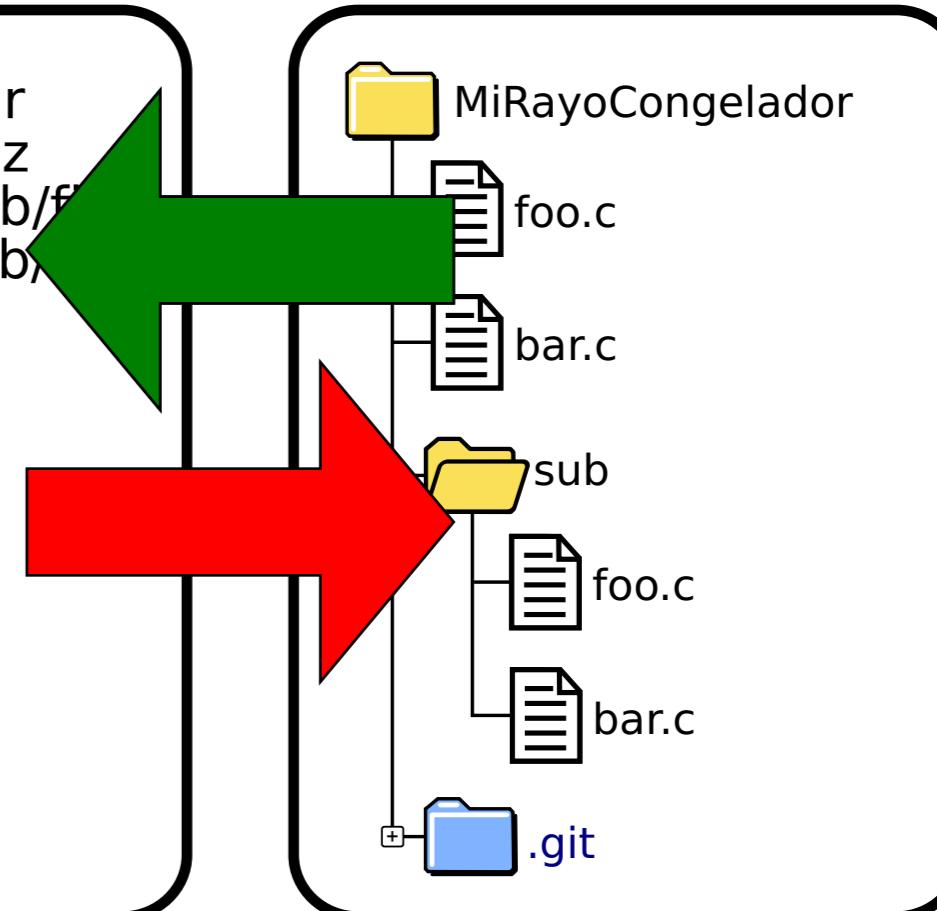
git read-tree  
git reset

## Repositorio



## Área de preparación

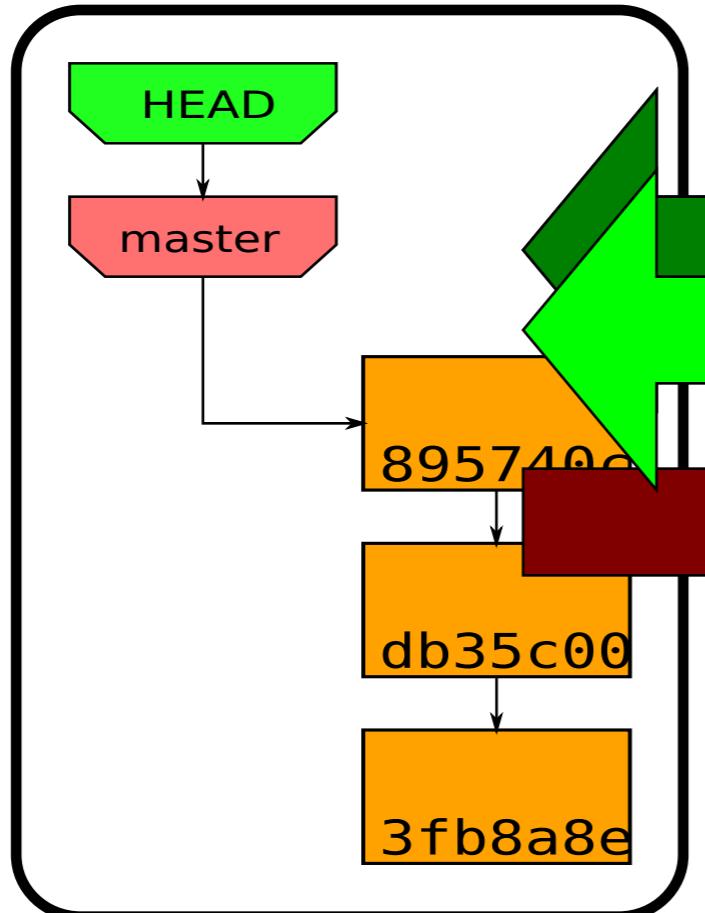
```
644 20b024 0 bar  
644 1d52a6 0 baz  
644 20b024 0 sub/f  
43dbe0 0 sub/
```



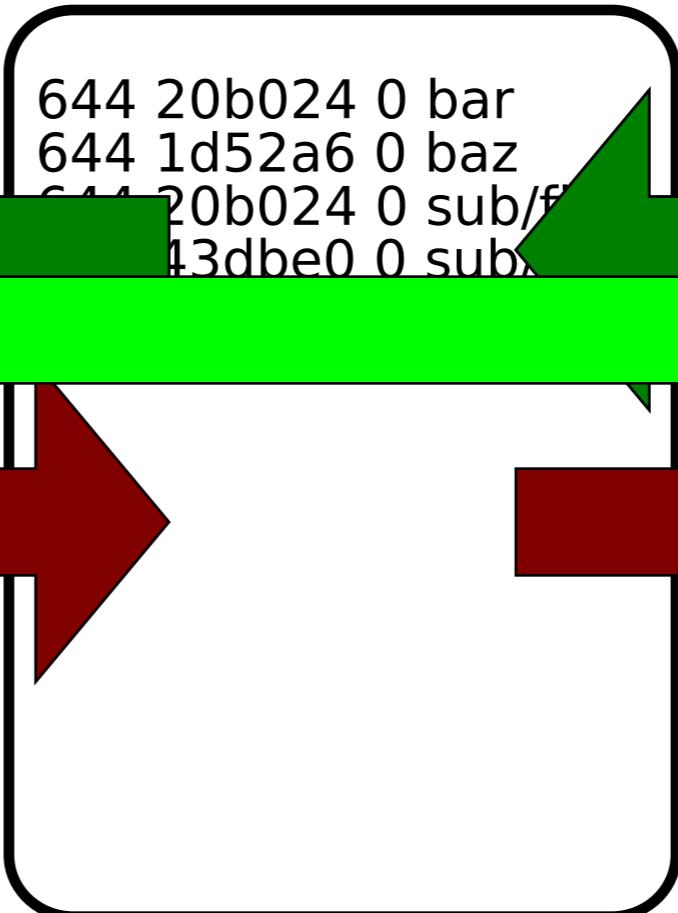
## Árbol de trabajo

"cargar un fichero"  
`git checkout -- <fichero>`

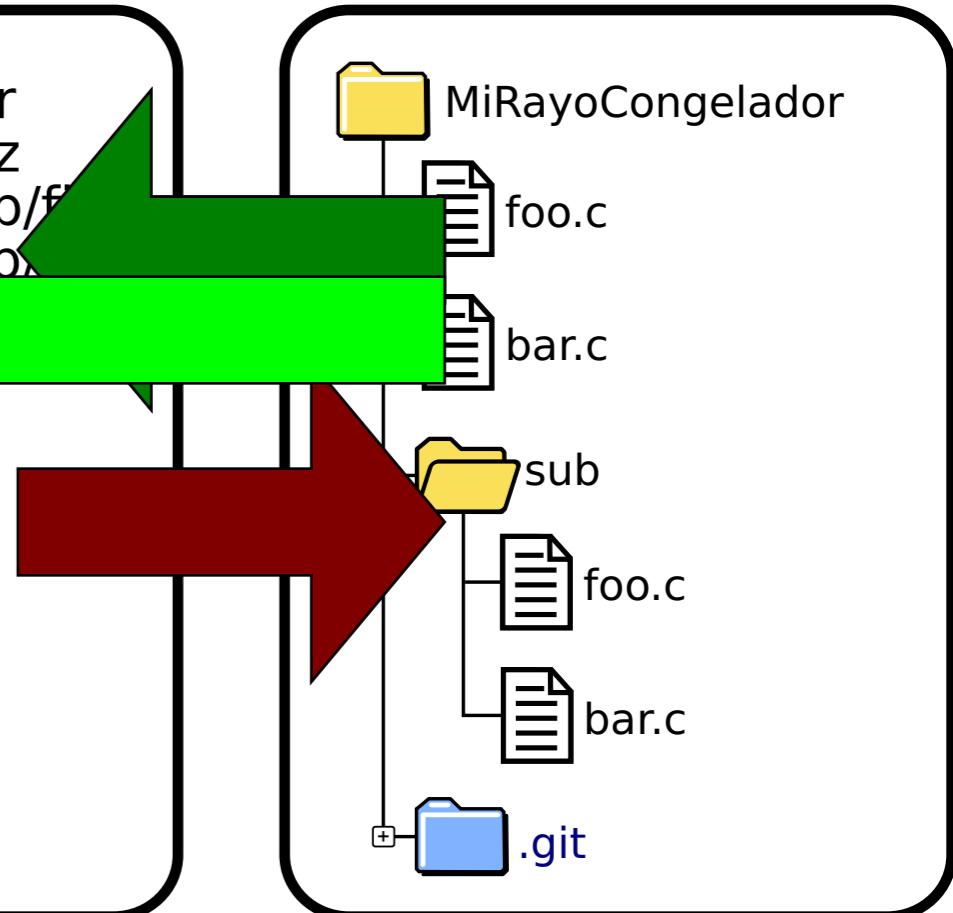
# Repositorio



# Área de preparación

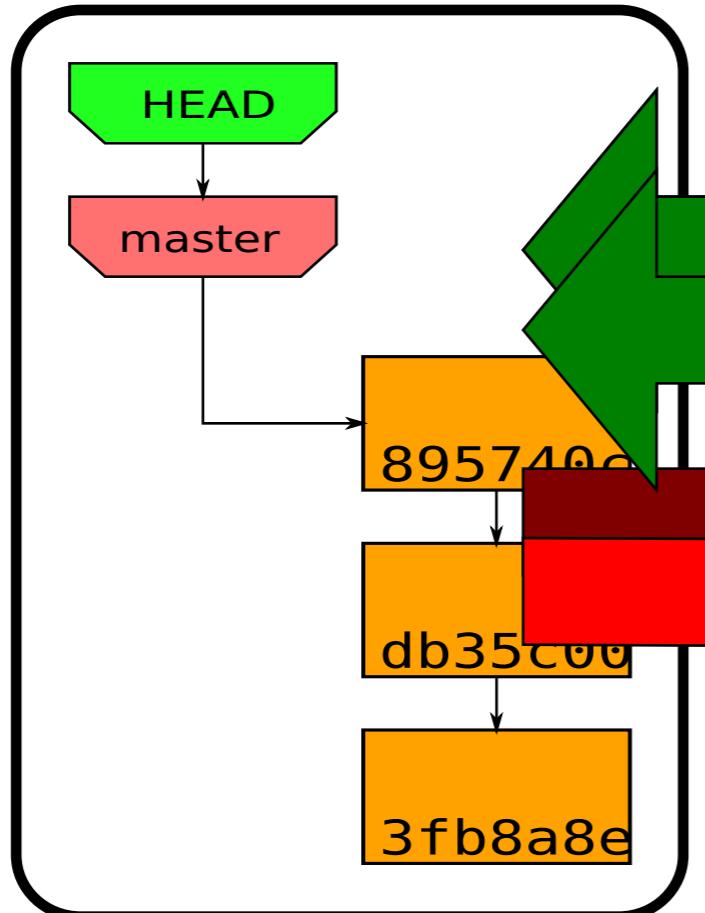


# Árbol de trabajo

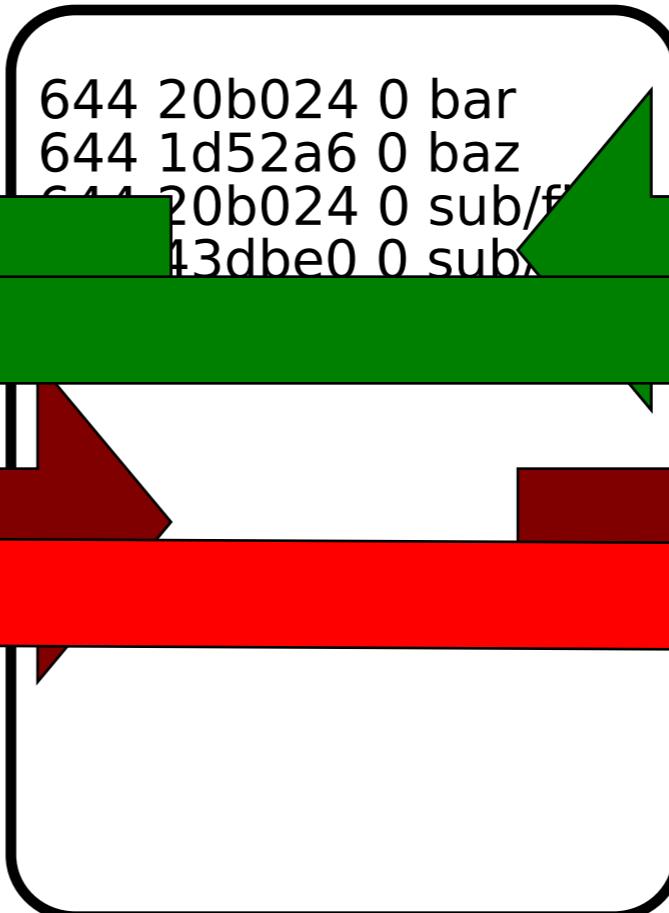


prepara y envía  
`git commit -a`  
`git commit <fichero>`

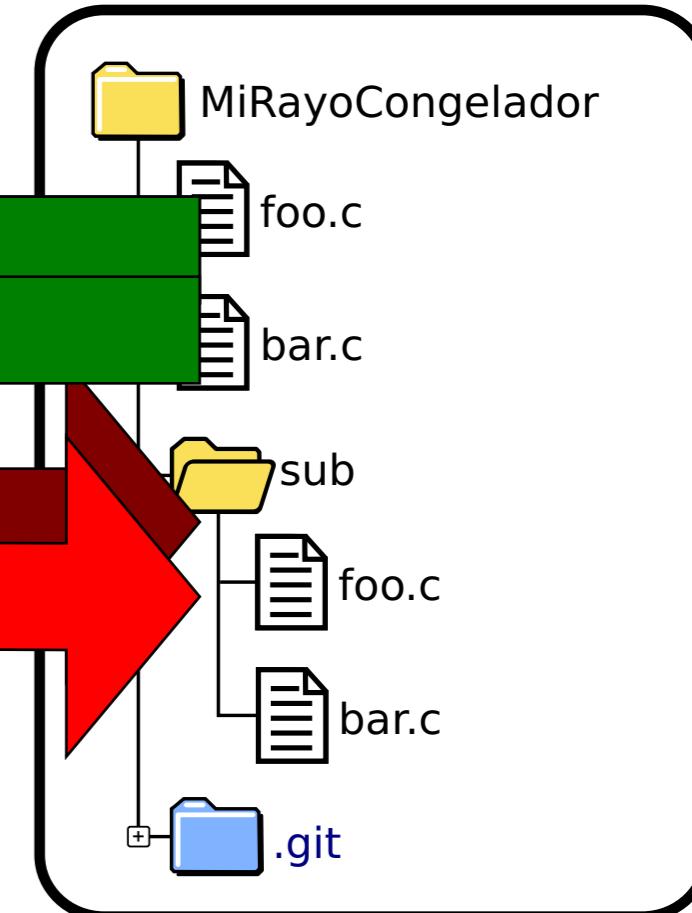
# Repositorio



# Área de preparación



# Árbol de trabajo



cambio de revisión  
`git checkout <revisión>`

.git

.git	
-- HEAD	referencia revisión actual
-- config	configuración local al repo
-- description	descripción para gitweb
-- hooks	ganchos para eventos
`-- ...	
-- index	área de preparación
-- info	
-- exclude	patrones locales a ignorar
`-- refs	referencias
-- logs	
`-- ...	
-- objects	datos del reflog
-- XX	objetos sueltos
`-- ...	
`-- info	
`-- packs	lista ficheros pack
`-- pack	índices y ficheros pack
`-- ...	
-- packed-refs	índice de todas las refs
-- refs	
-- heads	
`-- master	rama principal
`-- remotes	ramas remotas
`-- ...	
`-- stash	pseudo-rama de alijo
`-- tags	referencias de etiquetas

.git	
-- HEAD	referencia revisión actual
-- config	configuración local al repo
-- description	descripción para gitweb
-- hooks	ganchos para eventos
-- index	área de preparación
-- info	
\-- exclude	patrones locales a ignorar
\-- refs	referencias
-- logs	
-- objects	datos del reflog
\-- XX	objetos sueltos
\-- info	
\-- packs	lista ficheros pack
\-- pack	índices y ficheros pack
-- packed-refs	
-- refs	índice de todas las refs
\-- heads	
\-- master	rama principal
\-- remotes	ramas remotas
\-- stash	
\-- tags	pseudo-rama de alijo
	referencias de etiquetas

.git	
-- HEAD	referencia revisión actual
-- config	configuración local al repo
-- description	descripción para gitweb
-- hooks	ganchos para eventos
`-- ...	
-- index	área de preparación
-- info	
-- exclude	patrones locales a ignorar
`-- refs	referencias
-- logs	
`-- ...	
-- objects	datos del reflog
-- XX	objetos sueltos
`-- ...	
`-- info	
`-- packs	lista ficheros pack
`-- pack	índices y ficheros pack
`-- ...	
-- packed-refs	índice de todas las refs
-- refs	
-- heads	
`-- master	rama principal
`-- remotes	ramas remotas
`-- ...	
`-- stash	pseudo-rama de alijo
`-- tags	referencias de etiquetas

.git	
-- HEAD	referencia revisión actual
-- config	configuración local al repo
-- description	descripción para gitweb
-- hooks	ganchos para eventos
`-- ...	
-- index	área de preparación
-- info	
-- exclude	patrones locales a ignorar
`-- refs	referencias
-- logs	
`-- ...	
-- objects	datos del reflog
-- XX	objetos sueltos
`-- ...	
`-- info	
`-- packs	lista ficheros pack
`-- pack	índices y ficheros pack
`-- ...	
-- packed-refs	índice de todas las refs
-- refs	
-- heads	
`-- master	rama principal
`-- remotes	ramas remotas
`-- ...	
`-- stash	pseudo-rama de alijo
`-- tags	referencias de etiquetas

.git	
-- HEAD	referencia revisión actual
-- config	configuración local al repo
-- description	descripción para gitweb
-- hooks	ganchos para eventos
`-- ...	
-- index	área de preparación
-- info	
-- exclude	patrones locales a ignorar
`-- refs	referencias
-- logs	
`-- ...	
-- objects	datos del reflog
-- XX	objetos sueltos
`-- ...	
`-- info	
`-- packs	lista ficheros pack
`-- pack	índices y ficheros pack
`-- ...	
-- packed-refs	índice de todas las refs
-- refs	
-- heads	
`-- master	rama principal
`-- remotes	ramas remotas
`-- ...	
`-- stash	pseudo-rama de alijo
`-- tags	referencias de etiquetas

# Objetos

# Intro Objetos

```
.git
-- HEAD
-- config
-- description
-- hooks
`-- ...
-- index
-- info
`-- exclude
`-- refs
-- logs
`-- ...
-- objects
`-- XX
`-- ...
`-- info
`-- packs
`-- pack
`-- ...
-- packed-refs
-- refs
`-- heads
`-- master
`-- remotes
`-- ...
`-- stash
`-- tags
```

los objetos  
están aquí!

```
.git/objects
-- 21
`-- a2ad62733032aa190a4e52b4c517bdc46839ea
-- 3e
`-- 1f5aca211ecf48c9977d1db3ab3e14ebeca66e
-- 73
`-- ffb862b1a4ca28f05456e61190e78ca6eef9e0
-- d0
`-- fefa3d5eb85980970b1e856ff1d20c15f269cf
-- info
`-- packs
-- pack
`-- pack-03c4998d126189642f19d0dd2ff0fce1.idx
`-- pack-03c4998d126189642f19d0dd2ff0fce1.pack
```

```
.git/objects
-- 21
`-- a2ad62733032aa190a4e52b4c517bdc46839ea
-- 3e
`-- 1f5aca211ecf48c9977d1db3ab3e14ebeca66e
-- 73
`-- ffb862b1a4ca28f05456e61190e78ca6eef9e0
-- d0
`-- fefa3d5eb85980970b1e856ff1d20c15f269cf
-- info
`-- packs
-- pack
`-- pack-03c4998d126189642f19d0dd2ff0fce1.idx
`-- pack-03c4998d126189642f19d0dd2ff0fce1.pack
```

objetos sueltos

```
.git/objects
-- 21
`-- a2ad62733032aa190a4e52b4c517bdc46839ea
-- 3e
`-- 1f5aca211ecf48c9977d1db3ab3e14ebeca66e
-- 73
`-- ffb862b1a4ca28f05456e61190e78ca6eef9e0
-- d0
`-- fefa3d5eb85980970b1e856ff1d20c15f269cf
-- info
`-- packs
-- pack
`-- pack-03c4998d126189642f19d0dd2ff0fce1.idx
`-- pack-03c4998d126189642f19d0dd2ff0fce1.pack
```

ficheros pack

```
.git/objects
-- 21
`-- a2ad62733032aa190a4e52b4c517bdc46839ea
-- 3e
`-- 1f5aca211ecf48c9977d1db3ab3e14ebeca66e
-- 73
`-- ffb862b1a4ca28f05456e61190e78ca6eef9e0
-- d0
`-- fefa3d5eb85980970b1e856ff1d20c15f269cf
-- info
`-- packs
-- pack
`-- pack-03c4998d126189642f19d0dd2ff0fce1.idx
`-- pack-03c4998d126189642f19d0dd2ff0fce1.pack
```

hay que empaquetar y limpiar de vez en cuando

**git gc**

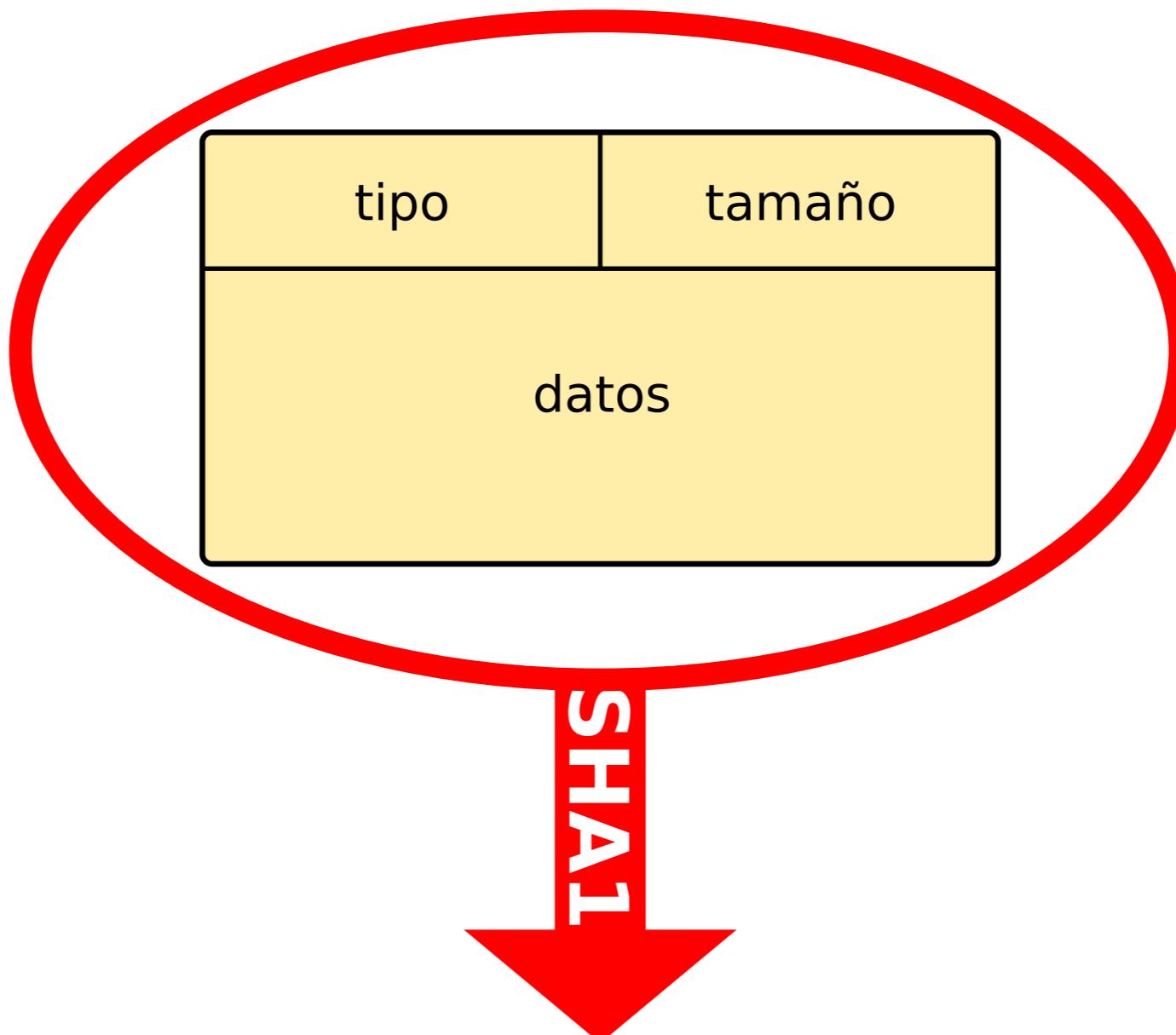
```
.git/objects
-- 21
`-- a2ad62733032aa190a4e52b4c517bdc46839ea
-- 3e
`-- 1f5aca211ecf48c9977d1db3ab3e14ebeca66e
-- 73
`-- ffb862b1a4ca28f05456e61190e78ca6eef9e0
-- d0
`-- fefa3d5eb85980970b1e856ff1d20c15f269cf
-- info
`-- packs
-- pack
`-- pack-03c4998d126189642f19d0dd2ff0fce1.idx
`-- pack-03c4998d126189642f19d0dd2ff0fce1.pack
```

para comprobar su consistencia

**git fsck**



# Direccional por contenido



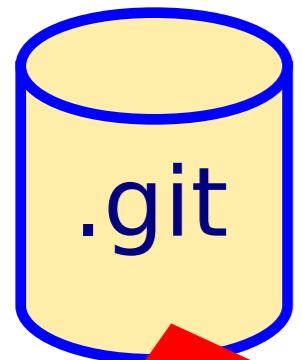
a2ad62733032aa190a4e52b4c517bdc46839ea  
← 20 bytes →

# Direccional por contenido



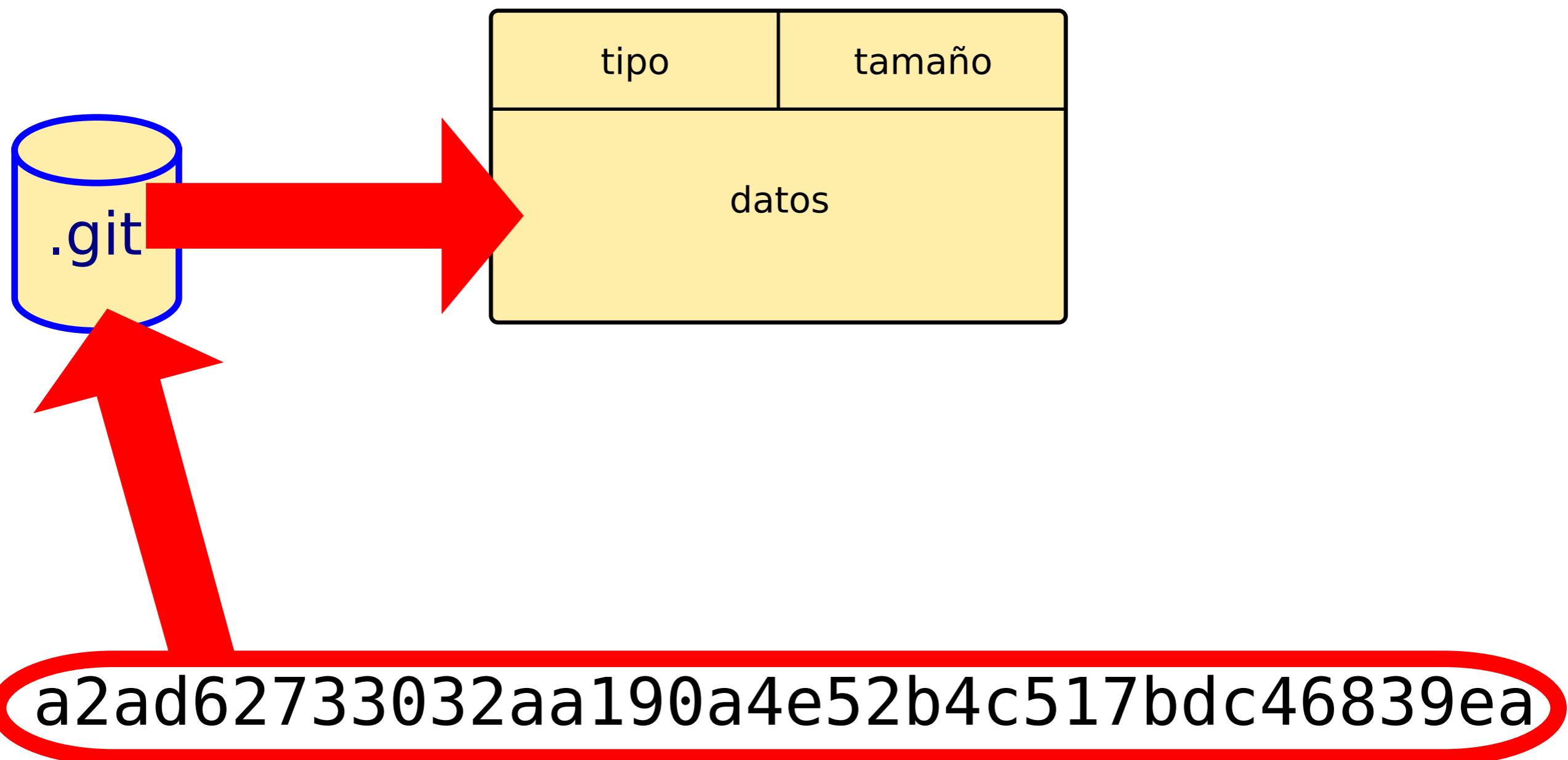
a2ad62733032aa190a4e52b4c517bdc46839ea

# Direccional por contenido

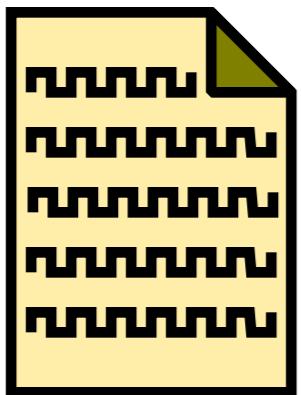


a2ad62733032aa190a4e52b4c517bdc46839ea

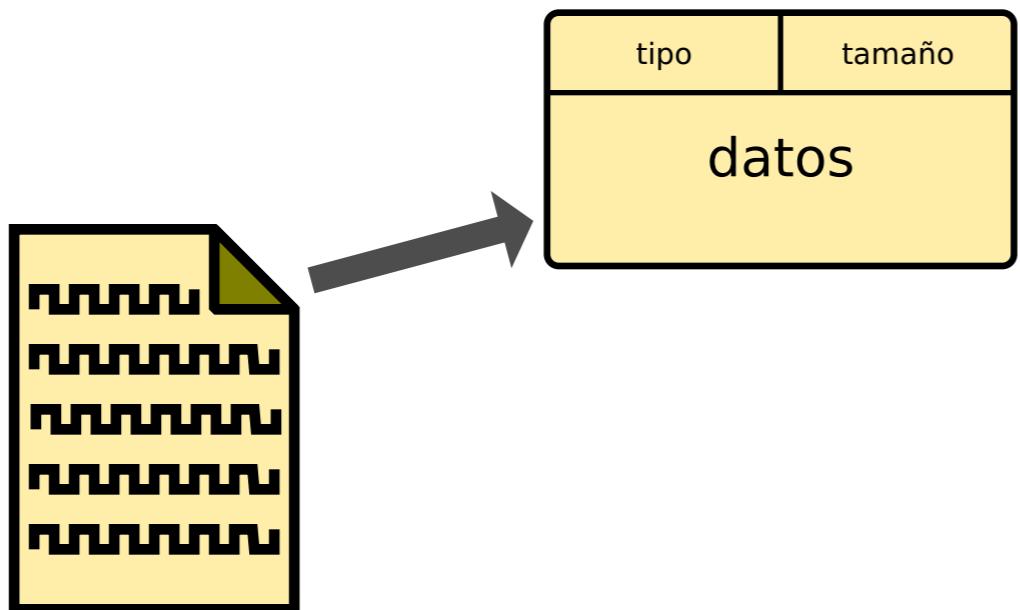
# Direccional por contenido



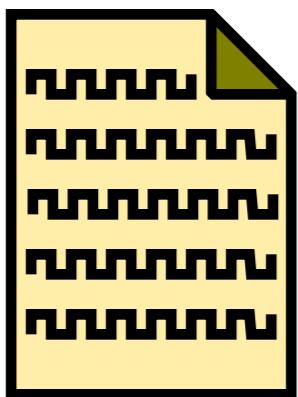
# Objetos inmutables



# Objetos inmutables



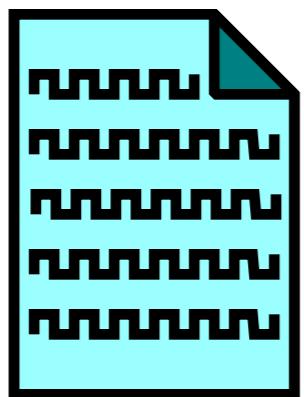
# Objetos inmutables



SHA1 →

4d45677c...

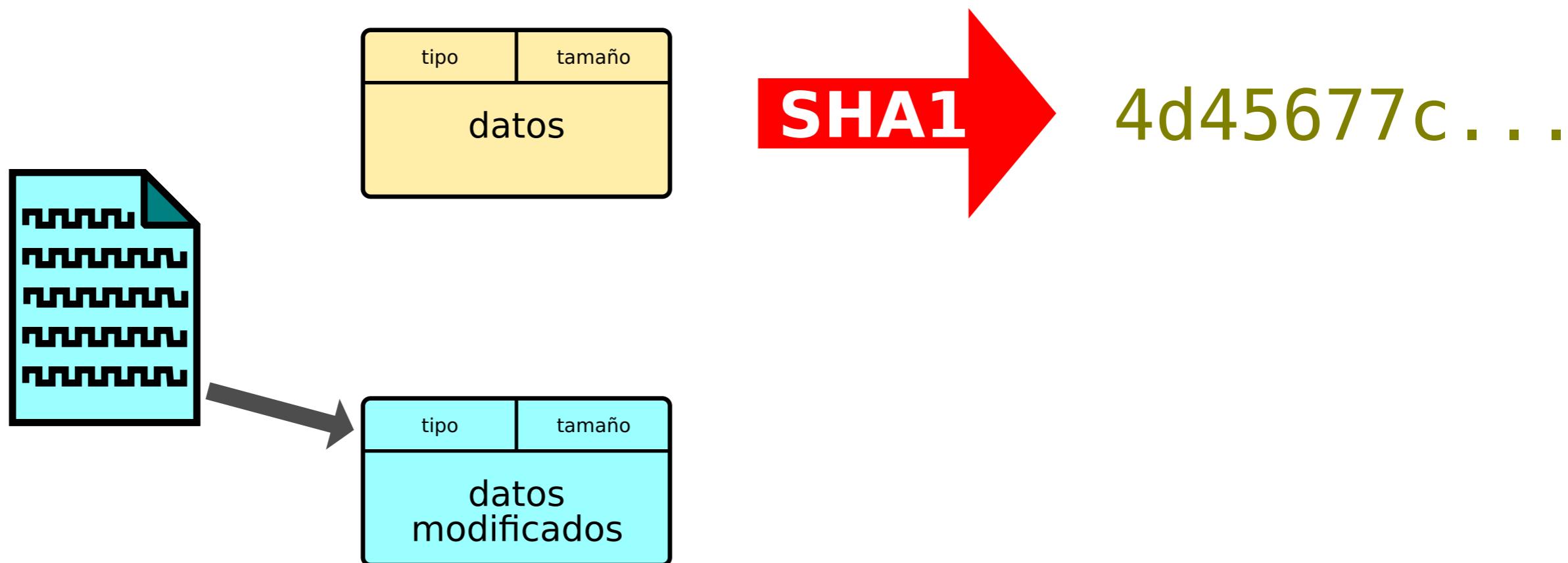
# Objetos inmutables



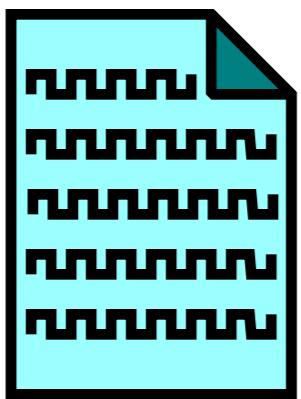
SHA1 →

4d45677c...

# Objetos inmutables



# Objetos inmutables



SHA1 →

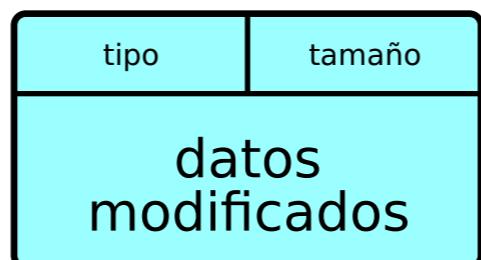
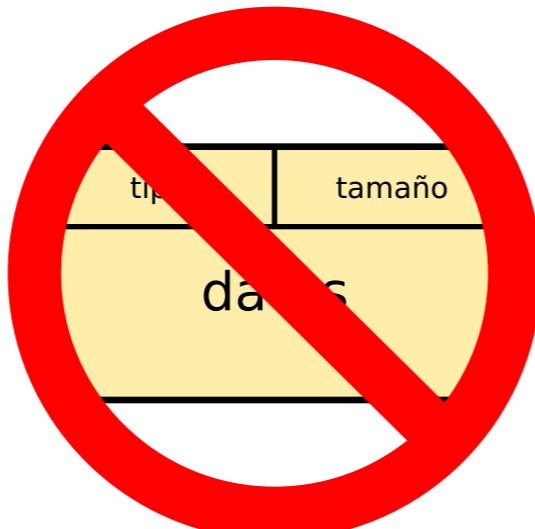
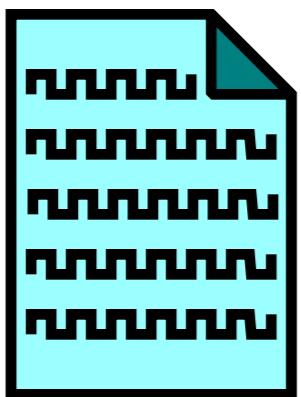
4d45677c...



SHA1 →

e4b898aa...

# Objetos inmutables



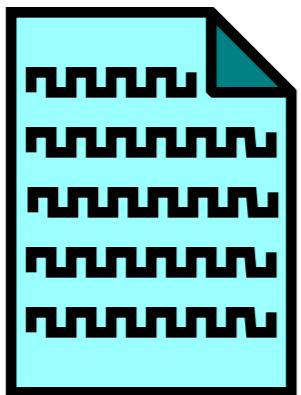
SHA1 →

4d45677c...

SHA1 →

e4b898aa...

# Objetos inmutables



tipo	tamaño
datos modificados	

SHA1 →

e4b898aa...

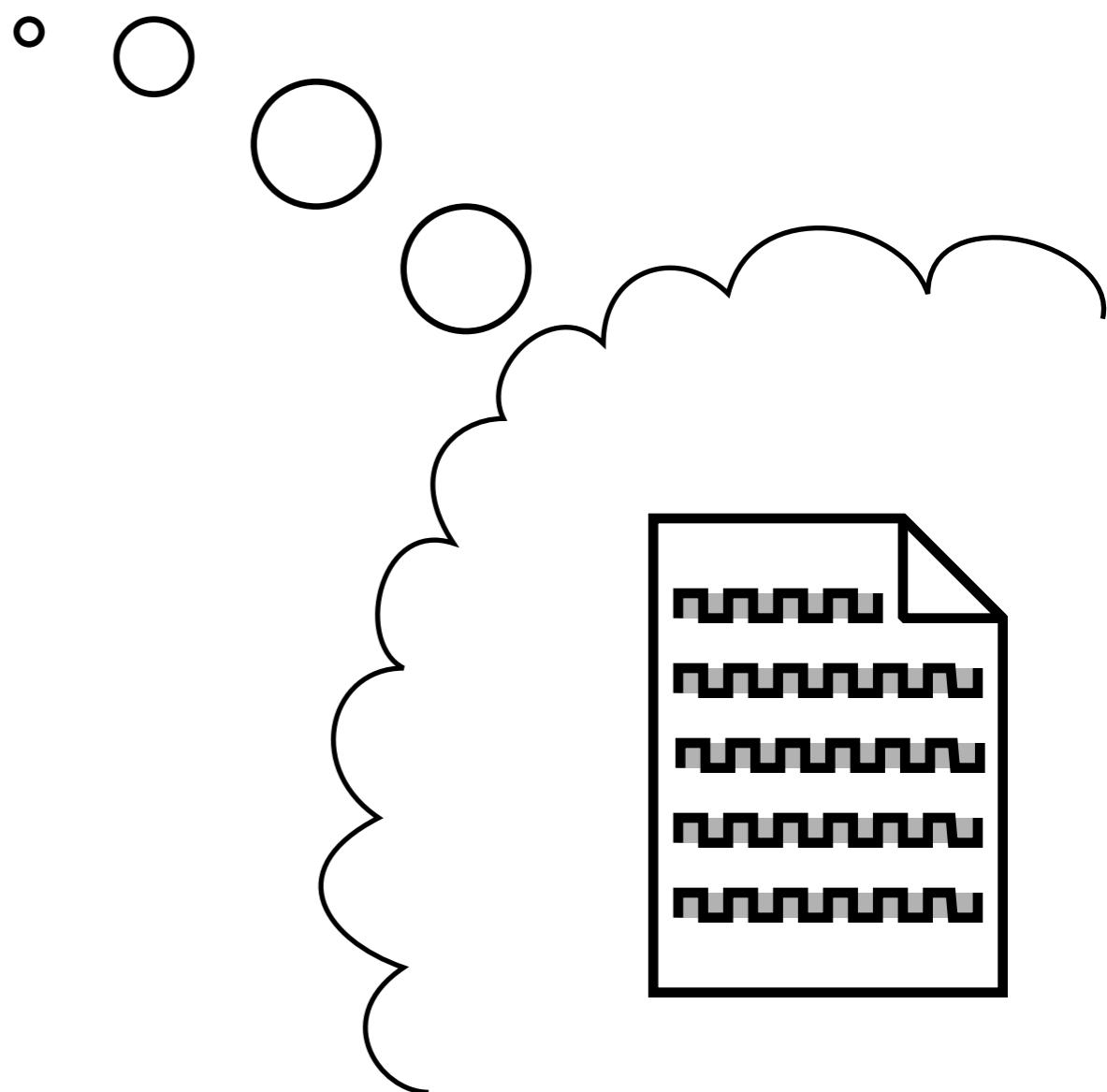
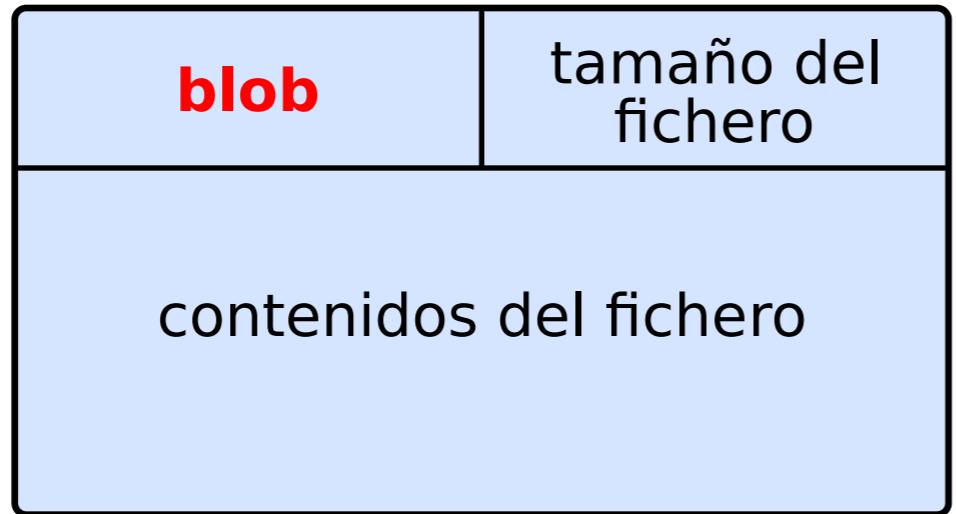
# Tipos

- bloque de bytes ("blob")
- árbol ("tree")
- revisión ("commit")
- etiqueta ("tag")

# Tipos

<b>blob</b>	tamaño del fichero
contenidos del fichero	

# Tipos

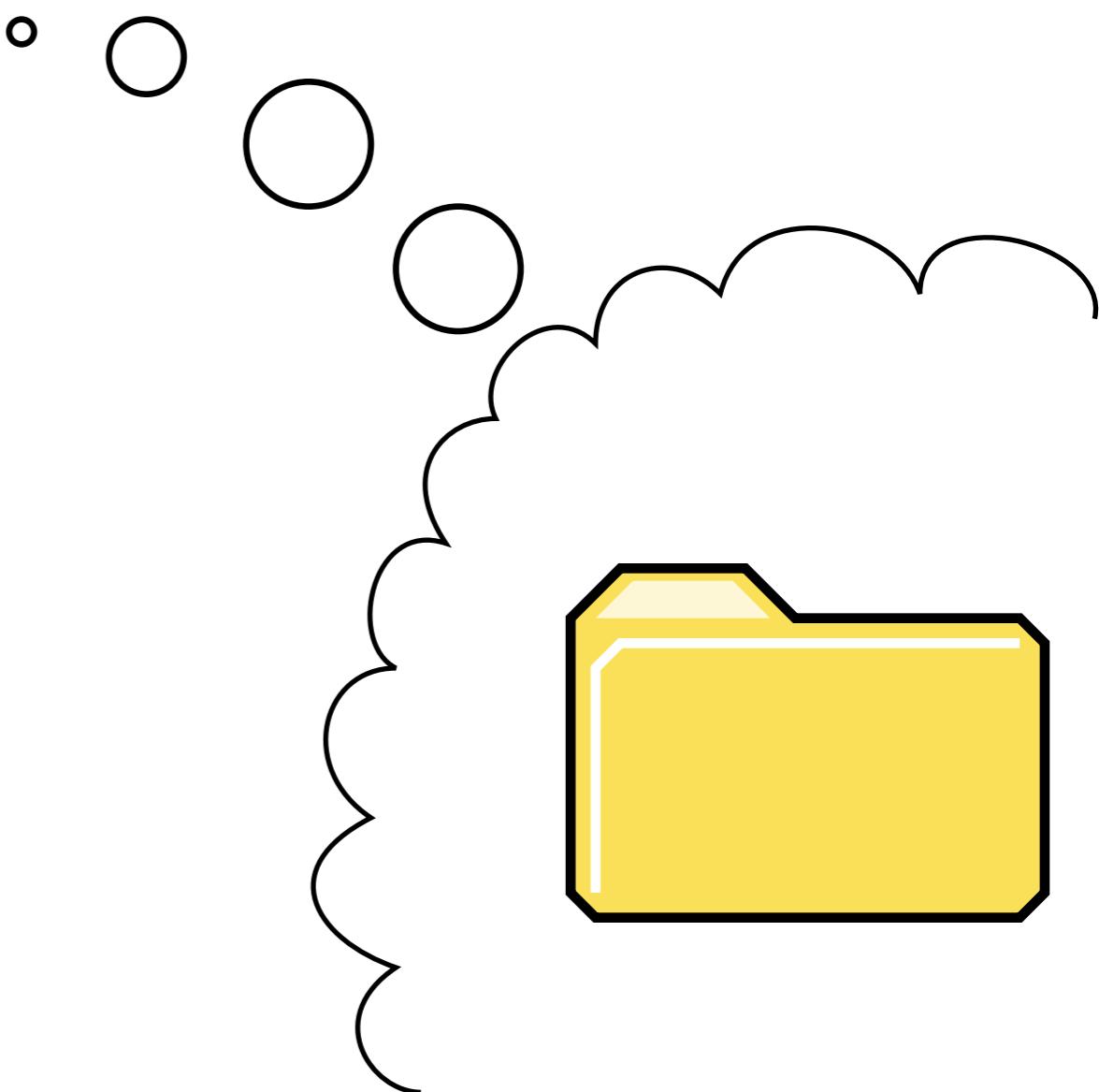


# Tipos

<b>tree</b>	tamaño del árbol
referencias a tree y/o blob	

# Tipos

tree	tamaño del árbol
referencias a tree y/o blob	



# Tipos

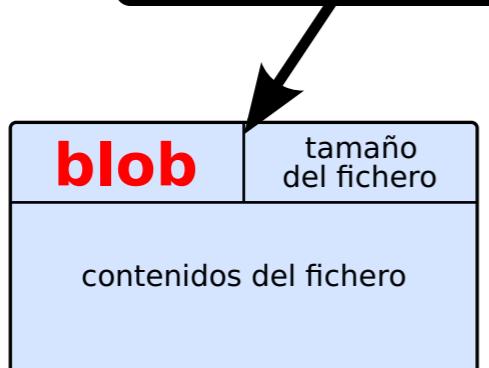
tree	tamaño del árbol
100644 blob 1f98957 foo.txt	
040000 tree 44f93d0 subdir	

# Tipos

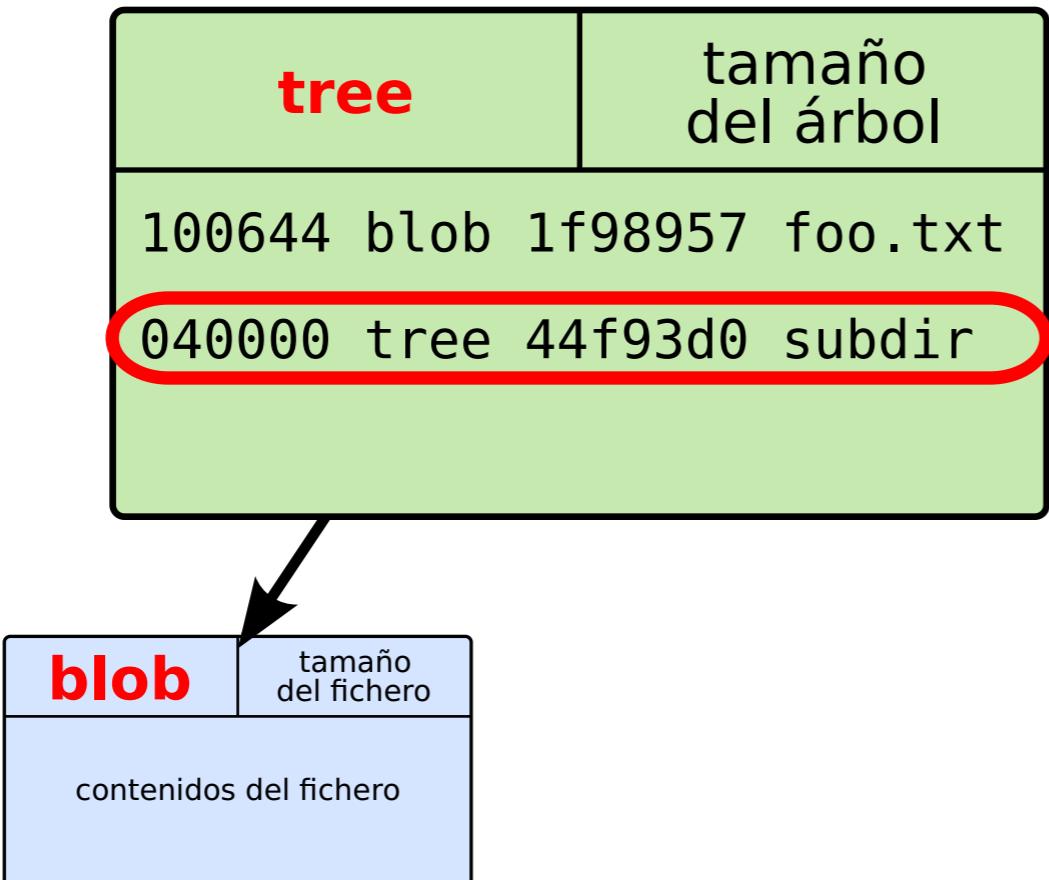
tree	tamaño del árbol
100644 blob 1f98957 foo.txt	
040000 tree 44f93d0 subdir	

# Tipos

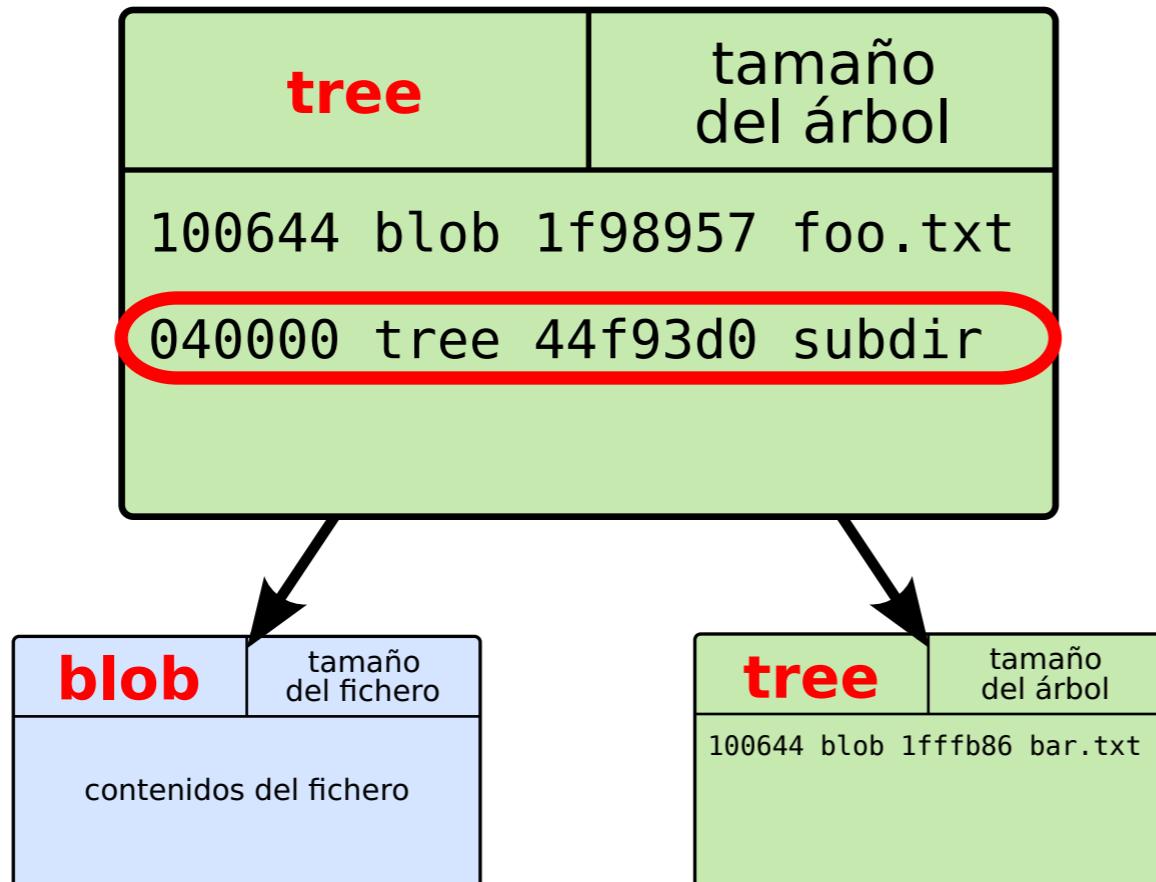
<b>tree</b>	tamaño del árbol
<b>100644 blob 1f98957 foo.txt</b>	
040000 tree 44f93d0 subdir	



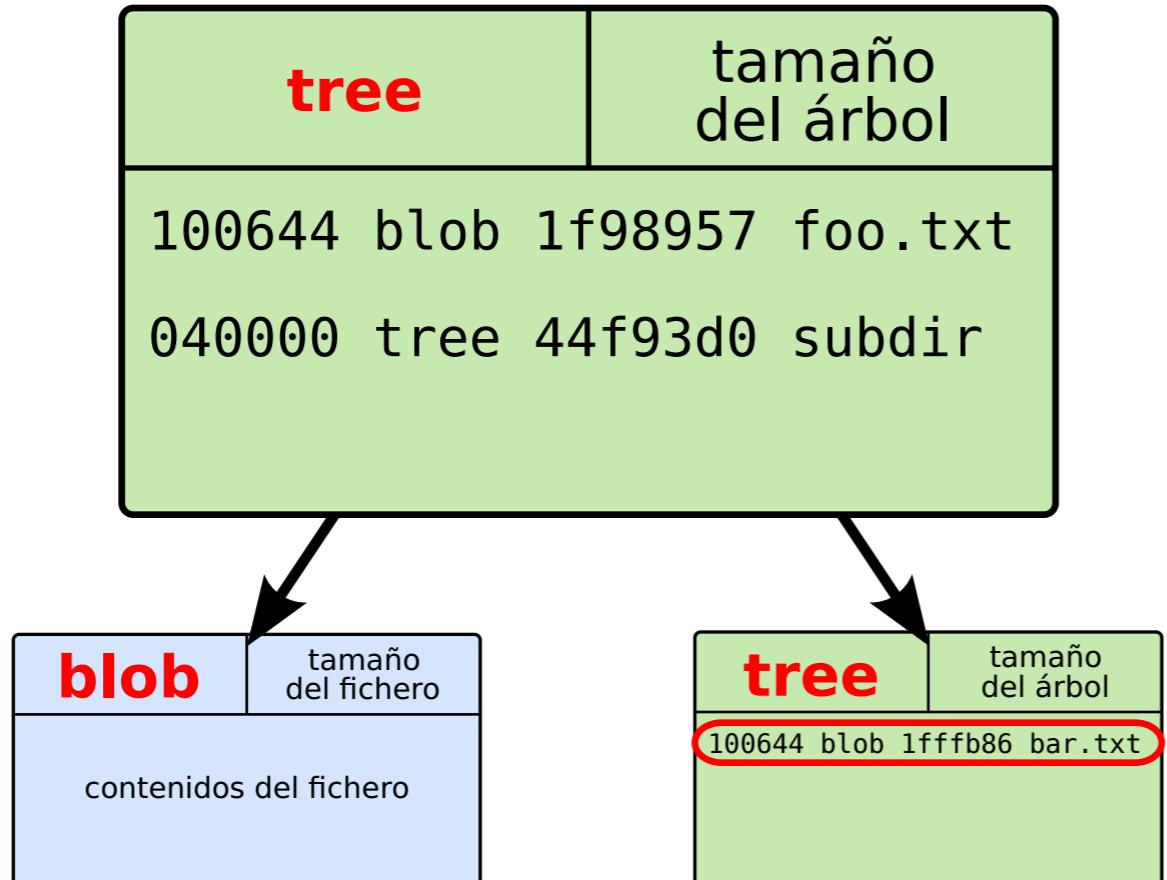
# Tipos



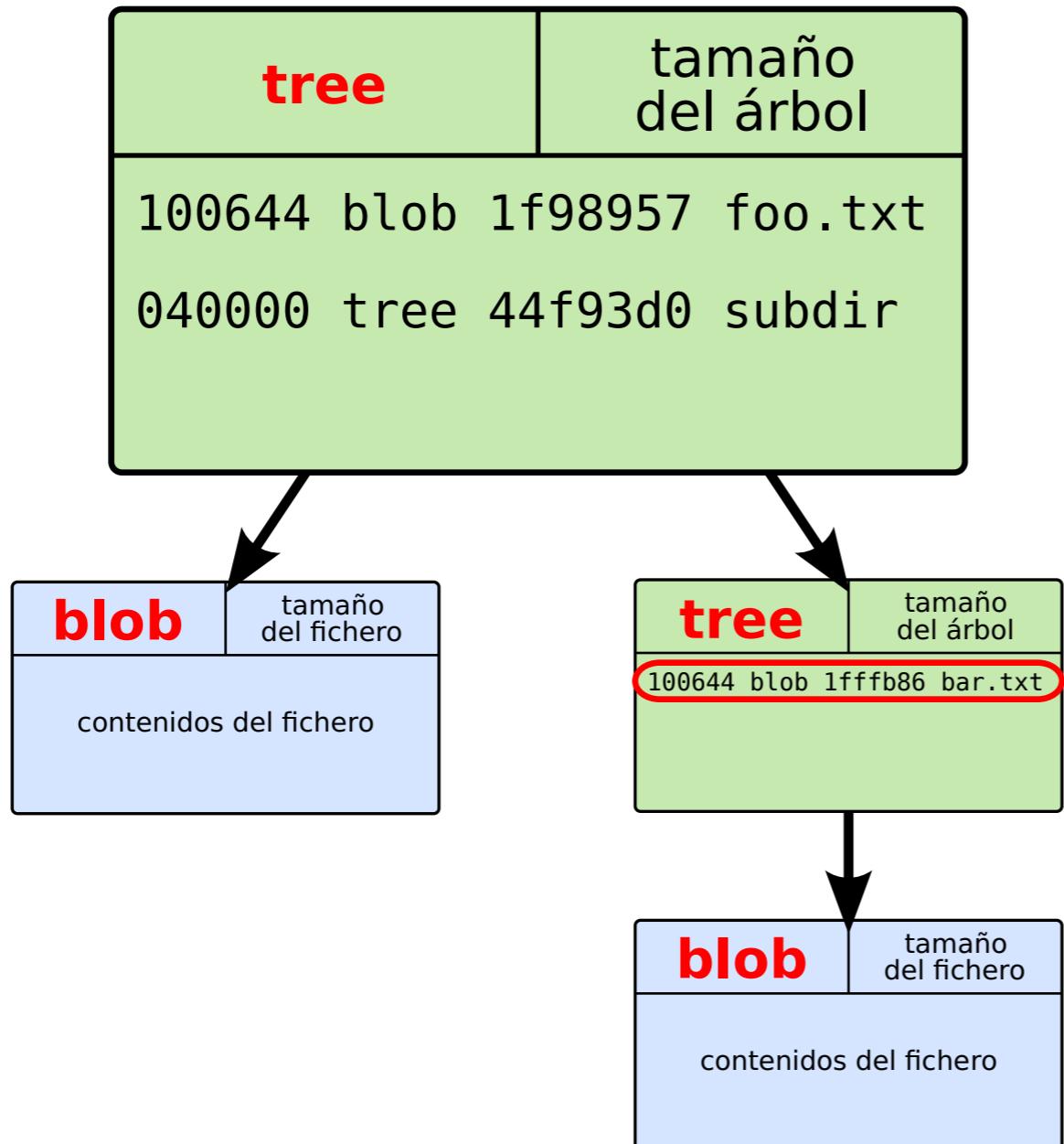
# Tipos



# Tipos



# Tipos



# Tipos

<b>commit</b>	tamaño de la revisión
metadatos de la revisión	

# Tipos

<b>commit</b>	tamaño de la revisión
metadatos de la revisión	

```
tree      7d34d18d4763f49930c25d5fcfb1d3f1c6eedbf8
parent    2df316d2558819a3305cf9ea3f0db6354bf2dc77
author    Bart Trojanowski <bart@jukie.net>
          Sun Sep 27 13:22:03 2009 -0400
committer Bart Trojanowski <bart@jukie.net>
          Sun Sep 27 13:22:03 2009 -0400
```

mensaje de la revisión

# Tipos

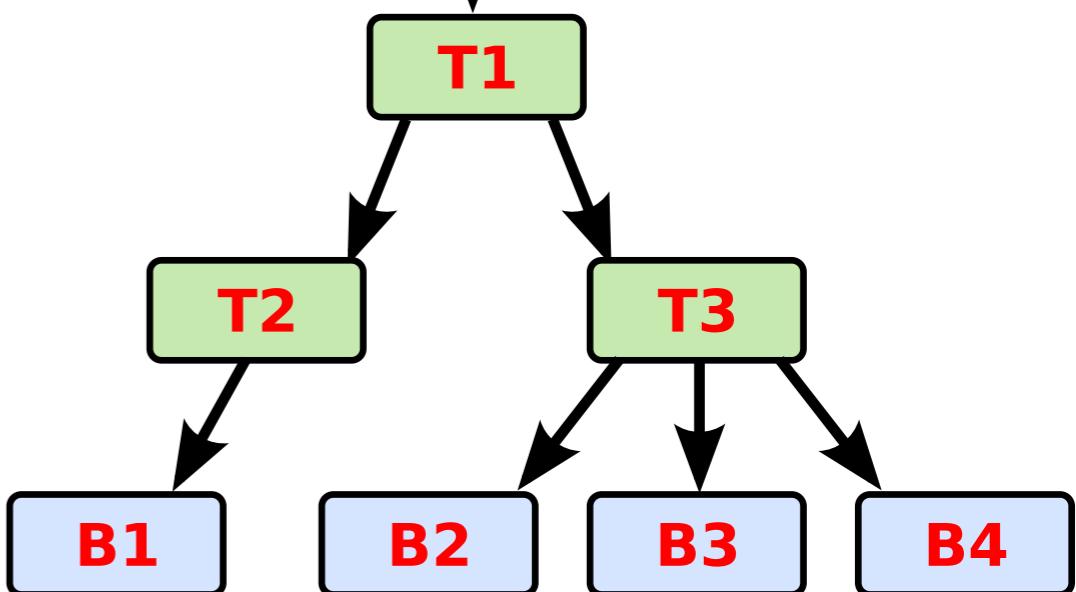
commit	tamaño de la revisión
tree	7d34d18d4763f49...
parent	2df316d2558819a...
author	Bart <bart@..> fecha
committer	Bart <bart@..> fecha
mensaje de la revisión	

# Tipos

commit	tamaño de la revisión
tree	7d34d18d4763f49...
parent	2d1316d2558819a...
author	Bart <bart@..> fecha
committer	Bart <bart@..> fecha
mensaje de la revisión	

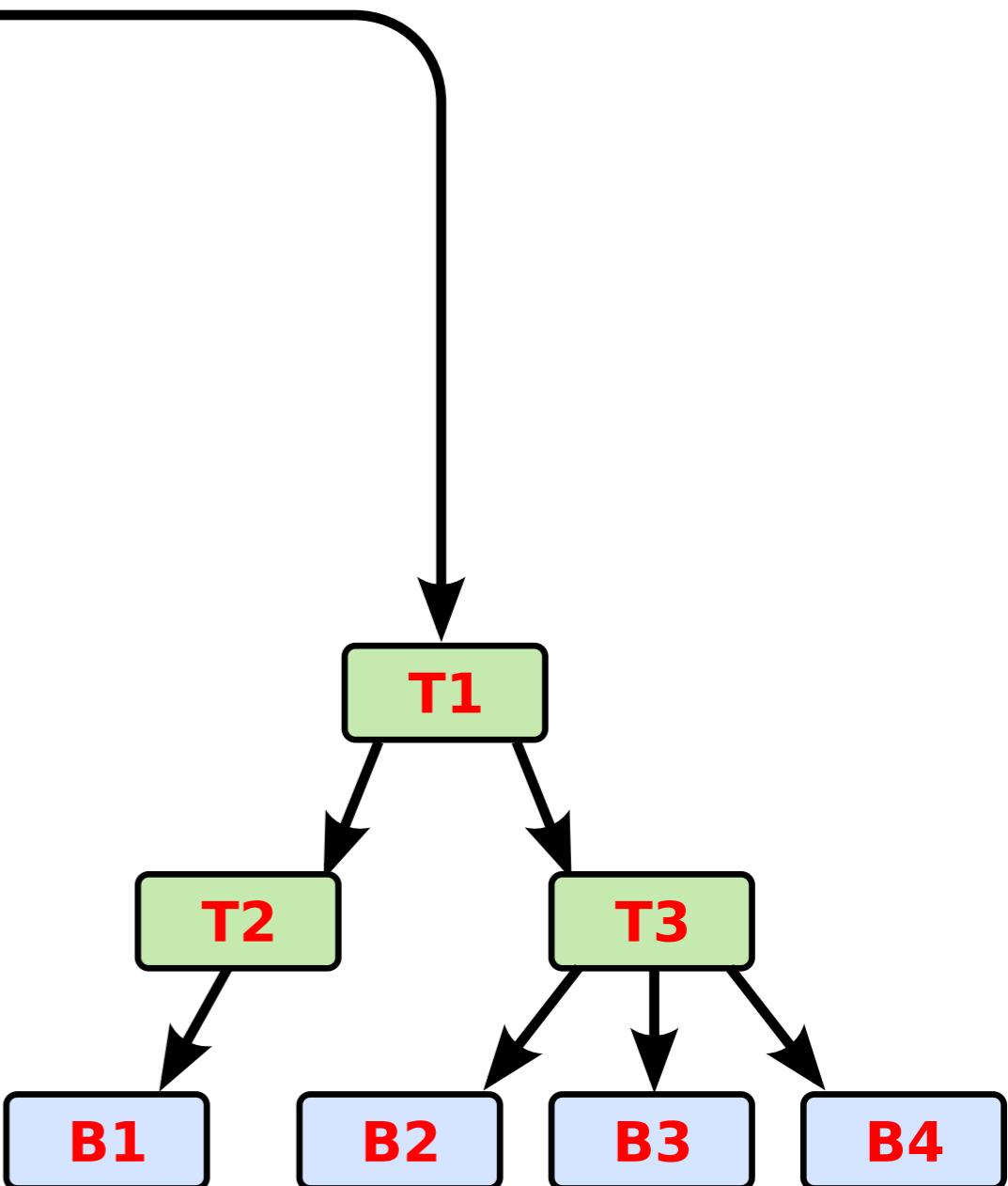
# Tipos

commit	tamaño de la revisión
tree	7d34d18d4763f49...
parent	2d1316d2558819a...
author	Bart <bart@..> fecha
committer	Bart <bart@..> fecha
mensaje de la revisión	



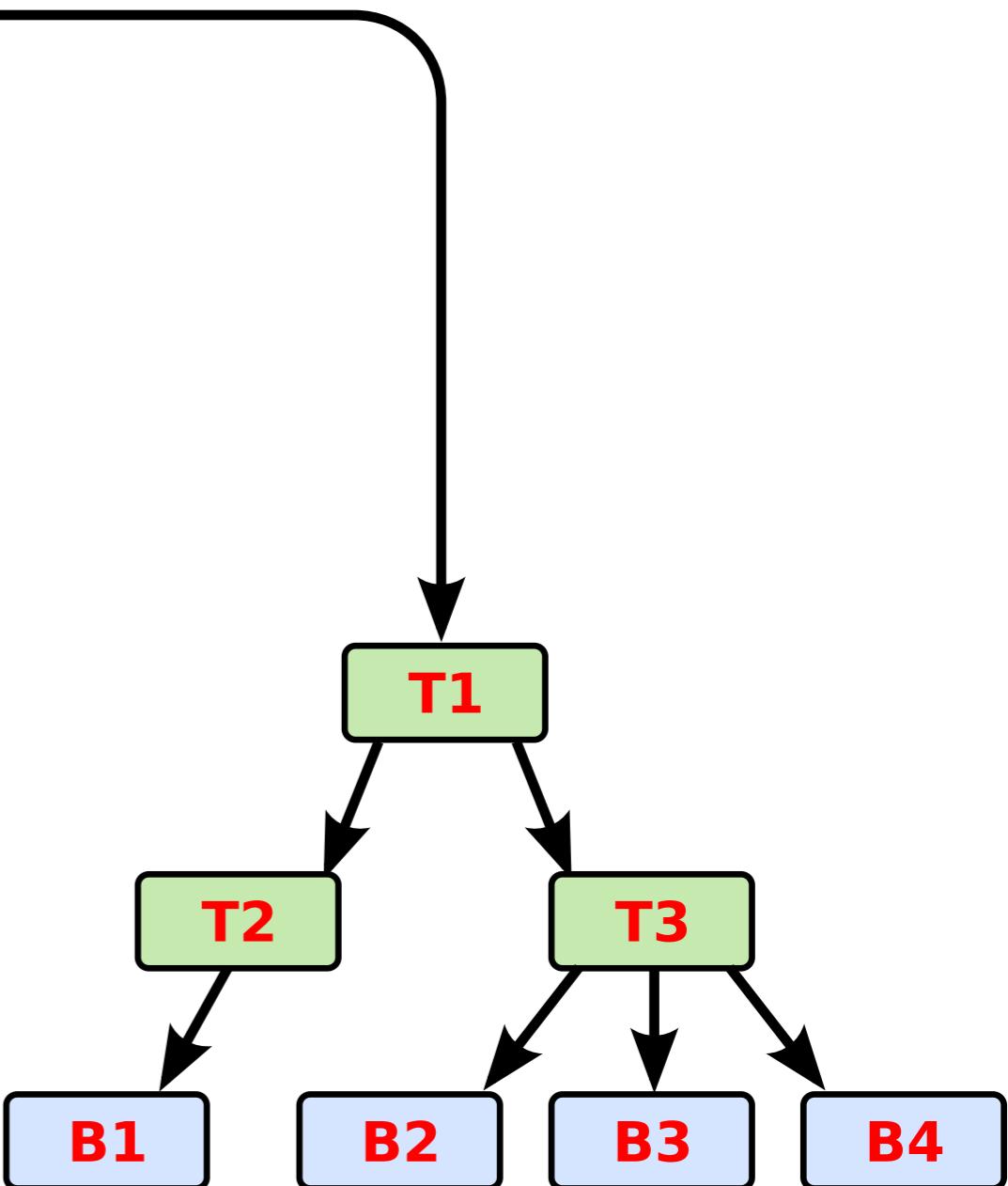
# Tipos

commit	tamaño de la revisión
tree	7d34d18d4763f49...
parent	2df316d2558819a
author	Bart <bart@...> fecha
committer	Bart <bart@...> fecha
mensaje de la revisión	



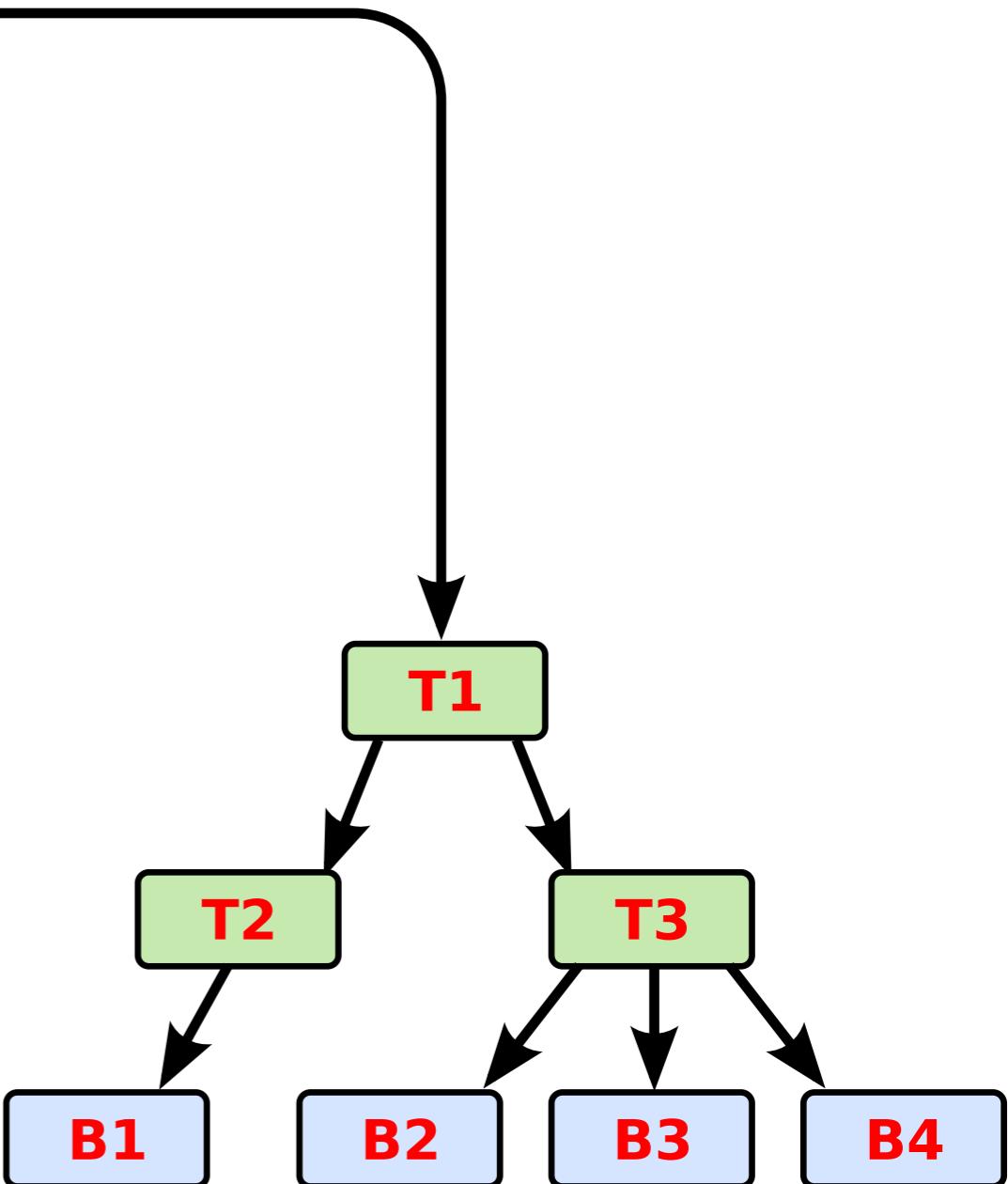
# Tipos

commit	tamaño de la revisión
tree	7d34d18d4763f49...
parent	2df316d2558819a...
author	Bart <bart@...> fecha
committer	Bart <bart@...> fecha
mensaje de la revisión	



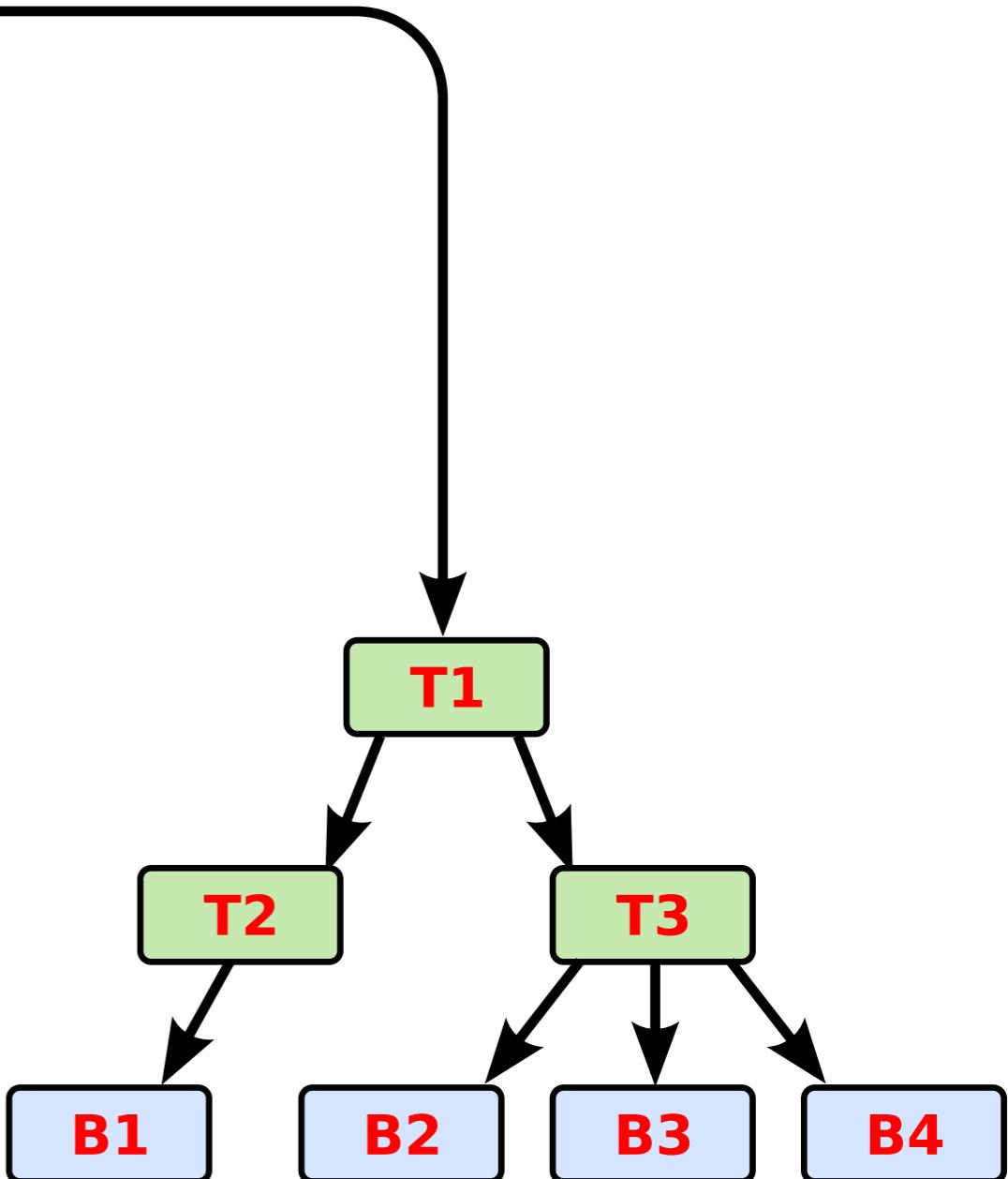
# Tipos

commit	tamaño de la revisión
tree	7d34d18d4763f49...
parent	2df316d2558819a...
author	Bart <bart@..> fecha
committer	Bart <bart@..> fecha
mensaje de la revisión	



# Tipos

commit	tamaño de la revisión
tree	7d34d18d4763f40
parent	2df316d2558819a...
author	Bart <bart@..> fecha
committer	Bart <bart@..> fecha
mensaje de la revisión	



# Tipos

commit	tamaño de la revisión
tree	7d34d18d4763f40
parent	2df316d2558819a...
author	Bart <bart@..> fecha
committer	Bart <bart@..> fecha
mensaje de la revisión	

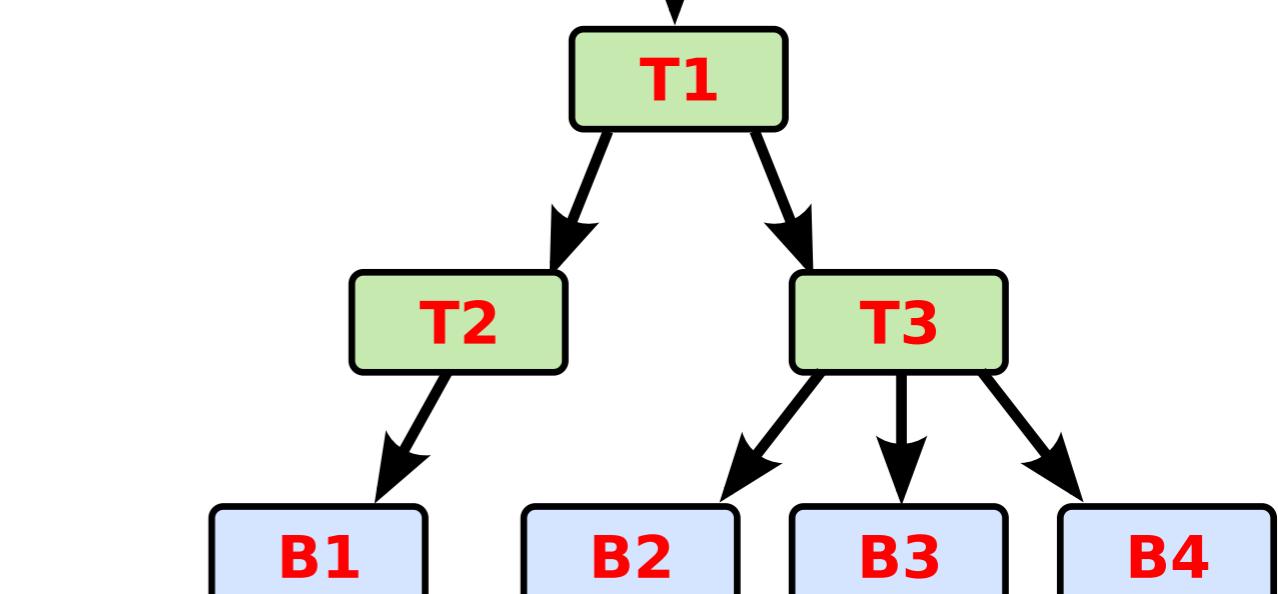
commit

B1

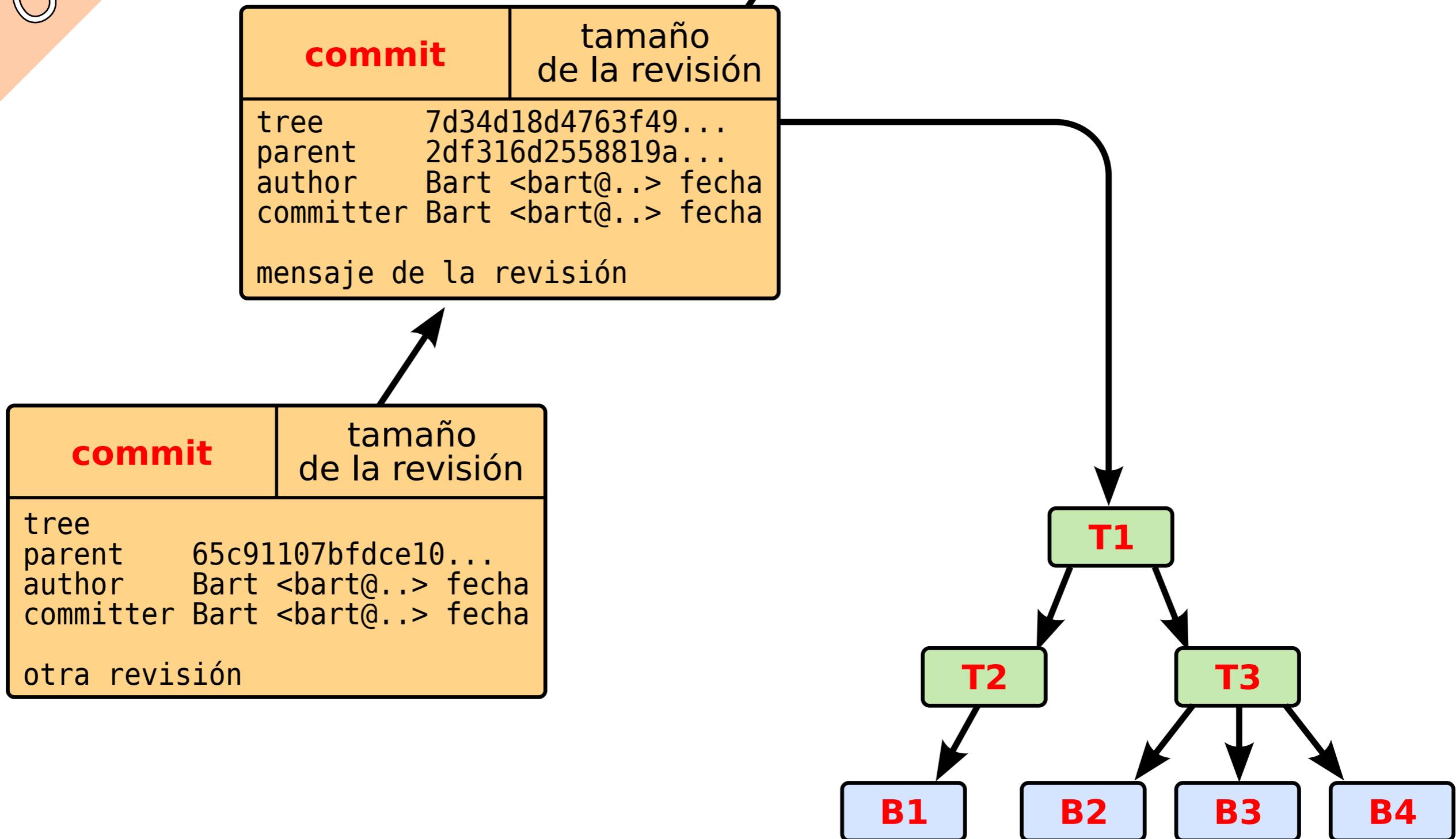
B2

B3

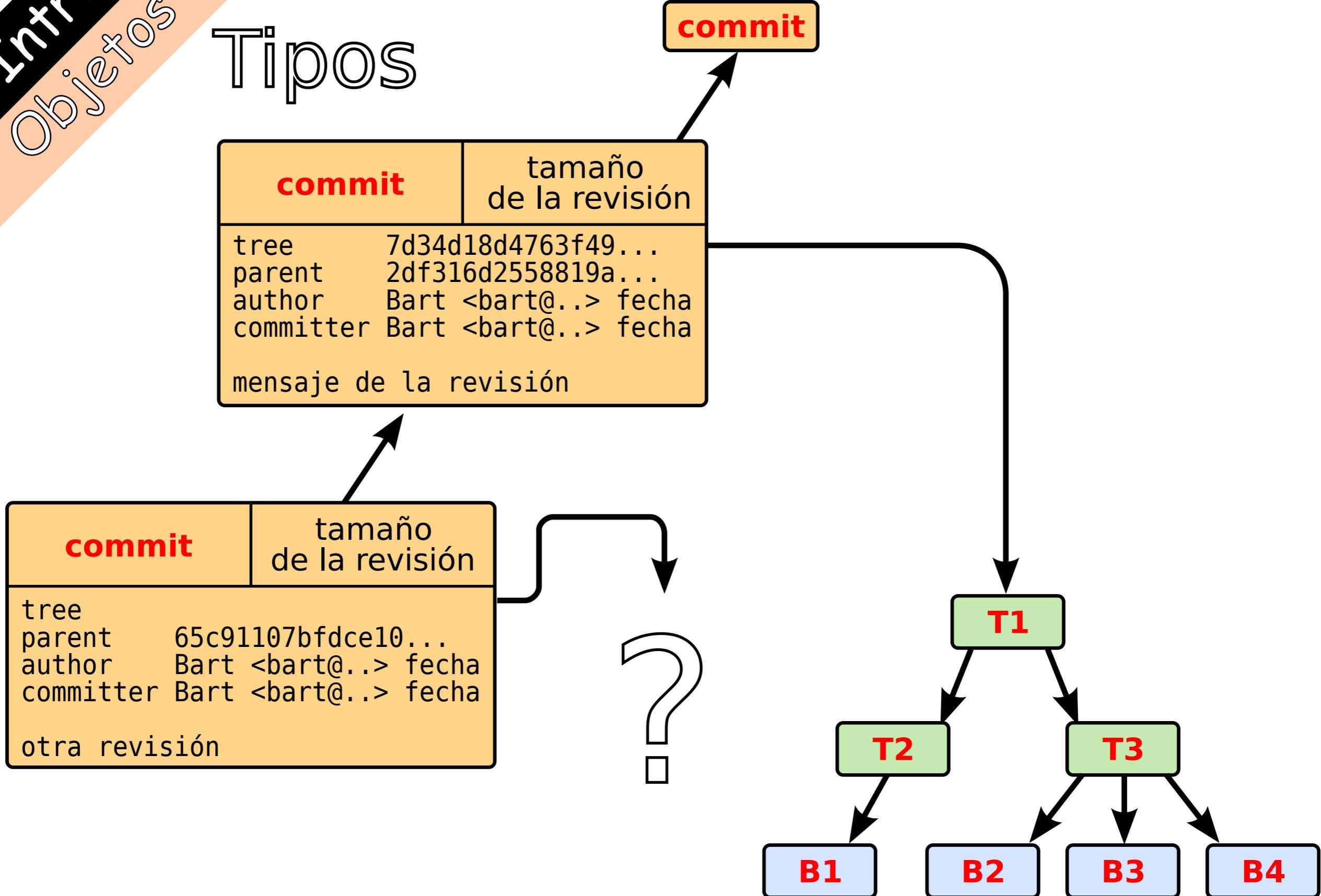
B4



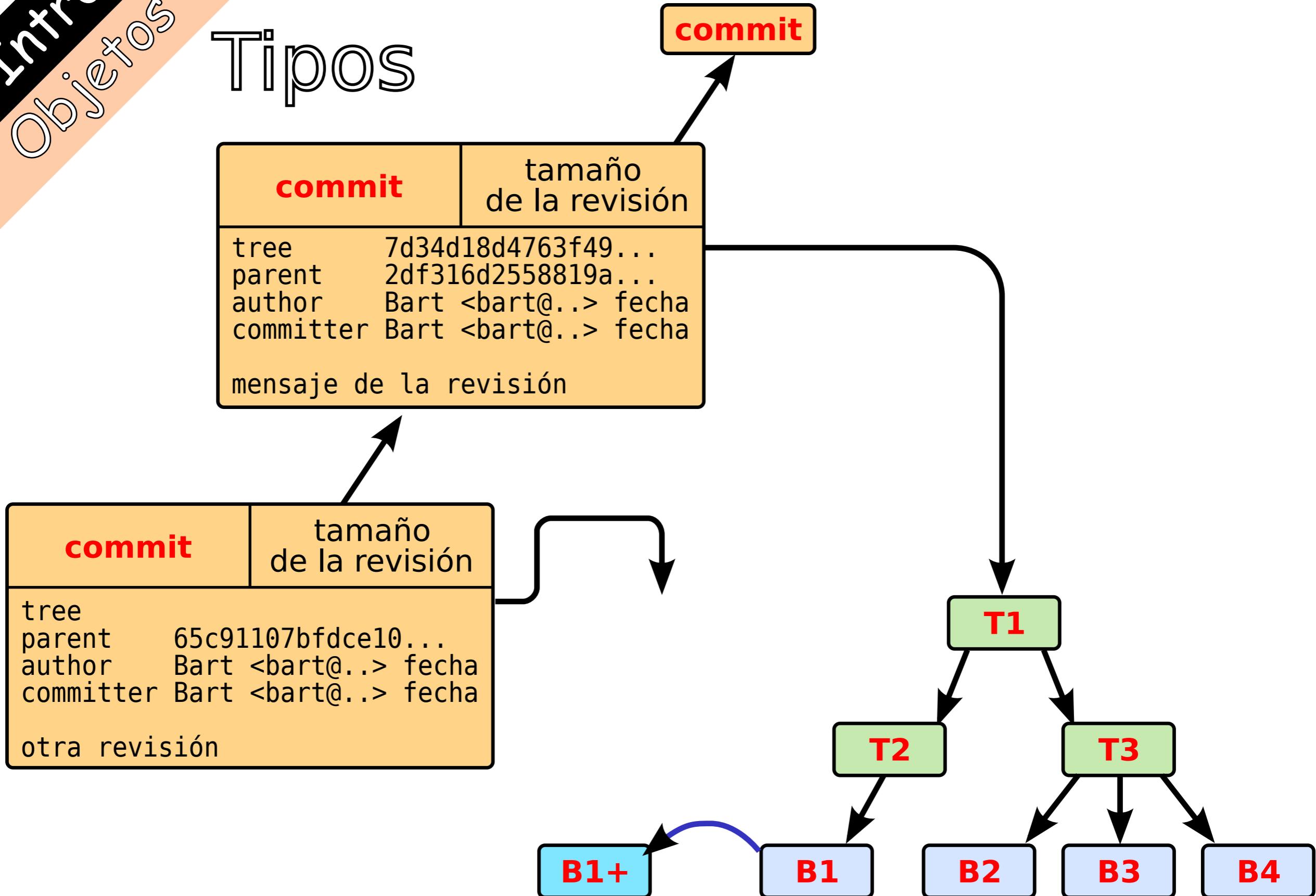
# Tipos



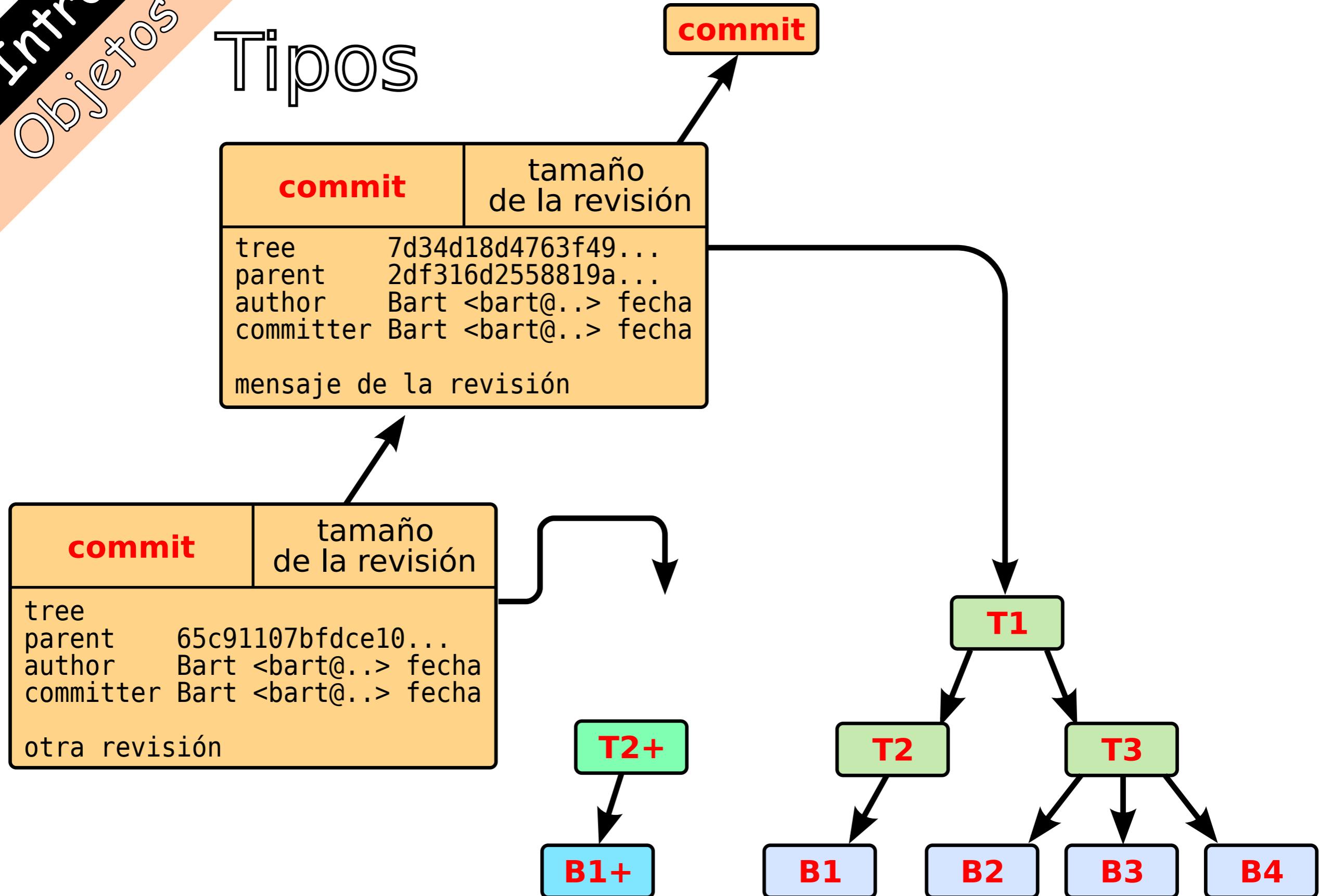
# Tipos



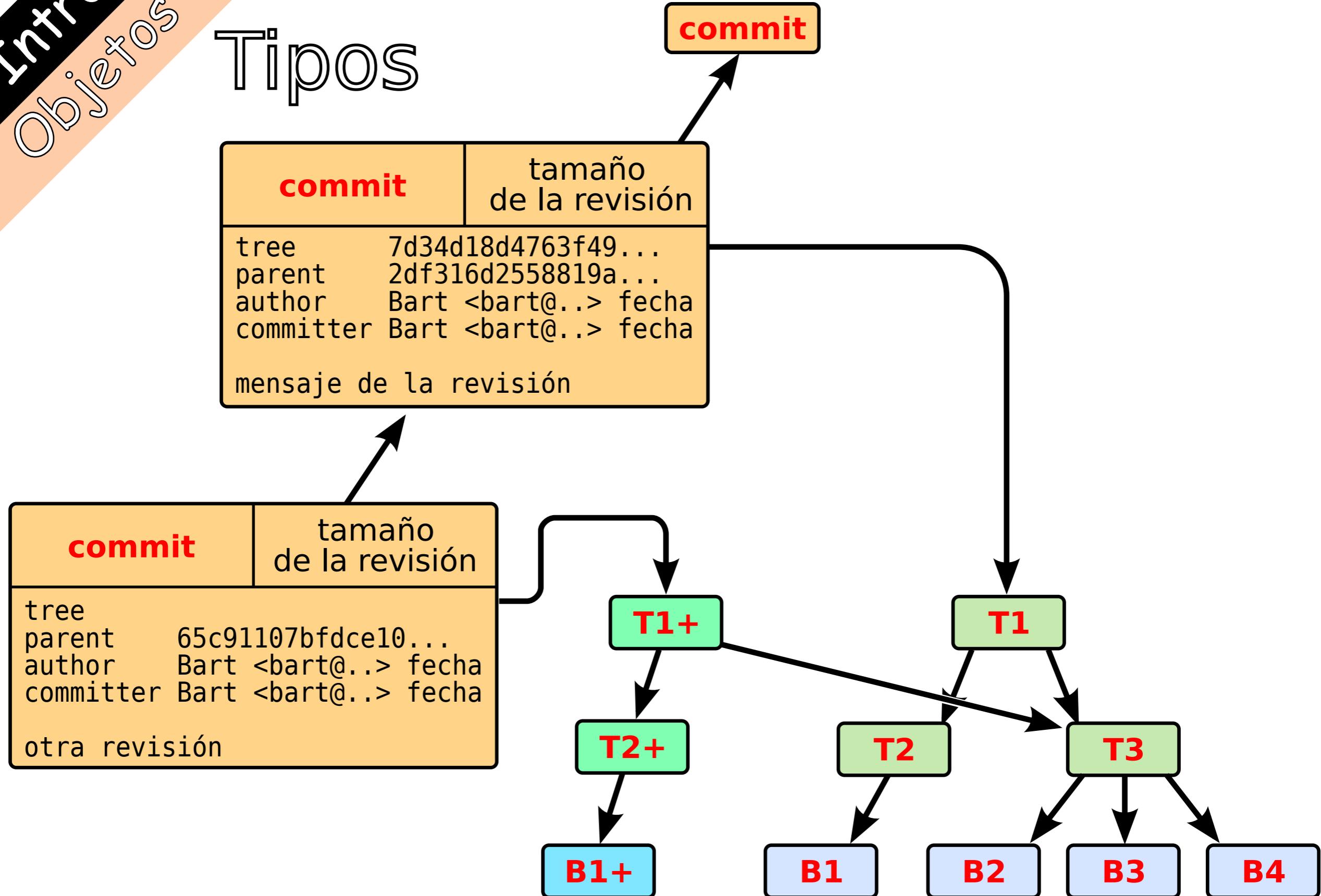
# Tipos



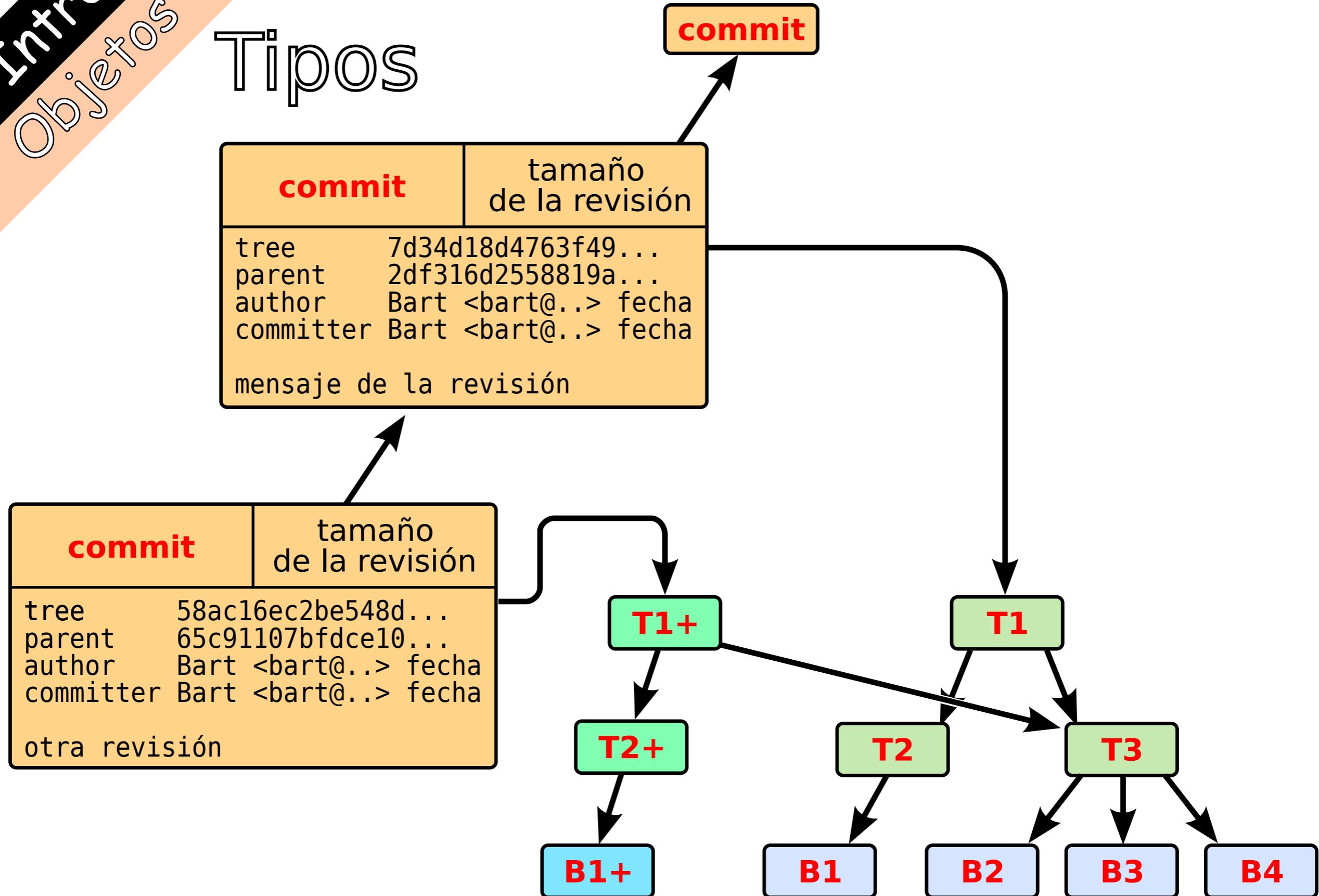
# Tipos



# Tipos



# Tipos



# Tipos

tag	tamaño de la etiqueta
object	8e1d4d898739f66...
type	commit
tag	v0.1
tagger	Bart <bart@..> fecha
mensaje de la etiqueta	

# Tipos

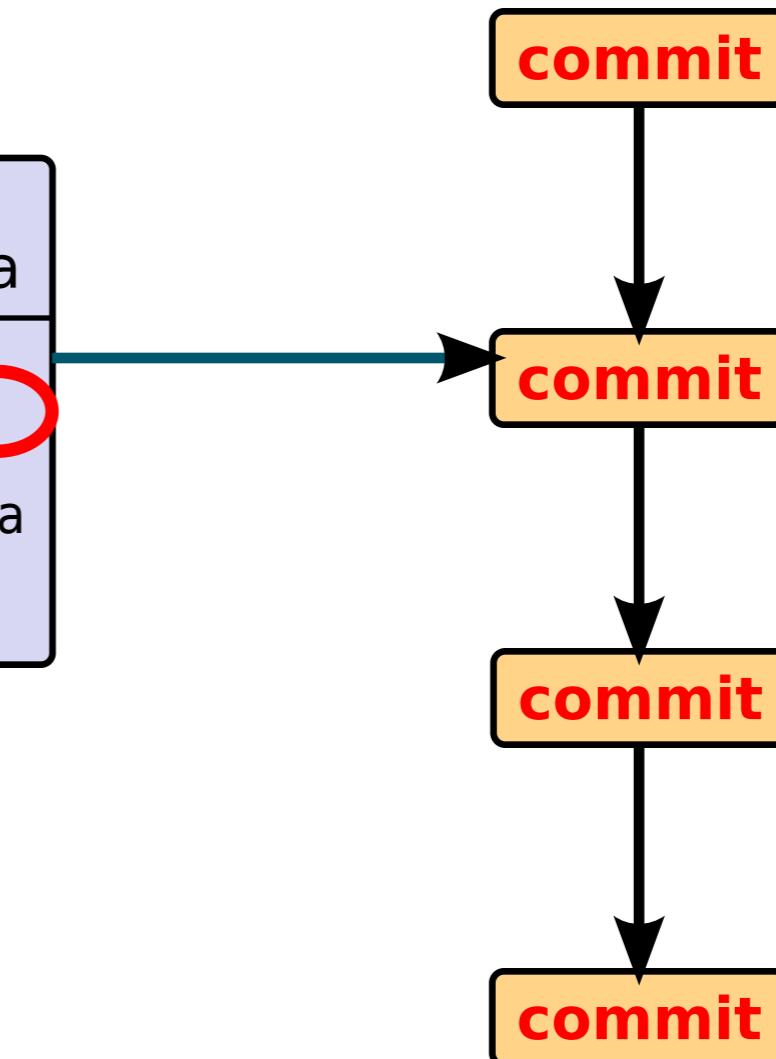
tag	tamaño de la etiqueta
object	8e1d4d898739f66...
type	commit
tag	v0.1
tagger	Bart <bart@..> fecha
mensaje de la etiqueta	

# Tipos

tag	tamaño de la etiqueta
object	8e1d4d808730f66
type	commit
tag	v0.1
tagger	Bart <bart@..> fecha
mensaje de la etiqueta	

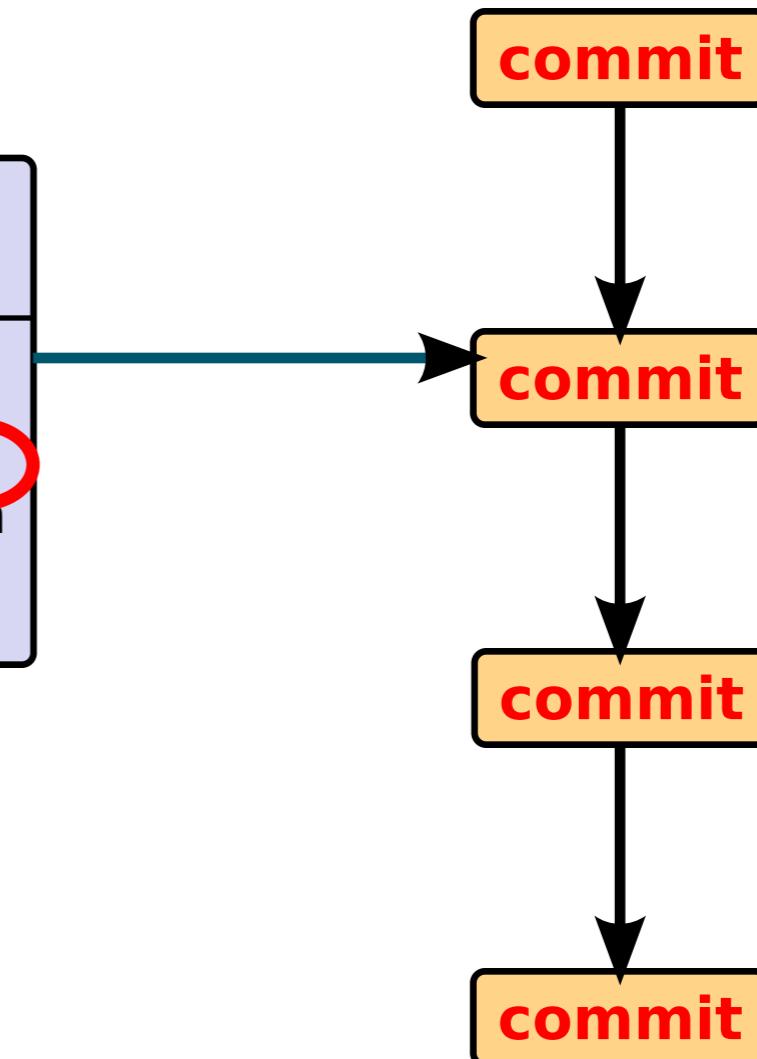
# Tipos

tag	tamaño de la etiqueta
object	8e1d4d808730f66
type	commit
tag	v0.1
tagger	Bart <bart@..> fecha
mensaje de la etiqueta	



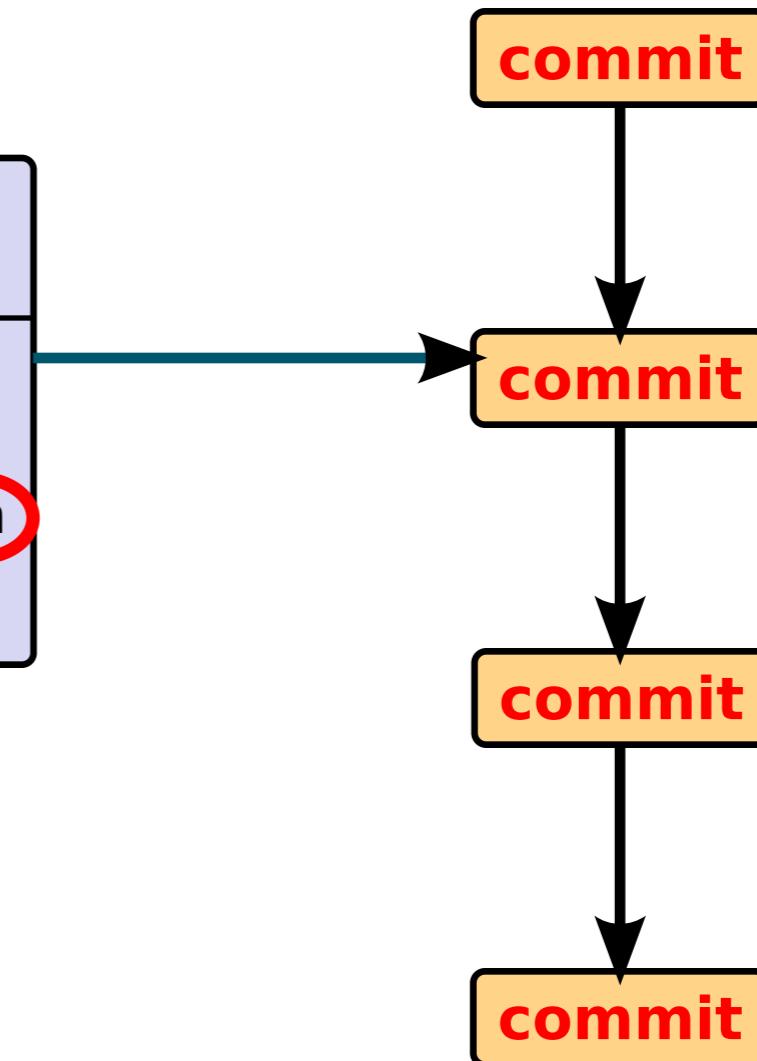
# Tipos

tag	tamaño de la etiqueta
object	8e1d4d898739f66...
type	commit
tag	v0.1
tagger	Barri <barri@...> fecha
mensaje de la etiqueta	



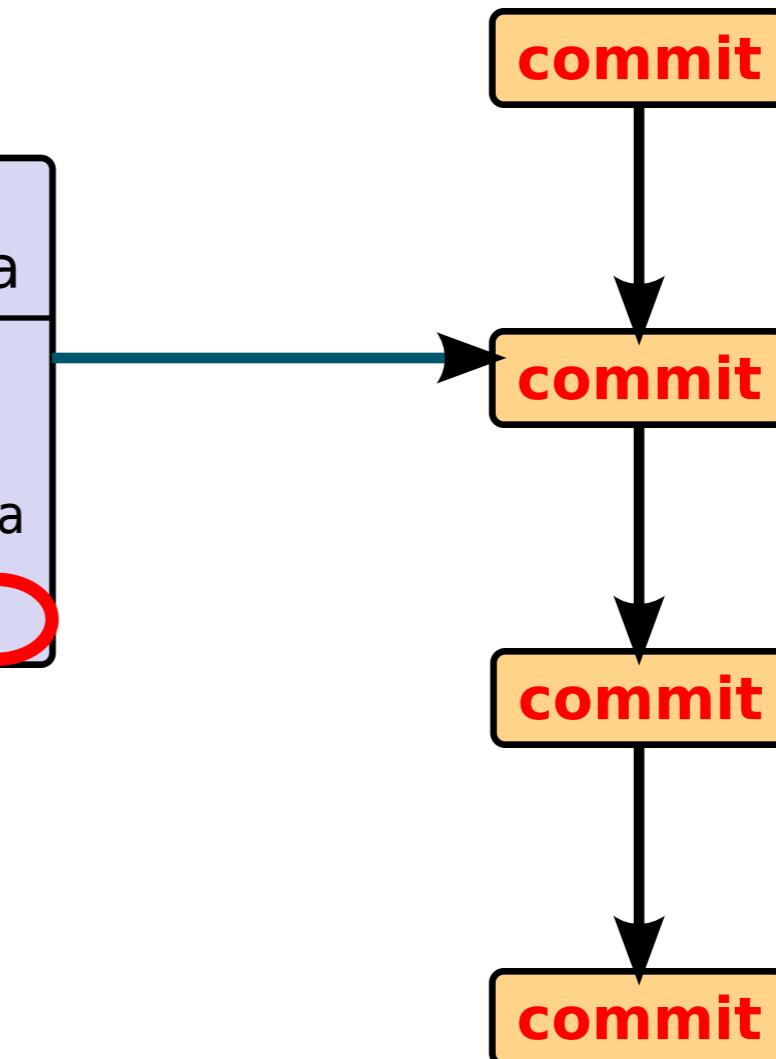
# Tipos

tag	tamaño de la etiqueta
object	8e1d4d898739f66...
type	commit
tag	v0.1
tagger	Bart <bart@..> fecha
mensaje de la etiqueta	



# Tipos

tag	tamaño de la etiqueta
object	8e1d4d898739f66...
type	commit
tag	v0.1
tagger	Bart <bart@..> fecha
mensaje de la etiqueta	



# Órdenes

```
git <opciones> <orden> <opciones>
```

git <opciones> <orden> <opciones>

...como SVN

git <opciones> <orden> <opciones>

...como SVN  
...no asusta

# Órdenes de Git (144)

add	daemon	index-pack	pack-redundant	shell
add--interactive	describe	init	pack-refs	shortlog
am	diff	init-db	parse-remote	show
annotate	diff-files	instaweb	patch-id	show-branch
apply	diff-index	log	peek-remote	show-index
archimport	diff-tree	lost-found	prune	show-ref
archive	difftool	ls-files	prune-packed	stage
bisect	difftool--helper	ls-remote	pull	stash
bisect--helper	fast-export	ls-tree	push	status
blame	fast-import	mailinfo	quiltimport	stripspace
branch	fetch	mailsplit	read-tree	submodule
bundle	fetch--tool	merge	rebase	svn
cat-file	fetch-pack	merge-base	rebase--interactive	symbolic-ref
check-attr	filter-branch	merge-file	receive-pack	tag
check-ref-format	fmt-merge-msg	merge-index	reflog	tar-tree
checkout	for-each-ref	merge-octopus	relink	unpack-file
checkout-index	format-patch	merge-one-file	remote	unpack-objects
cherry	fsck	merge-ours	repack	update-index
cherry-pick	fsck-objects	merge-recursive	repo-config	update-ref
citool	gc	merge-resolve	request-pull	update-server-info
clean	get-tar-commit-id	merge-subtree	rerere	upload-archive
clone	grep	merge-tree	reset	upload-pack
commit	gui	mergetool	rev-list	var
commit-tree	gui--askpass	mergetool--lib	rev-parse	verify-pack
config	hash-object	mktag	revert	verify-tag
count-objects	help	mktree	rm	web--browse
cvsexportcommit	http-fetch	mv	send-email	whatchanged
cvsimport	http-push	name-rev	send-pack	write-tree
cvsserver	imap-send	pack-objects	sh-setup	

# Inicialización y config. (4)

add	daemon	index-pack	pack-redundant	shell
add--interactive	describe	init	pack-refs	shortlog
am	diff	init-db	parse-remote	show
annotate	diff-files	instaweb	patch-id	show-branch
apply	diff-index	log	peek-remote	show-index
archimport	diff-tree	lost-found	prune	show-ref
archive	difftool	ls-files	prune-packed	stage
bisect	difftool--helper	ls-remote	pull	stash
bisect--helper	fast-export	ls-tree	push	status
blame	fast-import	mailinfo	quiltimport	stripspace
branch	fetch	mailsplit	read-tree	submodule
bundle	fetch--tool	merge	rebase	svn
cat-file	fetch-pack	merge-base	rebase--interactive	symbolic-ref
check-attr	filter-branch	merge-file	receive-pack	tag
check-ref-format	fmt-merge-msg	merge-index	reflog	tar-tree
checkout	for-each-ref	merge-octopus	relink	unpack-file
checkout-index	format-patch	merge-one-file	remote	unpack-objects
cherry	fsck	merge-recursive	repack	update-index
cherry-pick	fsck-objects	merge-resolve	repo-config	update-ref
citool	gc	merge-subtree	request-pull	update-server-info
clean	get-tar-commit-id	merge-tree	rerere	upload-archive
<b>clone</b>	grep	mergetool	reset	upload-pack
commit	gui	mergetool--lib	rev-list	var
commit-tree	gui--askpass	mktag	rev-parse	verify-pack
<b>config</b>	hash-object	mktree	revert	verify-tag
count-objects	help	mv	rm	web--browse
cvsexportcommit	http-fetch	name-rev	send-email	whatchanged
cvsimport	http-push	pack-objects	send-pack	write-tree
cvsserver	imap-send		sh-setup	

# Uso común (17)

**add**

add--interactive

am

annotate

apply

archimport

archive

bisect

bisect--helper

blame

**branch**

bundle

cat-file

check-attr

check-ref-format

**checkout**

checkout-index

cherry

cherry-pick

citool

clean

**clone****commit**

commit-tree

**config**

count-objects

cvsexportcommit

cvsimport

cvsserver

daemon

describe

**diff**

diff-files

diff-index

diff-tree

difftool

difftool--helper

fast-export

fast-import

**fetch**

fetch--tool

fetch-pack

filter-branch

fmt-merge-msg

for-each-ref

format-patch

fsck

fsck-objects

gc

get-tar-commit-id

**grep**

gui

gui--askpass

hash-object

help

http-fetch

http-push

imap-send

index-pack

**init**

init-db

instaweb

**log**

lost-found

ls-files

ls-remote

ls-tree

mailinfo

mailsplit

**merge**

merge-base

merge-file

merge-index

merge-octopus

merge-one-file

merge-ours

merge-recursive

merge-resolve

merge-subtree

merge-tree

mergetool

mergetool--lib

mktag

mktree

**mv**

name-rev

pack-objects

pack-redundant

pack-refs

parse-remote

patch-id

peek-remote

prune

prune-packed

**pull****push**

quiltimport

read-tree

rebase

rebase--interactive

receive-pack

reflog

relink

**remote**

repack

repo-config

request-pull

rerere

reset

rev-list

rev-parse

revert

**rm**

send-email

send-pack

sh-setup

shell

shortlog

**show**

show-branch

show-index

show-ref

**stage**

stash

**status**

stripspace

submodule

svn

symbolic-ref

**tag**

tar-tree

unpack-file

unpack-objects

update-index

update-ref

update-server-info

upload-archive

upload-pack

var

verify-pack

verify-tag

web--browse

whatchanged

write-tree

# Adicionales (12)

add	daemon	index-pack	pack-redundant	shell
add--interactive	describe	init	pack-refs	shortlog
am	diff	init-db	parse-remote	show
annotate	diff-files	instaweb	patch-id	show-branch
apply	diff-index	log	peek-remote	show-index
archimport	diff-tree	lost-found	prune	show-ref
archive	difftool	ls-files	prune-packed	stage
bisect	difftool--helper	ls-remote	pull	stash
bisect--helper	fast-export	ls-tree	push	status
blame	fast-import	mailinfo	quiltimport	stripspace
branch	fetch	mailsplit	read-tree	submodule
bundle	fetch--tool	merge	rebase	svn
cat-file	fetch-pack	merge-base	rebase--interactive	symbolic-ref
check-attr	filter-branch	merge-file	receive-pack	tag
check-ref-format	fmt-merge-msg	merge-index	reflog	tar-tree
checkout	for-each-ref	merge-octopus	relink	unpack-file
checkout-index	format-patch	merge-one-file	remote	unpack-objects
cherry	fsck	merge-recursive	repack	update-index
cherry-pick	fsck-objects	merge-resolve	repo-config	update-ref
citool	gc	merge-subtree	request-pull	update-server-info
clean	get-tar-commit-id	merge-tree	rerere	upload-archive
clone	grep	mergetool	reset	upload-pack
commit	gui	mergetool--lib	rev-list	var
commit-tree	gui--askpass	mktag	rev-parse	verify-pack
config	hash-object	mktree	revert	verify-tag
count-objects	help	mv	rm	web--browse
cvsexportcommit	http-fetch	name-rev	send-email	whatchanged
cvsimport	http-push	pack-objects	send-pack	write-tree
cvsserver	imap-send		sh-setup	

# Para integradores (11)

add	daemon	index-pack	pack-redundant	shell
add--interactive	describe	init	pack-refs	shortlog
am	diff	init-db	parse-remote	show
annotate	diff-files	instaweb	patch-id	show-branch
apply	diff-index	log	peek-remote	show-index
archimport	diff-tree	lost-found	prune	show-ref
archive	difftool	ls-files	prune-packed	stage
bisect	difftool--helper	ls-remote	pull	stash
bisect--helper	fast-export	ls-tree	push	status
blame	fast-import	mailinfo	quiltimport	stripspace
branch	fetch	mailsplit	read-tree	submodule
bundle	fetch--tool	merge	rebase	svn
cat-file	fetch-pack	merge-base	rebase--interactive	symbolic-ref
check-attr	filter-branch	merge-file	receive-pack	tag
check-ref-format	fmt-merge-msg	merge-index	reflog	tar-tree
checkout	for-each-ref	merge-octopus	relink	unpack-file
checkout-index	format-patch	merge-one-file	remote	unpack-objects
cherry	fsck	merge-recursive	repack	update-index
cherry-pick	fsck-objects	merge-subtree	repo-config	update-ref
citool	gc	merge-tree	request-pull	update-server-info
clean	get-tar-commit-id	mergetool	rerere	upload-archive
clone	grep	mergetool--lib	reset	upload-pack
commit	gui	mktag	rev-list	var
commit-tree	gui--askpass	mktree	rev-parse	verify-pack
config	hash-object	mv	revert	verify-tag
count-objects	help	name-rev	rm	web--browse
cvsexportcommit	http-fetch	pack-objects	send-email	whatchanged
cvsimport	http-push		send-pack	write-tree
cvsserver	imap-send		sh-setup	

# Uso específico (13)

add	daemon	index-pack	pack-redundant	shell
add--interactive	describe	init	pack-refs	shortlog
am	diff	init-db	parse-remote	show
annotate	diff-files	instaweb	patch-id	show-branch
apply	diff-index	log	peek-remote	show-index
archimport	diff-tree	lost-found	prune	show-ref
archive	difftool	ls-files	prune-packed	stage
bisect	difftool--helper	ls-remote	pull	stash
bisect--helper	fast-export	ls-tree	push	status
blame	fast-import	mailinfo	quiltimport	stripspace
branch	fetch	mailsplit	read-tree	submodule
bundle	fetch--tool	merge	rebase	svn
cat-file	fetch-pack	merge-base	rebase--interactive	symbolic-ref
check-attr	filter-branch	merge-file	receive-pack	tag
check-ref-format	fmt-merge-msg	merge-index	reflog	tar-tree
checkout	for-each-ref	merge-octopus	relink	unpack-file
checkout-index	format-patch	merge-one-file	remote	unpack-objects
cherry	fsck	merge-recursive	repack	update-index
cherry-pick	fsck-objects	merge-resolve	repo-config	update-ref
citool	gc	merge-subtree	request-pull	update-server-info
clean	get-tar-commit-id	merge-tree	rerere	upload-archive
clone	grep	mergetool	reset	upload-pack
commit	gui	mergetool--lib	rev-list	var
commit-tree	gui--askpass	mktag	rev-parse	verify-pack
config	hash-object	mktree	revert	verify-tag
count-objects	help	mv	rm	web--browse
cvsexportcommit	http-fetch	name-rev	send-email	whatchanged
cvsimport	http-push	pack-objects	send-pack	write-tree
cvsserver	imap-send		sh-setup	

# ... el resto es "fontanería"

add	daemon	index-pack	pack-redundant	shell
add--interactive	describe	init	pack-refs	shortlog
am	diff	init-db	parse-remote	show
annotate	diff-files	instaweb	patch-id	show-branch
apply	diff-index	log	peek-remote	show-index
archimport	diff-tree	lost-found	prune	show-ref
archive	difftool	ls-files	prune-packed	stage
bisect	difftool--helper	ls-remote	pull	stash
bisect--helper	fast-export	ls-tree	push	status
blame	fast-import	mailinfo	quiltimport	stripspace
branch	fetch	mailsplit	read-tree	submodule
bundle	fetch--tool	merge	rebase	svn
cat-file	fetch-pack	merge-base	rebase--interactive	symbolic-ref
check-attr	filter-branch	merge-file	receive-pack	tag
check-ref-format	fmt-merge-msg	merge-index	reflog	tar-tree
checkout	for-each-ref	merge-octopus	relink	unpack-file
checkout-index	format-patch	merge-one-file	remote	unpack-objects
cherry	fsck	merge-ours	repack	update-index
cherry-pick	fsck-objects	merge-recursive	repo-config	update-ref
citool	gc	merge-resolve	request-pull	update-server-info
clean	get-tar-commit-id	merge-subtree	rerere	upload-archive
clone	grep	merge-tree	reset	upload-pack
commit	gui	mergetool	rev-list	var
commit-tree	gui--askpass	mergetool--lib	rev-parse	verify-pack
config	hash-object	mktag	revert	verify-tag
count-objects	help	mktree	rm	web--browse
cvsexportcommit	http-fetch	mv	send-email	whatchanged
cvsimport	http-push	name-rev	send-pack	write-tree
cvsserver	imap-send	pack-objects	sh-setup	

# Órdenes más comunes

**add**

add--interactive

am

annotate

apply

archimport

archive

bisect

bisect--helper

blame

**branch**

bundle

cat-file

check-attr

check-ref-format

**checkout**

checkout-index

cherry

cherry-pick

citool

clean

**clone**

**commit**

commit-tree

**config**

count-objects

cvsexportcommit

cvsimport

cvsserver

daemon

describe

**diff**

diff-files

diff-index

diff-tree

difftool

difftool--helper

fast-export

fast-import

**fetch**

fetch--tool

fetch-pack

filter-branch

fmt-merge-msg

for-each-ref

format-patch

fsck

fsck-objects

gc

get-tar-commit-id

**grep**

gui

gui--askpass

hash-object

help

http-fetch

http-push

imap-send

index-pack

**init**

init-db

instaweb

**log**

lost-found

ls-files

ls-remote

ls-tree

mailinfo

mailsplit

**merge**

merge-base

merge-file

merge-index

merge-octopus

merge-one-file

merge-recursive

merge-resolve

merge-subtree

merge-tree

mergetool

mergetool--lib

mktag

mktree

**mv**

name-rev

pack-objects

pack-redundant

pack-refs

parse-remote

patch-id

peek-remote

prune

prune-packed

**pull**

**push**

quiltimport

read-tree

rebase

rebase--interactive

receive-pack

reflog

relink

**remote**

repack

repo-config

request-pull

rerere

reset

rev-list

rev-parse

revert

**rm**

send-email

send-pack

sh-setup

shell

shortlog

**show**

show-branch

show-index

show-ref

stage

stash

**status**

stripspace

submodule

svn

symbolic-ref

**tag**

tar-tree

unpack-file

unpack-objects

update-index

update-ref

update-server-info

upload-archive

upload-pack

var

verify-pack

verify-tag

web--browse

whatchanged

write-tree

# Ayuda

git help

# git help

```
usage: git [--version] [--exec-path[=GIT_EXEC_PATH]] [--html-path]
           [-p|--paginate|--no-pager] [--bare] [--git-dir=GIT_DIR]
           [--work-tree=GIT_WORK_TREE] [--help] COMMAND [ARG$]
```

The most commonly used git commands are:

add	Add file contents to the index
bisect	Find by binary search the change that introduced a bug
branch	List, create, or delete branches
checkout	Checkout a branch or paths to the working tree
clone	Clone a repository into a new directory
commit	Record changes to the repository
diff	Show changes between commits, commit and working tree, etc
fetch	Download objects and refs from another repository
grep	Print lines matching a pattern
init	Create an empty git repository or reinitialize an existing one
log	Show commit logs
merge	Join two or more development histories together
mv	Move or rename a file, a directory, or a symlink
pull	Fetch from and merge with another repository or a local branch
push	Update remote refs along with associated objects
rebase	Forward-port local commits to the updated upstream head
reset	Reset current HEAD to the specified state
rm	Remove files from the working tree and from the index
show	Show various types of objects
status	Show the working tree status
tag	Create, list, delete or verify a tag object signed with GPG

See 'git help COMMAND' for more information on a specific command.

# git help

```
usage: git [--version] [--exec-path[=GIT_EXEC_PATH]] [--html-path]
           [-p|--paginate|--no-pager] [--bare] [--git-dir=GIT_DIR]
           [--work-tree=GIT_WORK_TREE] [--help] COMMAND [ARGST]
```

The most commonly used git commands are:

add	Add file contents to the index
bisect	Find by binary search the change that introduced a bug
branch	List, create, or delete branches
checkout	Checkout a branch or paths to the working tree
clone	Clone a repository into a new directory
commit	Record changes to the repository
diff	Show changes between commits, commit and working tree, etc
fetch	Download objects and refs from another repository
grep	Print lines matching a pattern
init	Create an empty git repository or reinitialize an existing one
log	Show commit logs
merge	Join two or more development histories together
mv	Move or rename a file, a directory, or a symlink
pull	Fetch from and merge with another repository or a local branch
push	Update remote refs along with associated objects
rebase	Forward-port local commits to the updated upstream head
reset	Reset current HEAD to the specified state
rm	Remove files from the working tree and from the index
show	Show various types of objects
status	Show the working tree status
tag	Create, list, delete or verify a tag object signed with GPG

See 'git help COMMAND' for more information on a specific command.

# git help

```
usage: git [--version] [--exec-path[=GIT_EXEC_PATH]] [--html-path]
           [-p|--paginate|--no-pager] [--bare] [--git-dir=GIT_DIR]
           [--work-tree=GIT_WORK_TREE] [--help] COMMAND [ARG$]
```

The most commonly used git commands are:

add	Add file contents to the index
bisect	Find by binary search the change that introduced a bug
branch	List, create, or delete branches
checkout	Checkout a branch or paths to the working tree
clone	Clone a repository into a new directory
commit	Record changes to the repository
diff	Show changes between commits, commit and working tree, etc
fetch	Download objects and refs from another repository
grep	Print lines matching a pattern
init	Create an empty git repository or reinitialize an existing one
log	Show commit logs
merge	Join two or more development histories together
mv	Move or rename a file, a directory, or a symlink
pull	Fetch from and merge with another repository or a local branch
push	Update remote refs along with associated objects
rebase	Forward-port local commits to the updated upstream head
reset	Reset current HEAD to the specified state
rm	Remove files from the working tree and from the index
show	Show various types of objects
status	Show the working tree status
tag	Create, list, delete or verify a tag object signed with GPG

See 'git help COMMAND' for more information on a specific command.

git help --all

# git help --all

```
usage: git [--version] [--exec-path[=GIT_EXEC_PATH]] [--html-path]
           [-p|--paginate|--no-pager] [--bare] [--git-dir=GIT_DIR]
           [--work-tree=GIT_WORK_TREE] [--help] COMMAND [ARGST]
```

```
available git commands in '/home/oxygen/bart/usr/libexec/git-core'
```

add	gc	quiltimport
add--interactive	get-tar-commit-id	read-tree
am	grep	rebase
annotate	gui	rebase--interactive
apply	gui--askpass	receive-pack
archimport	hash-object	reflog
archive	help	relink
bisect	http-fetch	remote
bisect--helper	http-push	repack
blame	imap-send	repo-config
branch	index-pack	request-pull
bundle	init	rerere
cat-file	init-db	reset
check-attr	instaweb	rev-list
check-ref-format	log	rev-parse
checkout	lost-found	revert
checkout-index	ls-files	rm
cherry	ls-remote	send-email
cherry-pick	ls-tree	send-pack
citool	mailinfo	sh-setup
clean	mailsplit	shell
clone	merge	shortlog
commit	merge-base	show
commit-tree	merge-file	show-branch

--- 8< ---

# git help --all

```
usage: git [--version] [--exec-path[=GIT_EXEC_PATH]] [--html-path]
           [-p|--paginate|--no-pager] [--bare] [--git-dir=GIT_DIR]
           [--work-tree=GIT_WORK_TREE] [--help] COMMAND [ARGST]
```

```
available git commands in '/home/oxygen/bart/usr/libexec/git-core'
```

add	gc	quiltimport
add--interactive	get-tar-commit-id	read-tree
am	grep	rebase
annotate	gui	rebase--interactive
apply	gui--askpass	receive-pack
archimport	hash-object	reflog
archive	help	relink
bisect	http-fetch	remote
bisect--helper	http-push	repack
blame	imap-send	repo-config
branch	index-pack	request-pull
bundle	init	rerere
cat-file	init-db	reset
check-attr	instaweb	rev-list
check-ref-format	log	rev-parse
checkout	lost-found	revert
checkout-index	ls-files	rm
cherry	ls-remote	send-email
cherry-pick	ls-tree	send-pack
citool	mailinfo	sh-setup
clean	mailsplit	shell
clone	merge	shortlog
commit	merge-base	show
commit-tree	merge-file	show-branch

--- 8< ---

144

git <orden> -h

git add -h

# git add -h

```
usage: git add [options] [--] <filepattern>...

-n, --dry-run          dry run
-v, --verbose           be verbose

-i, --interactive       interactive picking
-p, --patch              interactive patching
-e, --edit                edit current diff and apply
-f, --force               allow adding otherwise ignored files
-u, --update             update tracked files
-N, --intent-to-add      record only the fact that the path will be added later
-A, --all                  add all, noticing removal of tracked files
--refresh                 don't add, only refresh the index
--ignore-errors           just skip files which cannot be added because of errors
```

# git add -h

```
usage: git add [options] [--] <filepattern>...
```

-n, --dry-run	dry run
-v, --verbose	be verbose
-i, --interactive	interactive picking
-p, --patch	interactive patching
-e, --edit	edit current diff and apply
-f, --force	allow adding otherwise ignored files
-u, --update	update tracked files
-N, --intent-to-add	record only the fact that the path will be added later
-A, --all	add all, noticing removal of tracked files
--refresh	don't add, only refresh the index
--ignore-errors	just skip files which cannot be added because of errors

# git add -h

usage: git add [options] [--] <filepattern>...

-n, --dry-run	dry run
-v, --verbose	be verbose
-i, --interactive	interactive picking
-p, --patch	interactive patching
-e, --edit	edit current diff and apply
-f, --force	allow adding otherwise ignored files
-u, --update	update tracked files
-N, --intent-to-add	record only the fact that the path will be added later
-A, --all	add all, noticing removal of tracked files
--refresh	don't add, only refresh the index
--ignore-errors	just skip files which cannot be added because of errors

`man git-<orden>`

`man git-<orden>`

`git help <orden>`

`man git-<orden>`

`git help <orden>`

`git <orden> --help`

# Configuración

# Repositorio

\$TRABAJO/.git/config

# Repositorio

\$TRABAJO/.git/config

# Repositorio

`$TRABAJO/.git/config`

`git config <variable>`

# Repositorio

\$TRABAJO/.git/config

git config <variable> <valor>

# Repositorio

\$TRABAJO/.git/config

```
git config alias.st "diff --stat"
```

# Repositorio

\$TRABAJO/.git/config

```
git config alias.st "diff --stat"
```

```
[alias]  
st = diff --stat
```

# Global

`$HOME/.gitconfig`

# Global

`$HOME/.gitconfig`

`git config --global <variable>`

# Global

\$HOME/.gitconfig

```
git config --global <variable> <valor>
```

# Global

\$HOME/.gitconfig

```
git config --global alias.st "diff --stat"
```

# Sobre ti

# Sobre ti

```
git config --global user.name "Tu Nombre"  
git config --global user.email "tu@dominio"
```

# Sobre ti

```
git config --global user.name "Tu Nombre"  
git config --global user.email "tu@dominio"  
  
git config --global color.pager true  
git config --global color.ui auto  
git config --global core.editor ????
```

git help config

~200 opciones

# Uso de Git

# Primeros pasos

# Nuevo repositorio

```
mkdir proyecto-nuevo  
cd proyecto-nuevo  
git init
```

# Nuevo repositorio

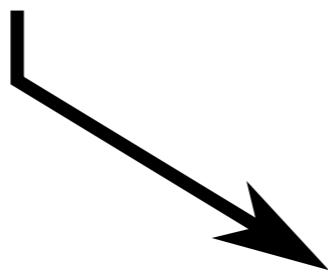
```
cd proyecto-existente  
git init
```

# Nuevo repositorio

```
cd $TRABAJO  
git init
```

# Nuevo repositorio

```
cd $TRABAJO  
git init
```



\$TRABAJO/.git

# Preparación de revisiones ("staging")

# ¿Qué incluir en la revisión?

# ¿Qué incluir en la revisión?

- nuevo contenido y modificaciones

```
git add <fichero>
```

```
git add .
```

# ¿Qué incluir en la revisión?

- nuevo contenido y modificaciones

```
git add <fichero>
```

```
git add .
```

- eliminaciones

```
git rm <fichero>
```

# ¿Qué incluir en la revisión?

- nuevo contenido y modificaciones

```
git add <fichero>
```

```
git add .
```

- eliminaciones

```
git rm <fichero>
```

- renombrados

```
git mv <antiguo> <nuevo>
```

# ¿Qué incluir en la revisión?

- nuevo contenido y modificaciones

`git add <fichero>`

`git add .`

- eliminaciones

`git rm <fichero>`

- renombrados

`git mv <antiguo> <nuevo>`

# ¿Qué no incluir?

# ¿Qué no incluir?

\$TRABAJO/.gitignore

- bajo control de versiones

\$TRABAJO/.git/info/exclude

- sólo para este repositorio

# ¿Qué no incluir?

\$TRABAJO/.gitignore

```
#comentario
*~
*[oa]
.*.Swp
!include.o
generado/*.c
```

# ¿Qué no incluir?

\$TRABAJO/.gitignore

```
#comentario
*~
*[oa]
.*.Swp
!include.o
generado/*.c
```

# ¿Qué no incluir?

\$TRABAJO/.gitignore

#comentario

```
*~  
*[oa]  
. *.swp  
!include.o  
generado/*.c
```

# ¿Qué no incluir?

\$TRABAJO/.gitignore

```
#comentario
*~
*[oa]
*.Swp
!include.o
generado/*.c
```

# ¿Qué no incluir?

\$TRABAJO/.gitignore

```
#comentario
*~
*[oa]
.*.Swp
!include.o
generado/*.c
```

# ¿Qué está preparado?

git status

# ¿Qué está preparado?

## git status

On branch master

Changes to be committed:

(use "git reset HEAD <file>..." to unstage)

new file: añadido-con-git-add  
deleted: borrado-con-git-rm

Changed but not updated:

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

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

modified: fichero-cambiado  
deleted: borrado-con-rm

Untracked files:

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

fichero-fuera-de-control-de-versiones

# ¿Qué está preparado?

## git status

On branch master

Changes to be committed:

(use "git reset HEAD <file>..." to unstage)

new file: añadido-con-git-add  
deleted: borrado-con-git-rm

Changed but not updated:

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

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

modified: fichero-cambiado  
deleted: borrado-con-rm

Untracked files:

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

fichero-fuera-de-control-de-versiones

# ¿Qué está preparado?

## git status

On branch master

Changes to be committed:

(use "git reset HEAD <file>..." to unstage)

new file: añadido-con-git-add  
deleted: borrado-con-git-rm

Changed but not updated:

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

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

modified: fichero-cambiado  
deleted: borrado-con-rm

Untracked files:

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

fichero-fuera-de-control-de-versiones

# ¿Qué está preparado?

## git status

On branch master

Changes to be committed:

(use "git reset HEAD <file>..." to unstage)

**new file:  
deleted:**

añadido-con-git-add  
borrado-con-git-rm

Changed but not updated:

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

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

**modified:  
deleted:**

fichero-cambiado  
borrado-con-rm

Untracked files:

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

**fichero-fuera-de-control-de-versiones**

# ¿Qué está preparado?

## git status

On branch master

Changes to be committed:

(use "git reset HEAD <file>..." to unstage)

new file: añadido-con-git-add  
deleted: borrado-con-git-rm

Changed but not updated:

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

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

modified: fichero-cambiado  
deleted: borrado-con-rm

Untracked files:

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

fichero-fuera-de-control-de-versiones

# ¿Qué está preparado?

## git status

On branch master

Changes to be committed:

(use "git reset HEAD <file>..." to unstage)

new file: añadido-con-git-add  
deleted: borrado-con-git-rm

Changed but not updated:

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

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

modified: fichero-cambiado  
deleted: borrado-con-rm

Untracked files:

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

fichero-fuera-de-control-de-versiones

# ¿Qué está preparado?

## git status

On branch master

Changes to be committed:

(use "git reset HEAD <file>..." to unstage)

new file: añadido-con-git-add  
deleted: borrado-con-git-rm

git diff  
--staged

Changed but not updated:

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

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

modified: fichero-cambiado  
deleted: borrado-con-rm

Untracked files:

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

fichero-fuera-de-control-de-versiones

# ¿Qué está preparado?

## git status

On branch master

Changes to be committed:

(use "git reset HEAD <file>..." to unstage)

new file: añadido-con-git-add  
deleted: borrado-con-git-rm

Changed but not updated:

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

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

modified: fichero-cambiado  
deleted: borrado-con-rm

Untracked files:

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

fichero-fuera-de-control-de-versiones

# ¿Qué está preparado?

## git status

On branch master

Changes to be committed:

(use "git reset HEAD <file>..." to unstage)

new file: añadido-con-git-add  
deleted: borrado-con-git-rm

Changed but not updated:

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

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

modified: fichero-cambiado  
deleted: borrado-con-rm

git diff

Untracked files:

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

fichero-fuera-de-control-de-versiones

# ¿Qué está preparado?

## git status

On branch master

Changes to be committed:

(use "git reset HEAD <file>..." to unstage)

new file: añadido-con-git-add  
deleted: borrado-con-git-rm

Changed but not updated:

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

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

modified: fichero-cambiado  
deleted: borrado-con-rm

Untracked files:

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

fichero-fuera-de-control-de-versiones

# Creación de revisiones ("commits")

## git commit

- lanza **\$EDITOR**
- permite editar el mensaje
- sólo entra lo preparado

`git commit <fichero>...`  
- sólo los ficheros indicados

`git commit <fichero>...`

- sólo los ficheros indicados

`git commit -a`

- todos los ficheros modificados  
bajo control de versiones

`git commit <fichero>...`

- sólo los ficheros indicados

`git commit -a`

- todos los ficheros modificados  
bajo control de versiones

`git commit -m "mensaje"`

- se indica el mensaje
- sin \$EDITOR

```
git init
echo algo > prueba
git add prueba
git commit -m "mi mensaje"
```

```
git init
echo algo > prueba
git add prueba
git commit -m "mi mensaje"
```

```
git init
echo algo > prueba
git add prueba
git commit -m "mi mensaje"
```

```
[master (root-commit) 3fb8a8e] mi mensaje
 1 files changed, 1 insertions(+), 0 deletions(-)
 create mode 100644 prueba
```

```
git init
echo algo > prueba
git add prueba
git commit -m "mi mensaje"
```

```
[master (root-commit) 3fb8a8e] mi mensaje
 1 files changed, 1 insertions(+), 0 deletions(-)
 create mode 100644 prueba
```

```
git init
echo algo > prueba
git add prueba
git commit -m "mi mensaje"
```

```
[master (root-commit) 3fb8a8e] mi mensaje
 1 files changed, 1 insertions(+), 0 deletions(-)
 create mode 100644 prueba
```

```
git init
echo algo > prueba
git add prueba
git commit -m "mi mensaje"
```

```
[master (root-commit) 3fb8a8e] mi mensaje
 1 files changed, 1 insertions(+), 0 deletions(-)
 create mode 100644 prueba
```

```
git init
echo algo > prueba
git add prueba
git commit -m "mi mensaje"
```

```
[master (root-commit) 3fb8a8e] mi mensaje
 1 files changed, 1 insertions(+), 0 deletions(-)
 create mode 100644 prueba
```

```
git init
echo algo > prueba
git add prueba
git commit -m "mi mensaje"
```

```
[master (root-commit) 3fb8a8e] mi mensaje
 1 files changed, 1 insertions(+), 0 deletions(-)
 create mode 100644 prueba
```

```
git init
echo algo > prueba
git add prueba
git commit -m "mi mensaje"
```

```
[master (root-commit) 3fb8a8e] mi mensaje
 1 files changed, 1 insertions(+), 0 deletions(-)
 create mode 100644 prueba
```

```
git init
echo algo > prueba
git add prueba
git commit -m "mi mensaje"
```

```
[master (root-commit) 3fb8a8e] mi mensaje
1 files changed, 1 insertions(+), 0 deletions(-)
create mode 100644 prueba
```

```
echo otra cosa >> prueba
git commit -m "otra revisión" -a
```

```
git init  
echo algo > prueba  
git add prueba  
git commit -m "mi mensaje"
```

```
[master (root-commit) 3fb8a8e] mi mensaje  
1 files changed, 1 insertions(+), 0 deletions(-)  
create mode 100644 prueba
```

```
echo otra cosa >> prueba  
git commit -m "otra revisión" -a
```

```
git init
echo algo > prueba
git add prueba
git commit -m "mi mensaje"
```

```
[master (root-commit) 3fb8a8e] mi mensaje
 1 files changed, 1 insertions(+), 0 deletions(-)
 create mode 100644 prueba
```

```
echo otra cosa >> prueba
git commit -m "otra revisión" -a
```

```
[master db35c00] otra revisión
 1 files changed, 1 insertions(+), 0 deletions(-)
```

```
git init
echo algo > prueba
git add prueba
git commit -m "mi mensaje"
```

```
[master (root-commit) 3fb8a8e] mi mensaje
 1 files changed, 1 insertions(+), 0 deletions(-)
 create mode 100644 prueba
```

```
echo otra cosa >> prueba
git commit -m "otra revisión" -a
```

```
[master db35c00] otra revisión
 1 files changed, 1 insertions(+), 0 deletions(-)
```

```
git init
echo algo > prueba
git add prueba
git commit -m "mi mensaje"
```

```
[master (root-commit) 3fb8a8e] mi mensaje
 1 files changed, 1 insertions(+), 0 deletions(-)
 create mode 100644 prueba
```

```
echo otra cosa >> prueba
git commit -m "otra revisión" -a
```

```
[master db35c00] otra revisión
 1 files changed, 1 insertions(+), 0 deletions(-)
```

```
git init  
echo algo > prueba  
git add prueba  
git commit -m "mi mensaje"
```

```
[master (root-commit) 3fb8a8e] mi mensaje  
1 files changed, 1 insertions(+), 0 deletions(-)  
create mode 100644 prueba
```

```
echo otra cosa >> prueba  
git commit -m "otra revisión" -a
```

```
[master db35c00] otra revisión  
1 files changed, 1 insertions(+), 0 deletions(-)
```

```
git init
echo algo > prueba
git add prueba
git commit -m "mi mensaje"
```

```
[master (root-commit) 3fb8a8e] mi mensaje
1 files changed, 1 insertions(+), 0 deletions(-)
create mode 100644 prueba
```

```
echo otra cosa >> prueba
git commit -m "otra revisión" -a
```

```
[master db35c00] otra revisión
1 files changed, 1 insertions(+), 0 deletions(-)
```

```
git init
echo algo > prueba
git add prueba
git commit -m "mi mensaje"
```

```
[master (root-commit) 3fb8a8e] mi mensaje
 1 files changed, 1 insertions(+), 0 deletions(-)
 create mode 100644 prueba
```

```
echo otra cosa >> prueba
git commit -m "otra revisión" -a
```

```
[master db35c00] otra revisión
 1 files changed, 1 insertions(+), 0 deletions(-)
```

# ID de revisiones

# ID de revisiones

git init

# ID de revisiones

`git init`

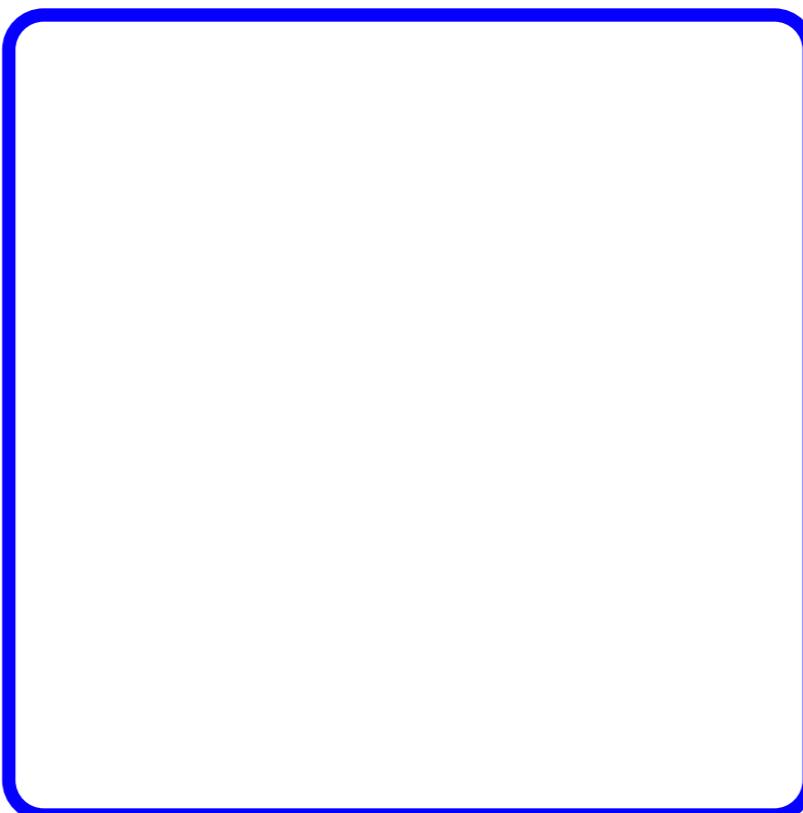
Initialized empty Git repository in \$TRABAJO/.git

# ID de revisiones

git init

Initialized empty Git repository in \$TRABAJO/.git

\$TRABAJO



# ID de revisiones

git init

Initialized empty Git repository in \$TRABAJO/.git

\$TRABAJO

.git

# ID de revisiones

git init

Initialized empty Git repository in \$TRABAJO/.git

\$TRABAJO

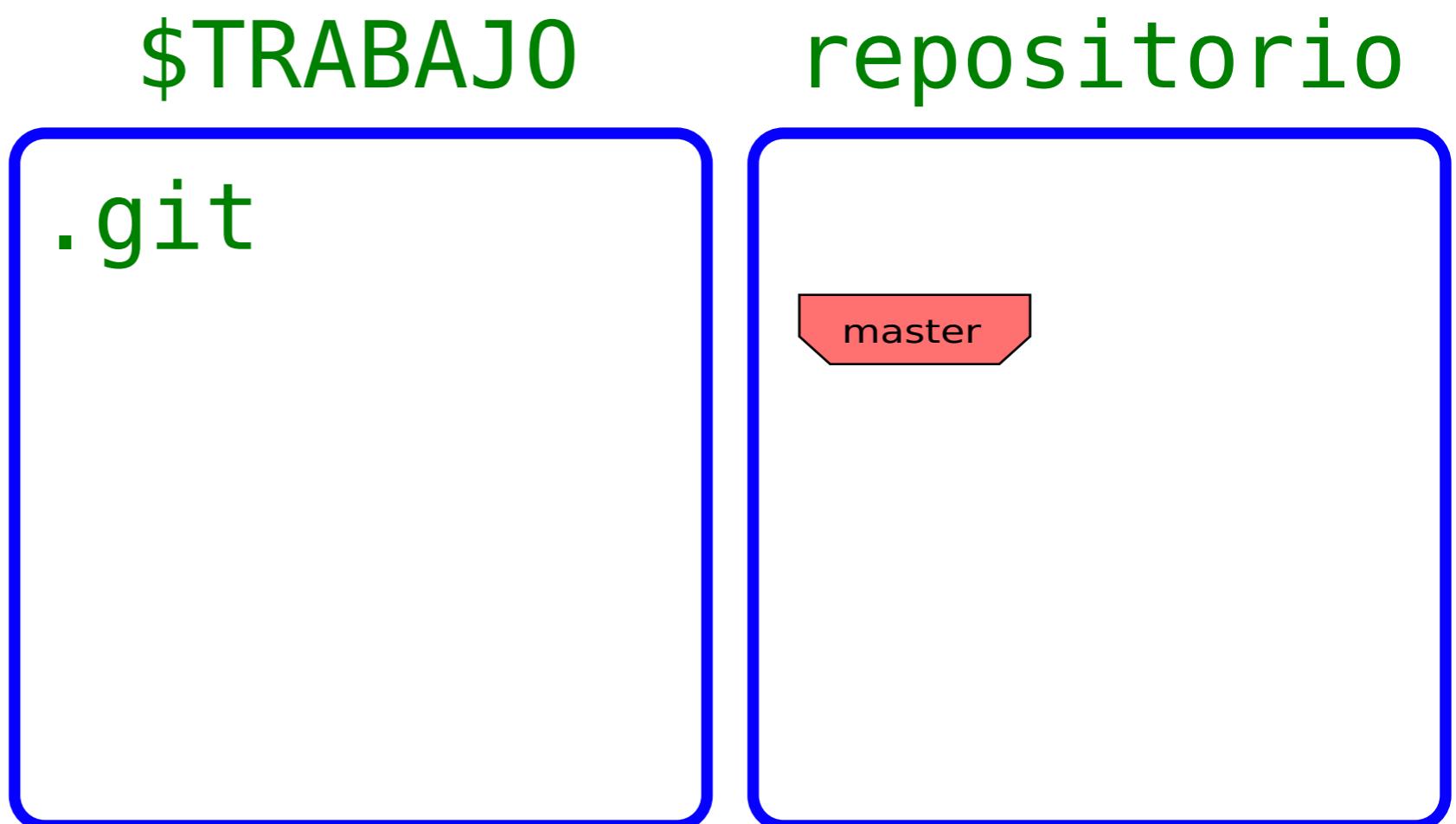
.git

repositorio

# ID de revisiones

git init

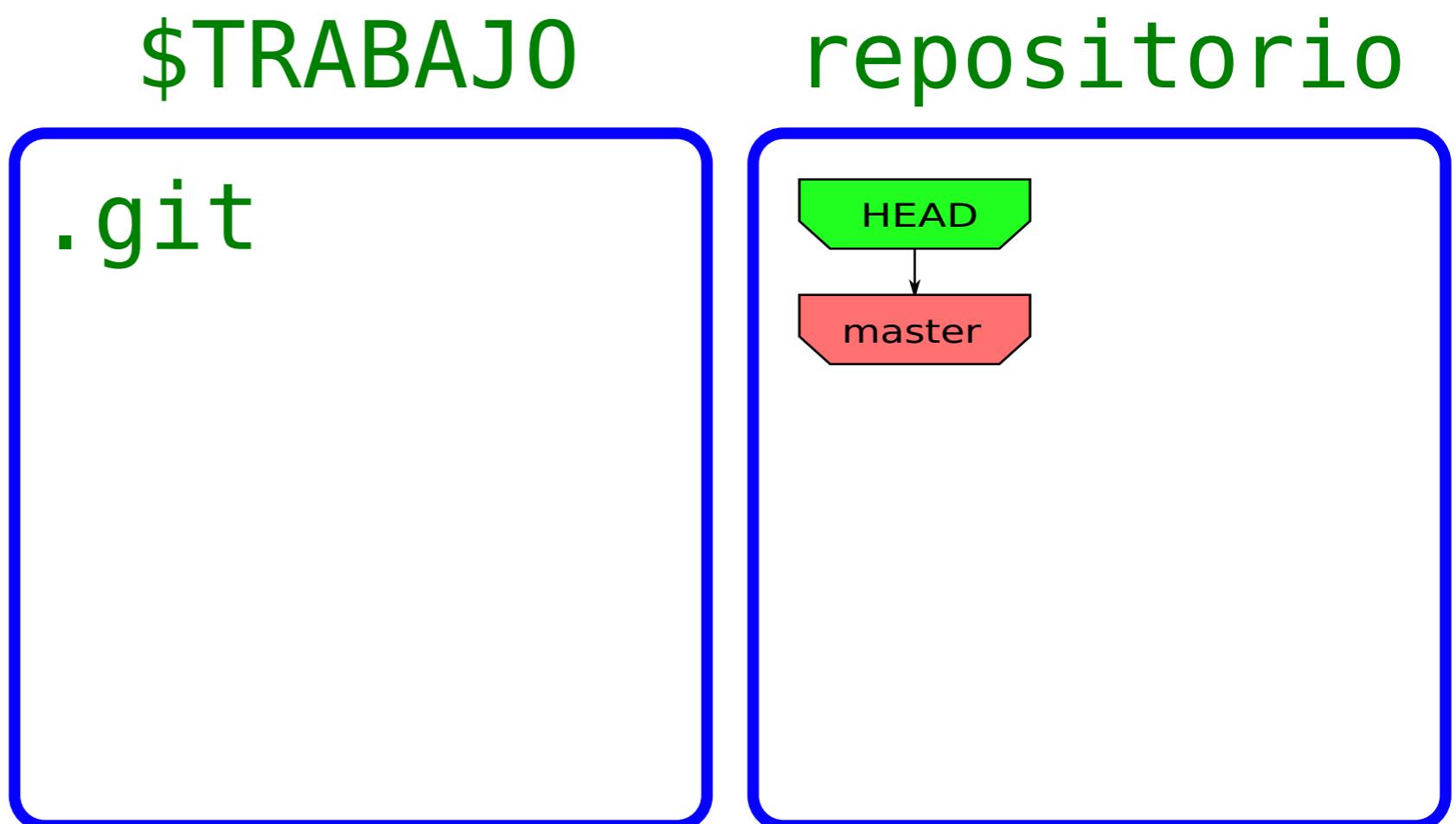
Initialized empty Git repository in \$TRABAJO/.git



# ID de revisiones

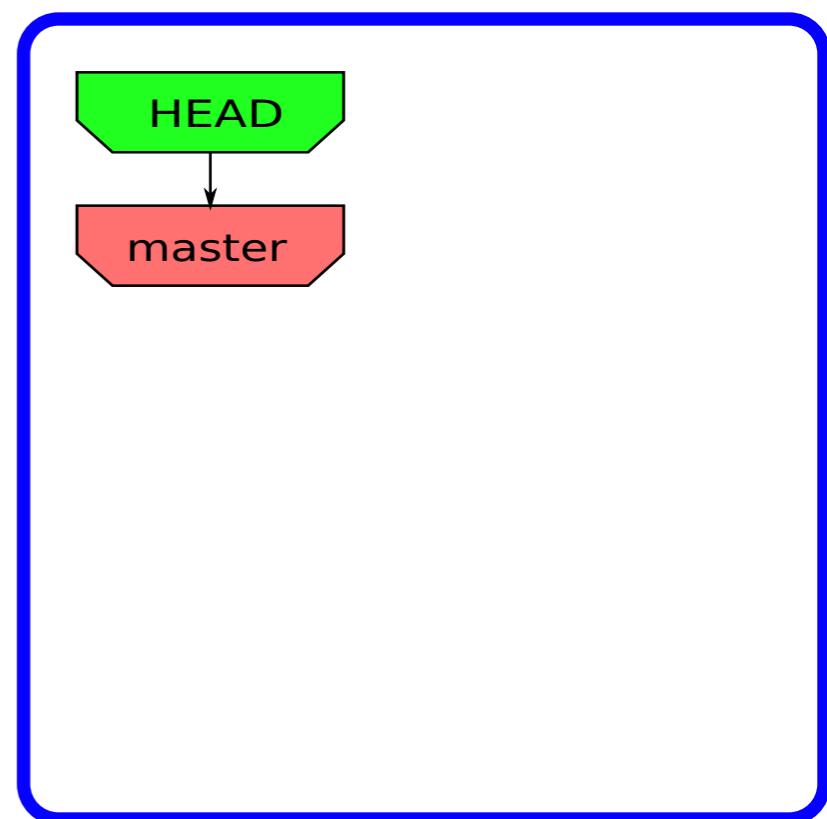
git init

Initialized empty Git repository in \$TRABAJO/.git



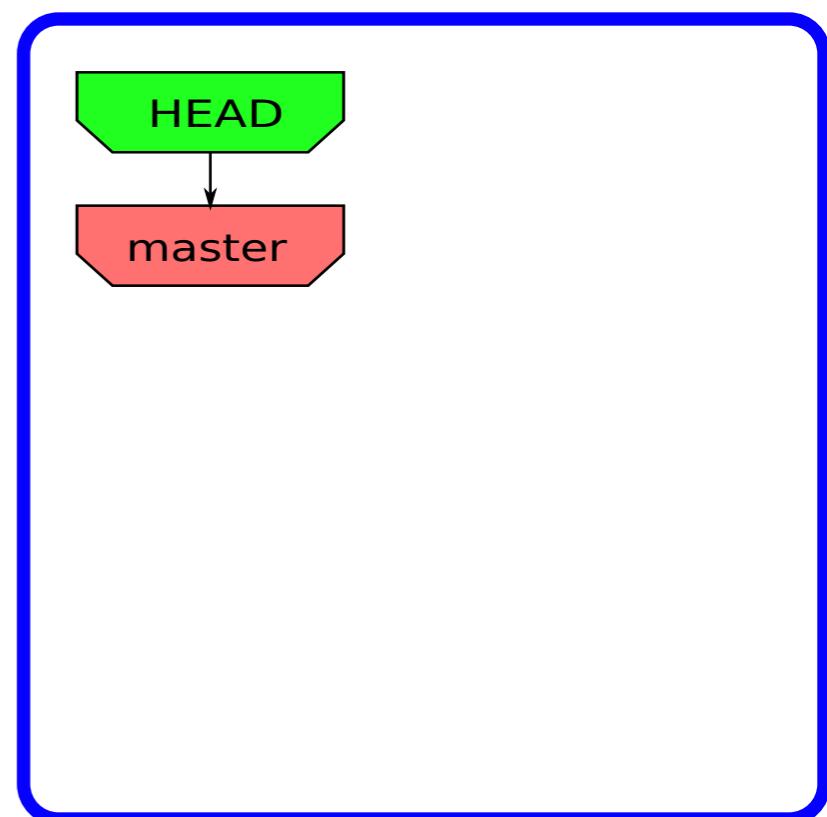
# ID de revisiones

git init



# ID de revisiones

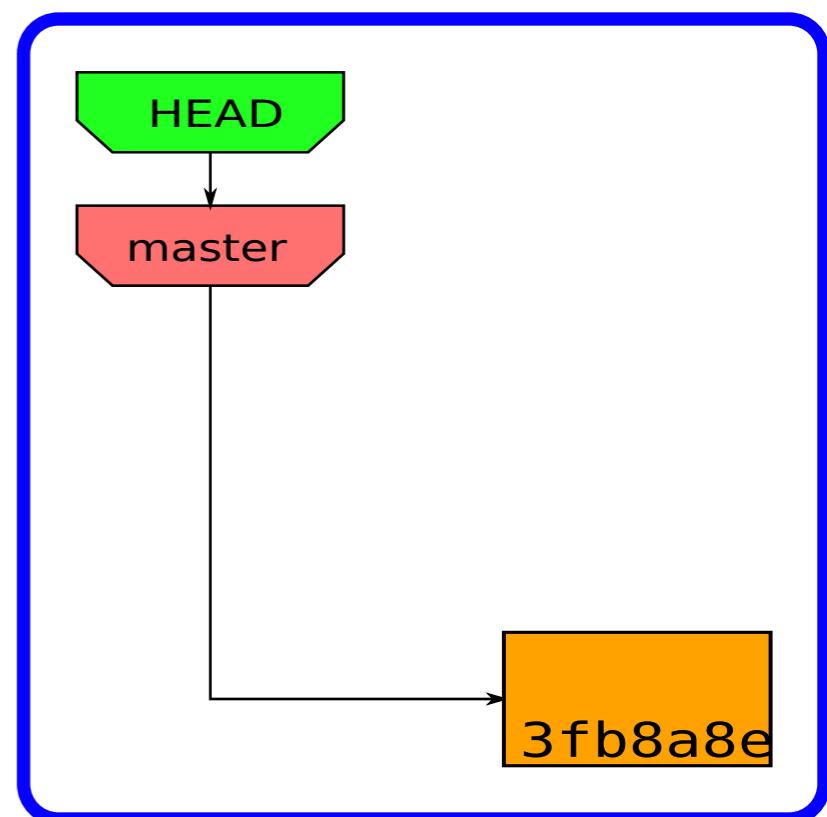
```
git init  
git commit -a -m "primero"
```



# ID de revisiones

```
git init  
git commit -a -m "primero"
```

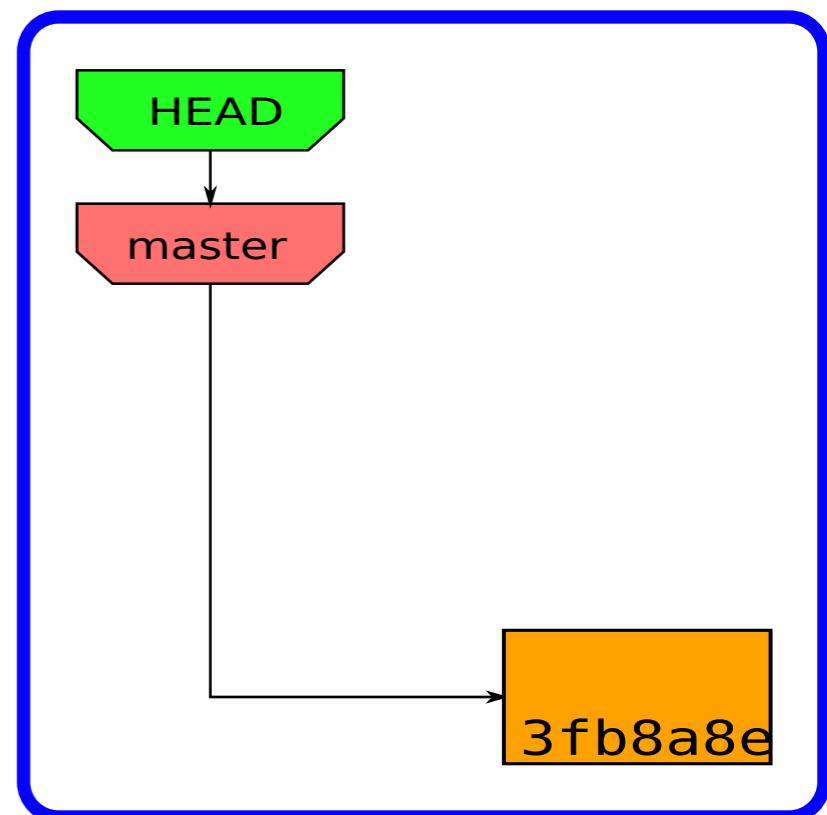
```
[master (root-commit) 3fb8a8e] primero  
1 files changed, 1 insertions(+), 0 deletions(-)  
create mode 100644 prueba
```



# ID de revisiones

```
git init  
git commit -a -m "primero"
```

```
[master (root-commit) 3fb8a8e] primero  
1 files changed, 1 insertions(+), 0 deletions(-)  
create mode 100644 prueba
```

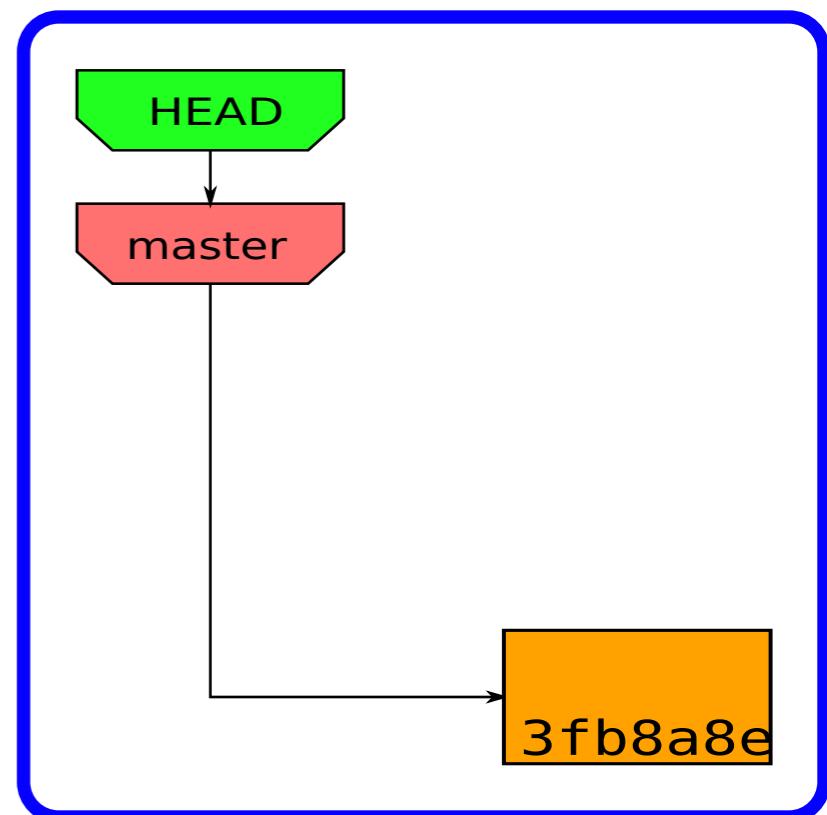


# ID de revisiones

```
git init  
git commit -a -m "primero"
```

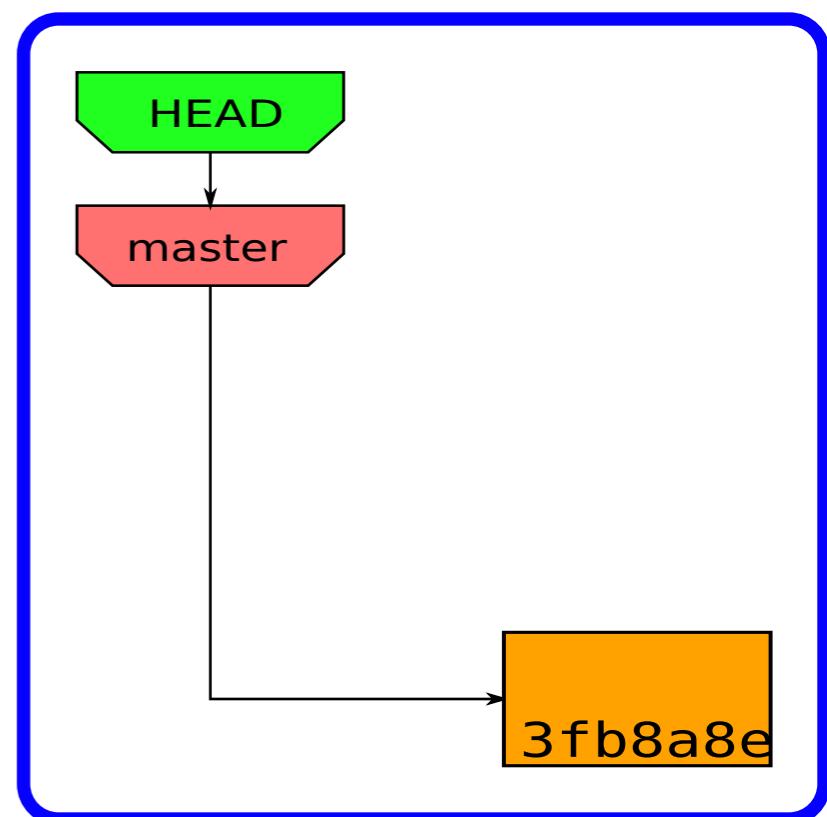
```
[master (root-commit) 3fb8a8e] primero  
1 files changed, 1 insertions(+), 0 deletions(-)  
create mode 100644 prueba
```

3fb8a8e74cde5324bf670856a8186d9e57942b20



# ID de revisiones

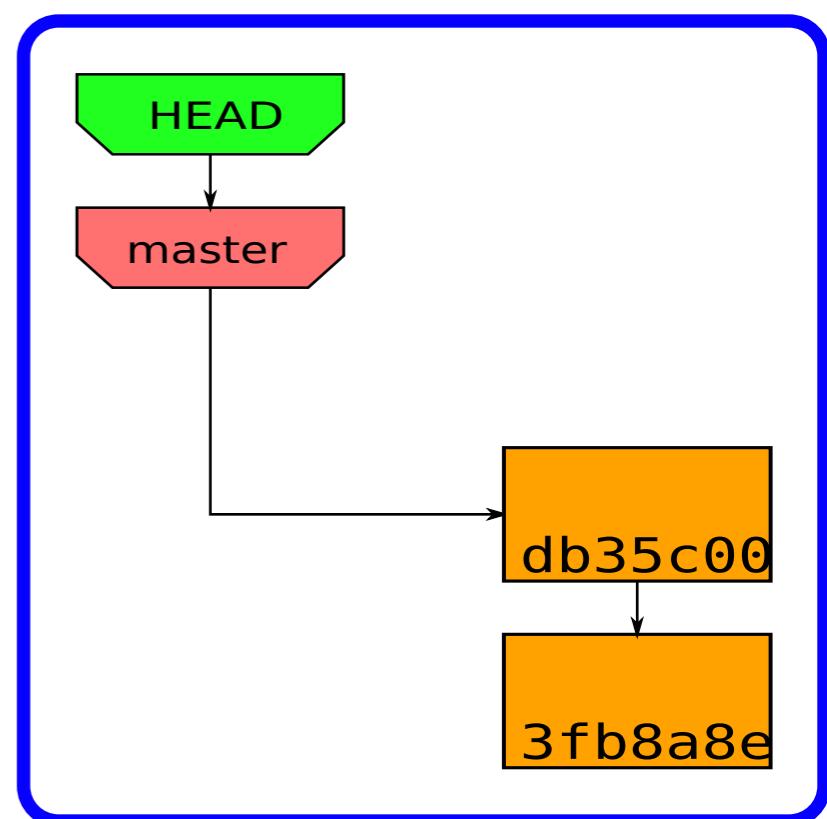
```
git init  
git commit -a -m "primero"  
git commit -a -m "segundo"
```



# ID de revisiones

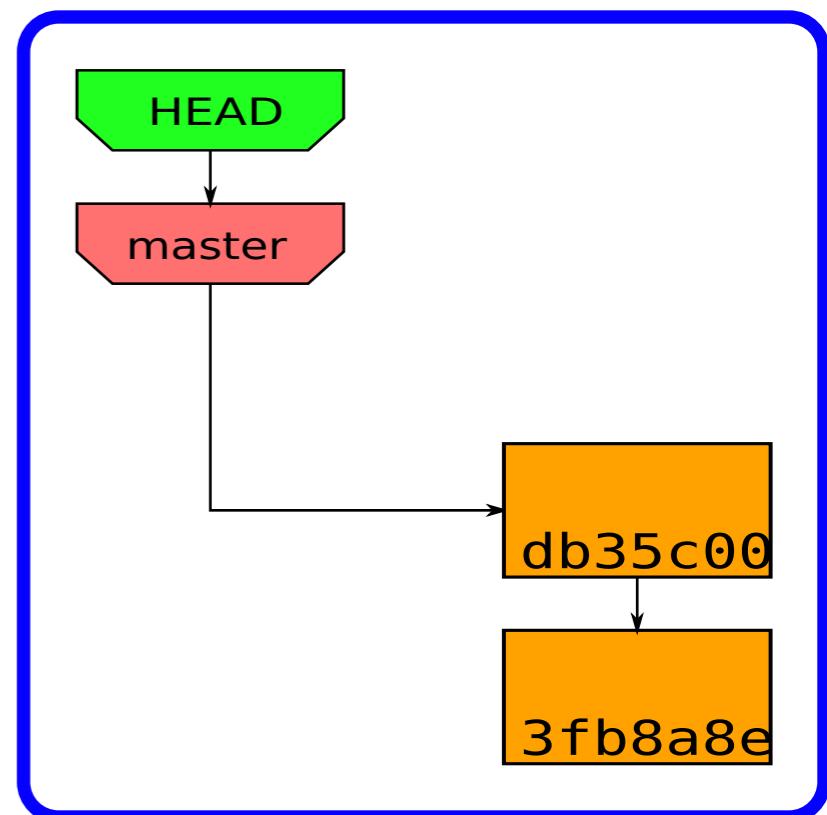
```
git init  
git commit -a -m "primero"  
git commit -a -m "segundo"
```

```
[master db35c00] segundo  
1 files changed, 1 insertions(+), 0 deletions(-)
```



# ID de revisiones

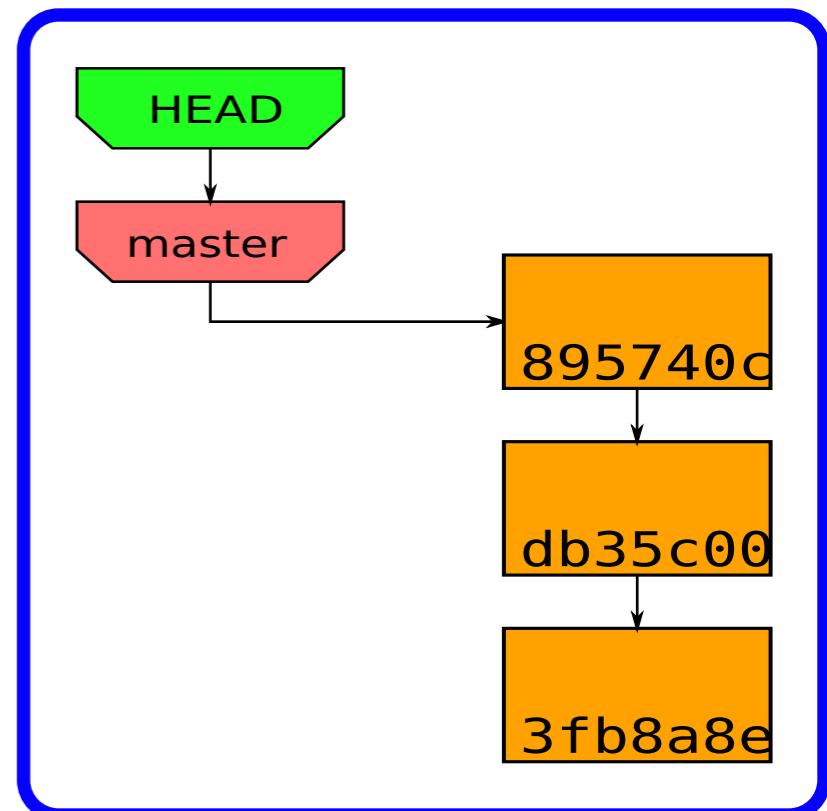
```
git init  
git commit -a -m "primero"  
git commit -a -m "segundo"  
git commit -a -m "tercero"
```



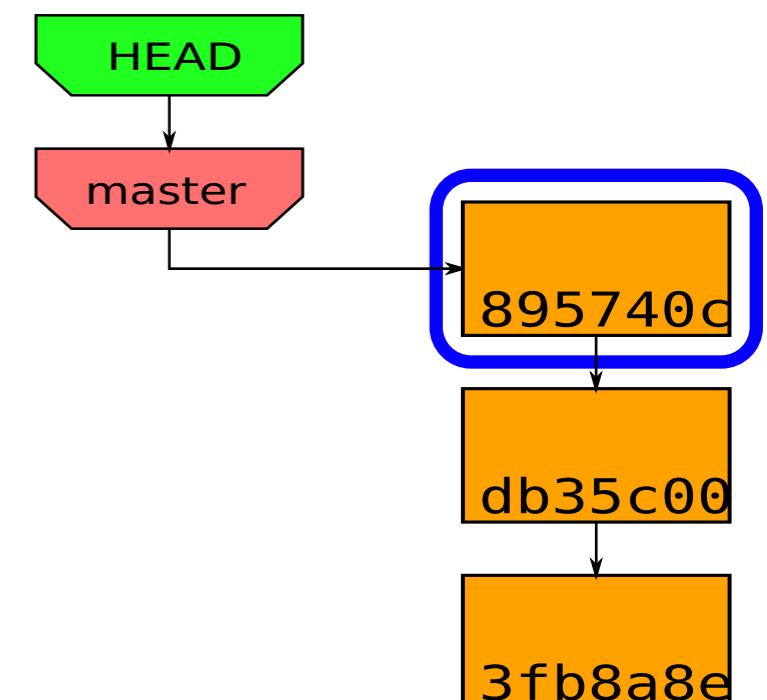
# ID de revisiones

```
git init  
git commit -a -m "primero"  
git commit -a -m "segundo"  
git commit -a -m "tercero"
```

```
[master 895740c] tercero  
1 files changed, 1 insertions(+), 0 deletions(-)
```

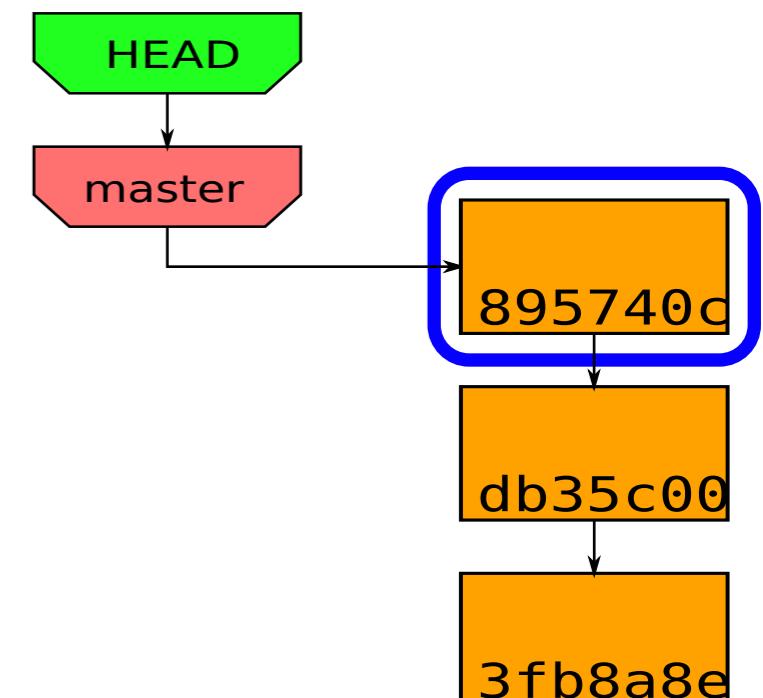


# Especificar revisiones por identificador SHA1



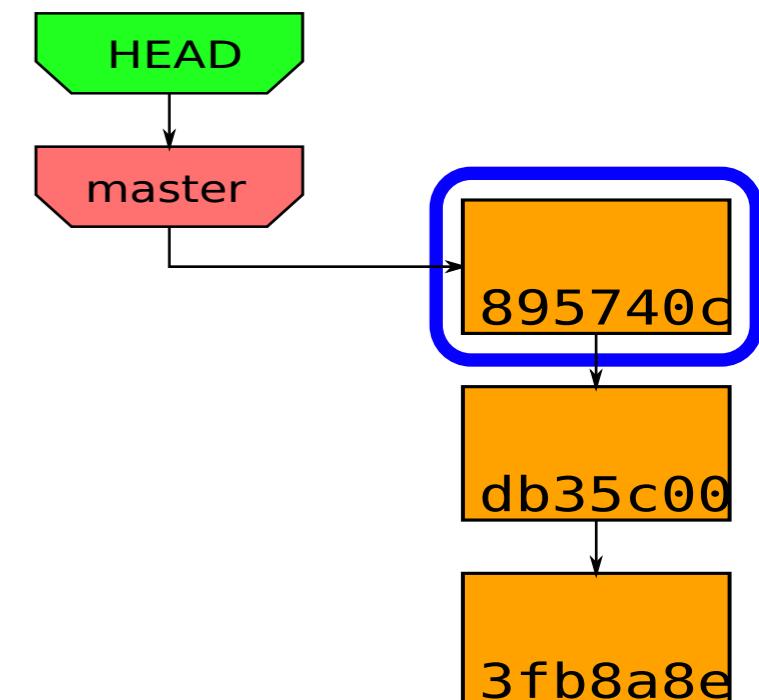
# Especificar revisiones por identificador SHA1

895740c342fdb03ce26b6417051ab6f2188a6d0



# Especificar revisiones por identificador SHA1

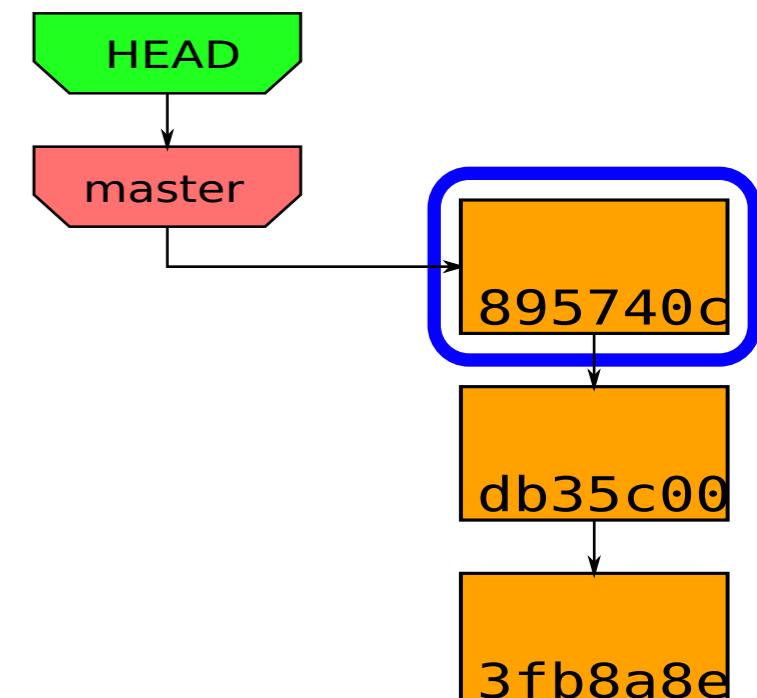
895740c342fdb03ce26b6417051ab6f2188a6d0



# Especificar revisiones por identificador SHA1

895740c342fdb03ce26b6417051ab6f2188a6d0

895740c

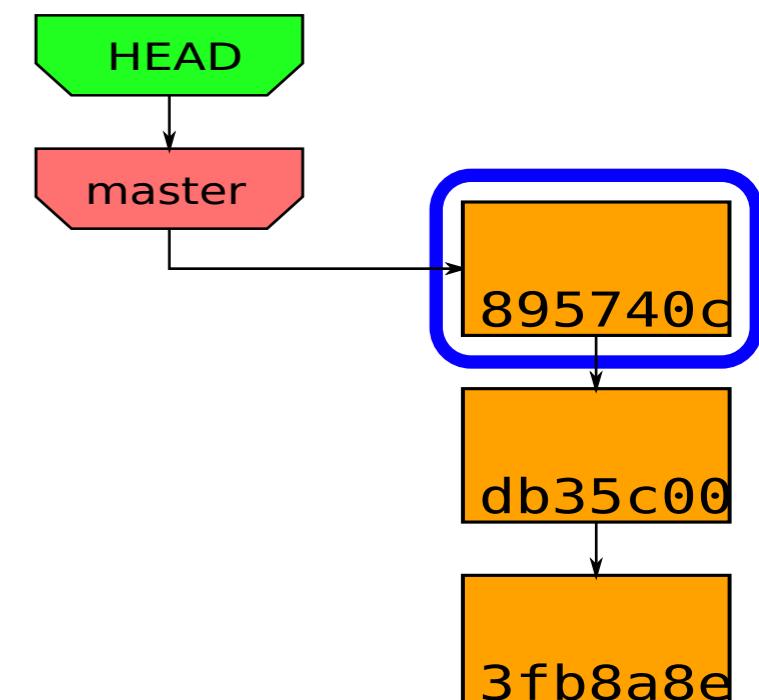


# Especificar revisiones por identificador SHA1

895740c342fdb03ce26b6417051ab6f2188a6d0

895740c

simbólicamente

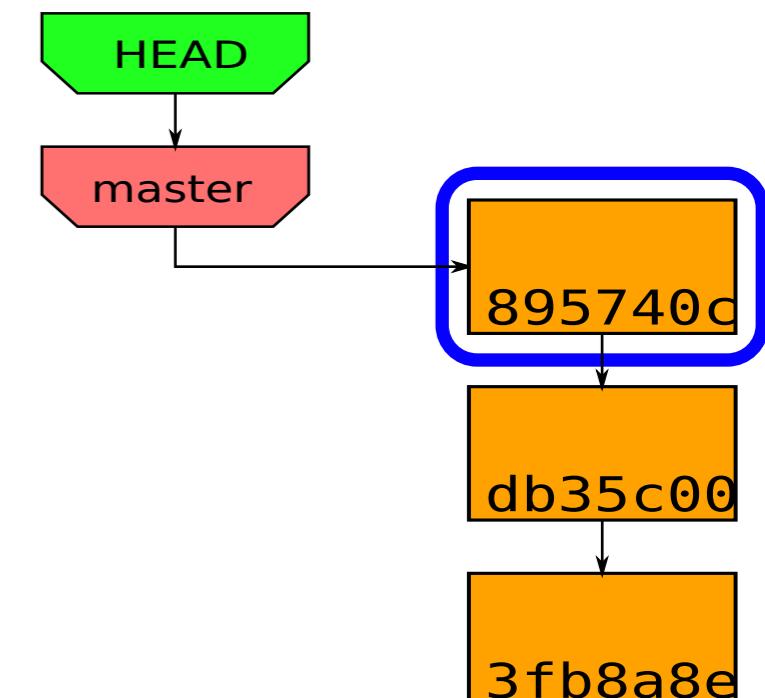


# Especificar revisiones por identificador SHA1

895740c342fdb03ce26b6417051ab6f2188a6d0

895740c

simbólicamente  
**master**



# Especificar revisiones por identificador SHA1

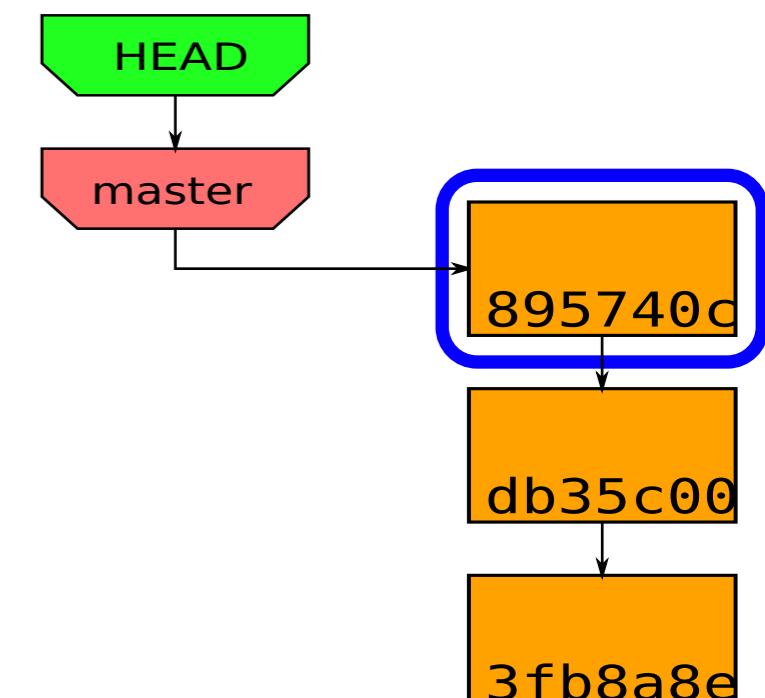
895740c342fdb03ce26b6417051ab6f2188a6d0

895740c

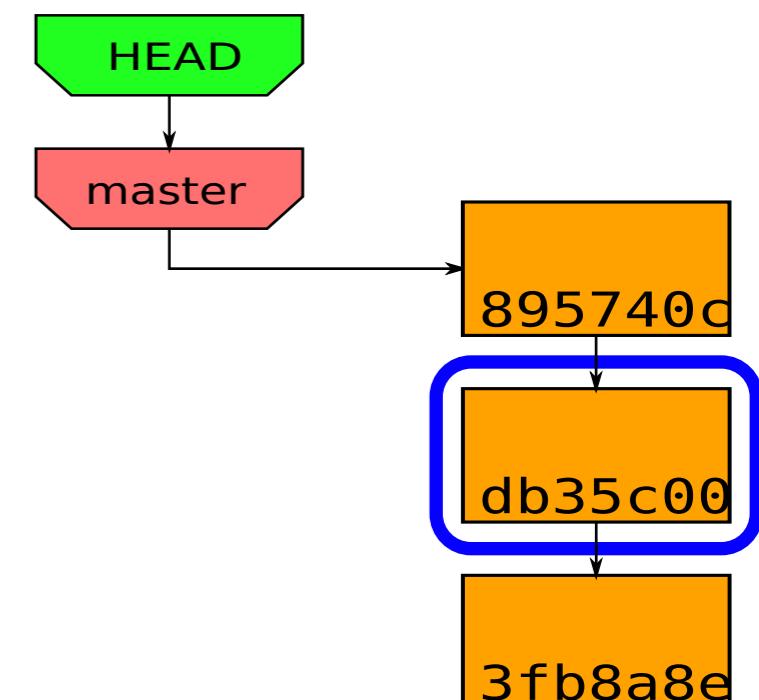
simbólicamente

master

HEAD

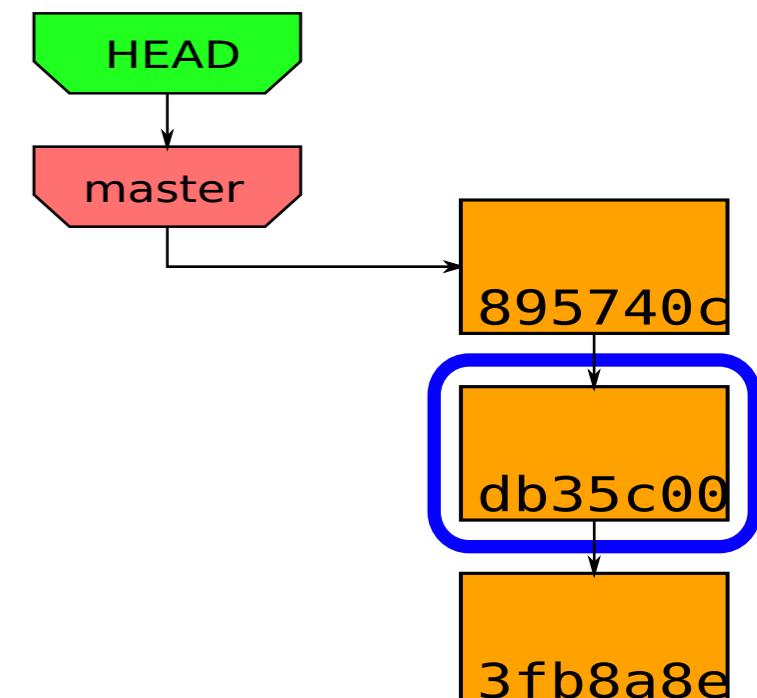


# Especificar revisiones por identificador SHA1



# Especificar revisiones por identificador SHA1

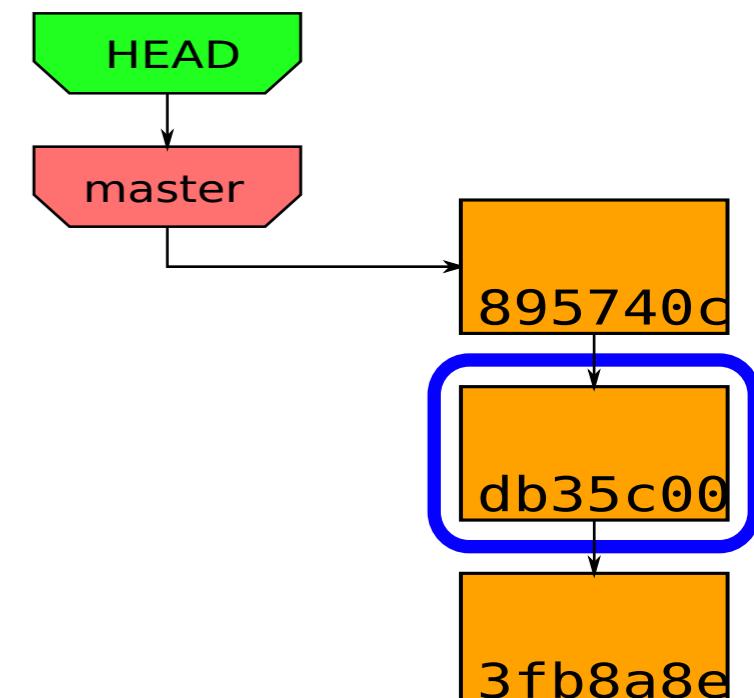
db35c00c63a7e98c4a89e180d317c9fb08e40f4b



# Especificar revisiones por identificador SHA1

db35c00c63a7e98c4a89e180d317c9fb08e40f4b

db35c00

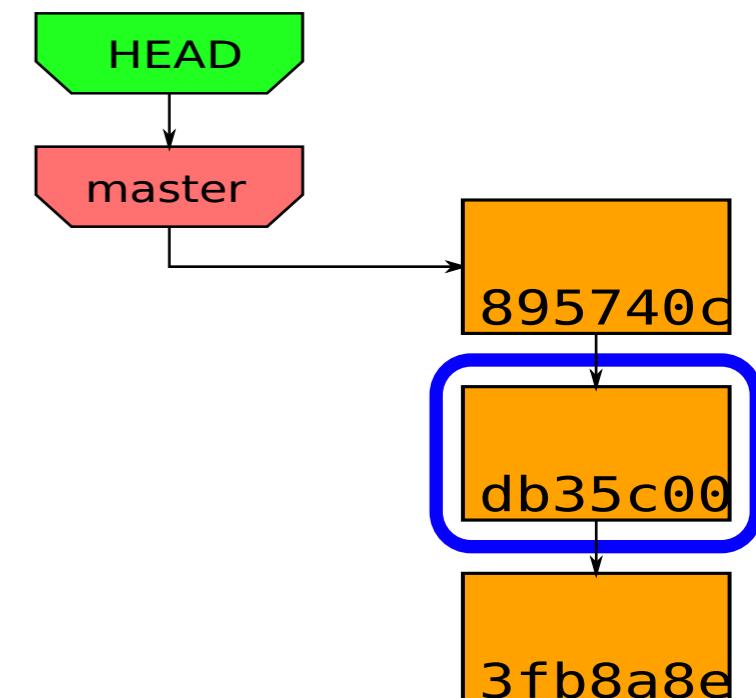


# Especificar revisiones por identificador SHA1

db35c00c63a7e98c4a89e180d317c9fb08e40f4b

db35c00

simbólicamente

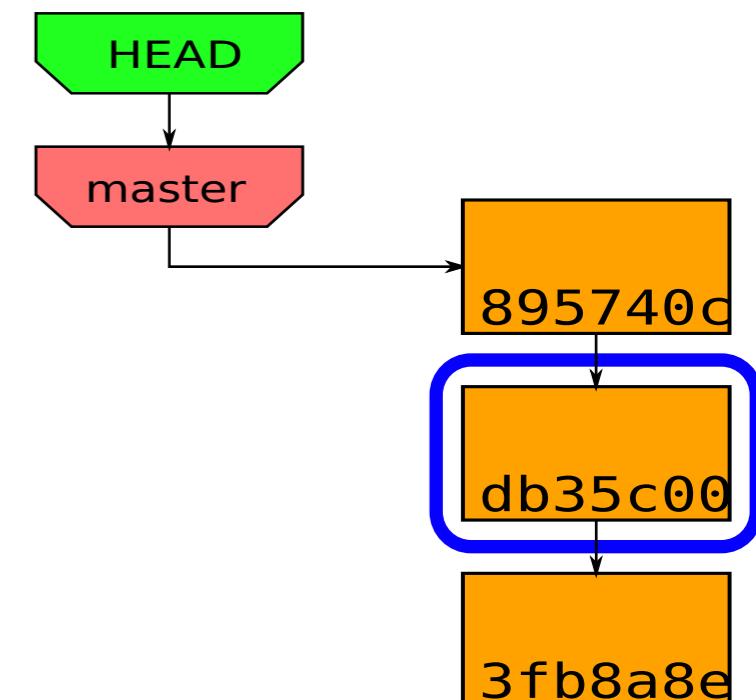


# Especificar revisiones por identificador SHA1

db35c00c63a7e98c4a89e180d317c9fb08e40f4b

db35c00

simbólicamente  
**master<sup>^</sup>**



# Especificar revisiones por identificador SHA1

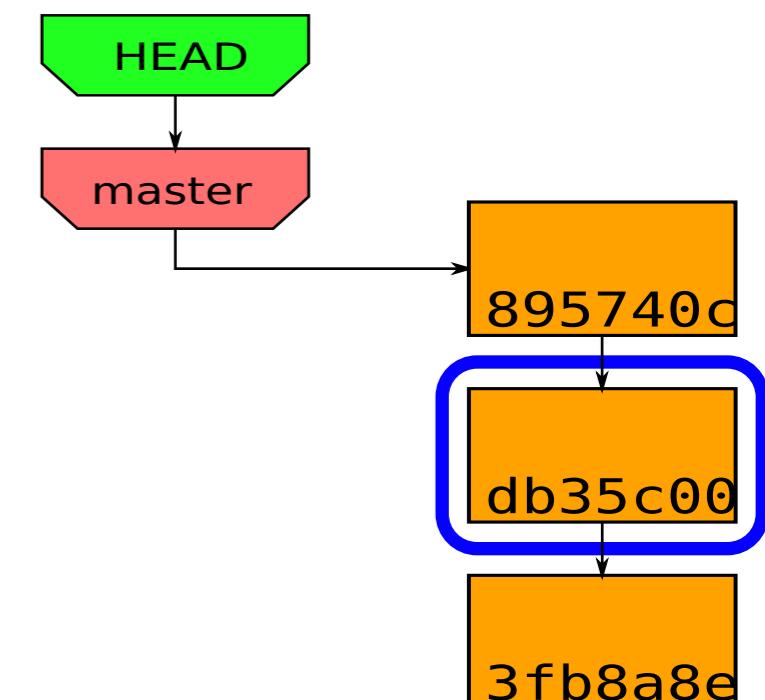
db35c00c63a7e98c4a89e180d317c9fb08e40f4b

db35c00

simbólicamente

master<sup>^</sup>

HEAD<sup>^</sup>



# Especificar revisiones por identificador SHA1

db35c00c63a7e98c4a89e180d317c9fb08e40f4b

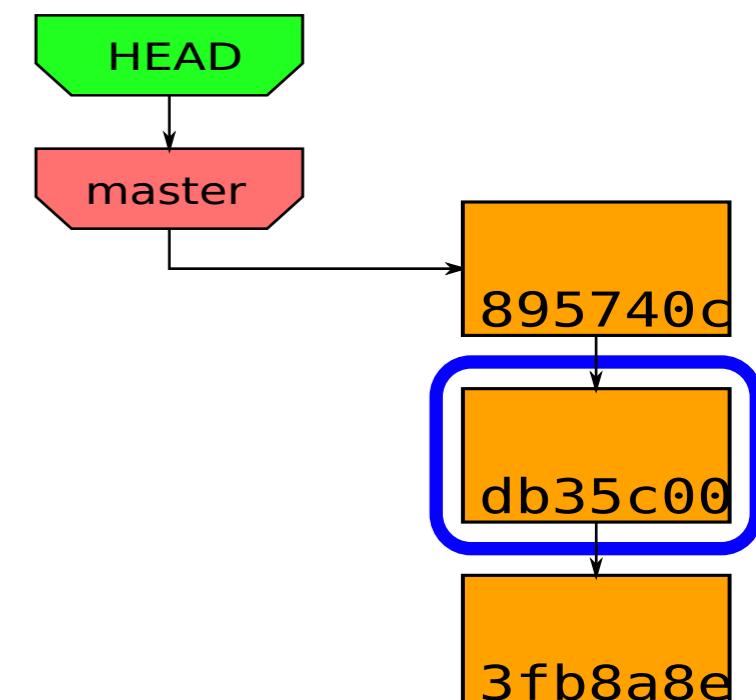
db35c00

**895740c<sup>^</sup>**

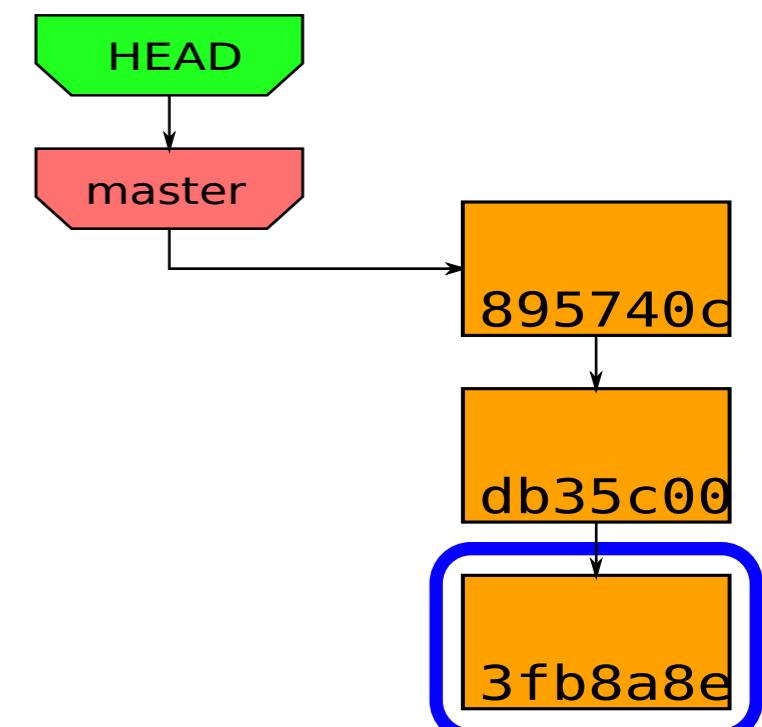
simbólicamente

**master<sup>^</sup>**

**HEAD<sup>^</sup>**

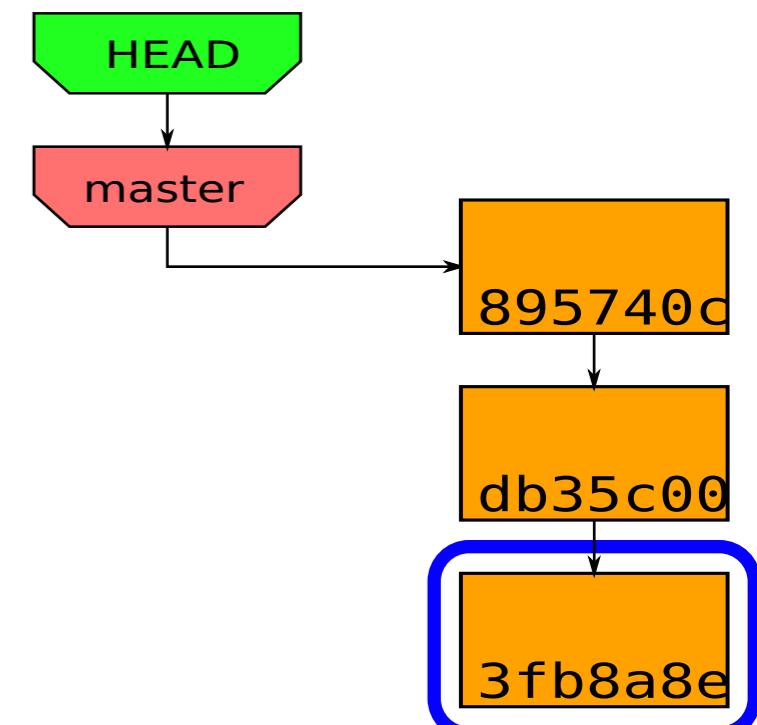


# Especificar revisiones por identificador SHA1



# Especificar revisiones por identificador SHA1

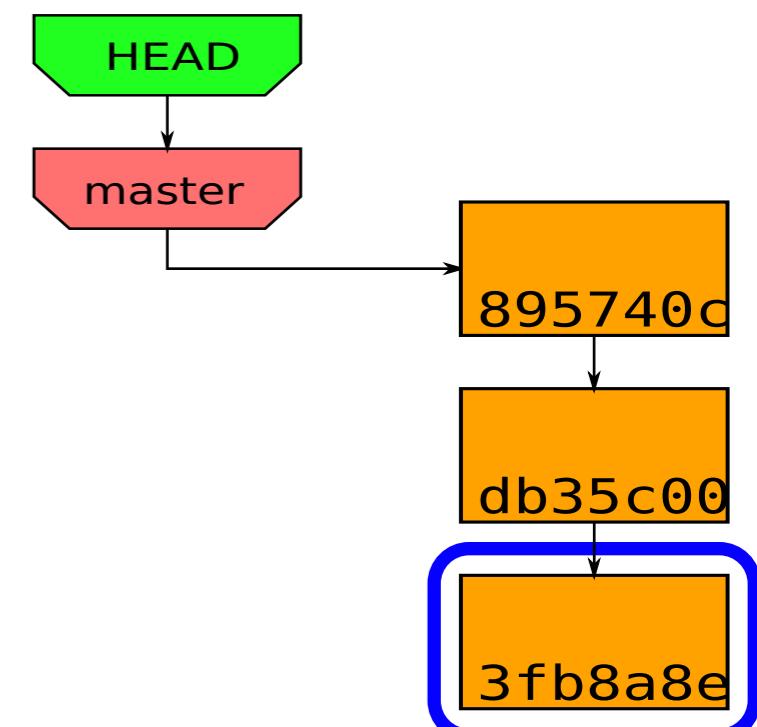
3fb8a8e74cde5324bf670856a8186d9e57942b20



# Especificar revisiones por identificador SHA1

3fb8a8e74cde5324bf670856a8186d9e57942b20

3fb8a8e

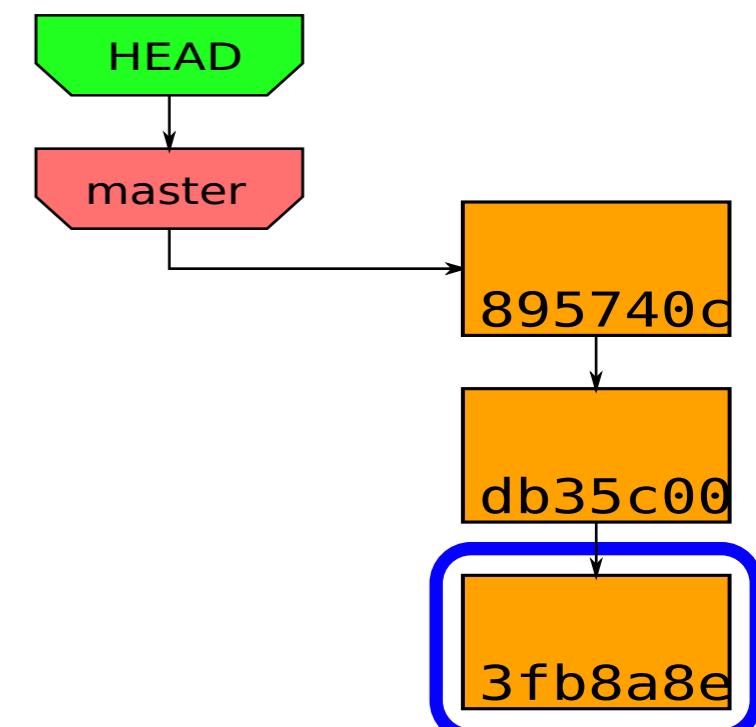


# Especificar revisiones por identificador SHA1

3fb8a8e74cde5324bf670856a8186d9e57942b20

3fb8a8e

simbólicamente



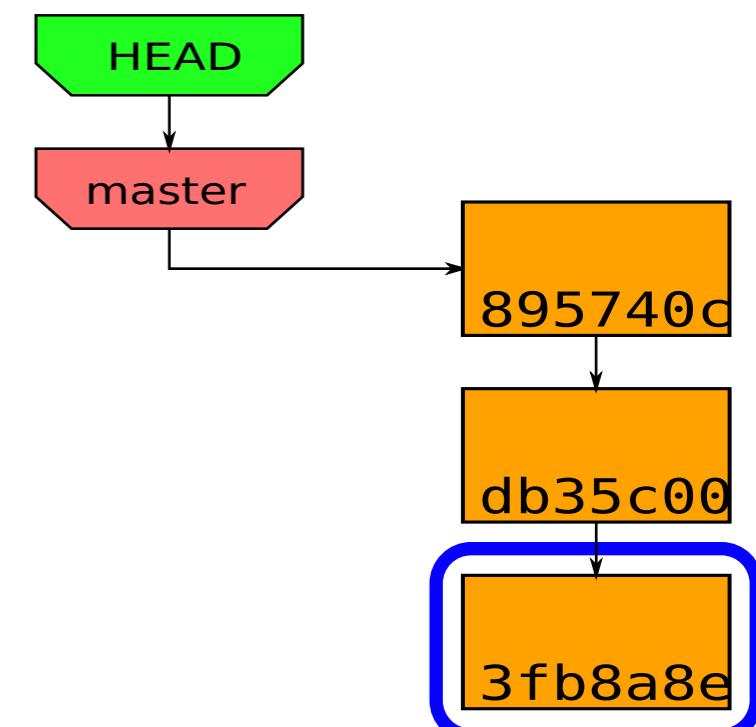
# Especificar revisiones por identificador SHA1

3fb8a8e74cde5324bf670856a8186d9e57942b20

3fb8a8e

simbólicamente

master<sup>^^</sup>



# Especificar revisiones por identificador SHA1

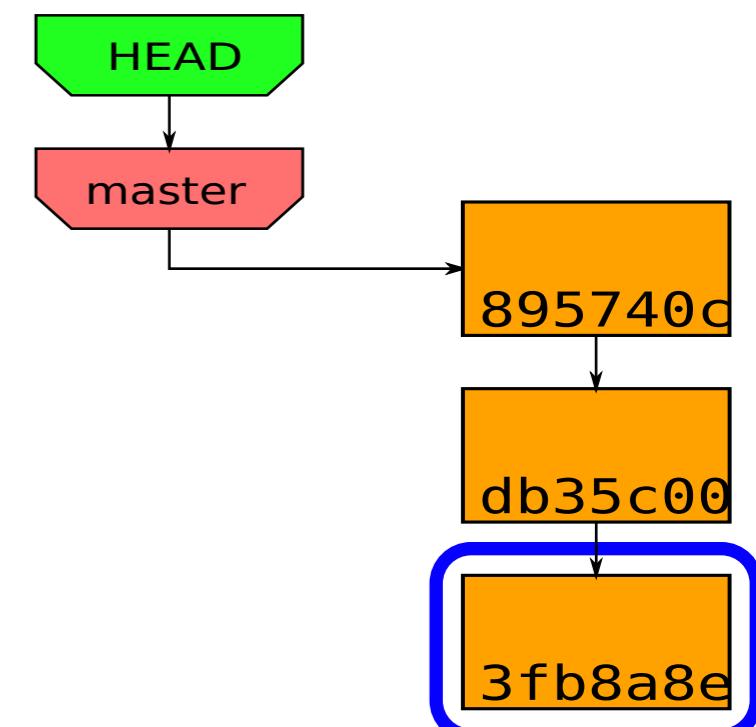
3fb8a8e74cde5324bf670856a8186d9e57942b20

3fb8a8e

simbólicamente

master<sup>^^</sup>

HEAD~2



# Especificar revisiones por identificador SHA1

3fb8a8e74cde5324bf670856a8186d9e57942b20

3fb8a8e

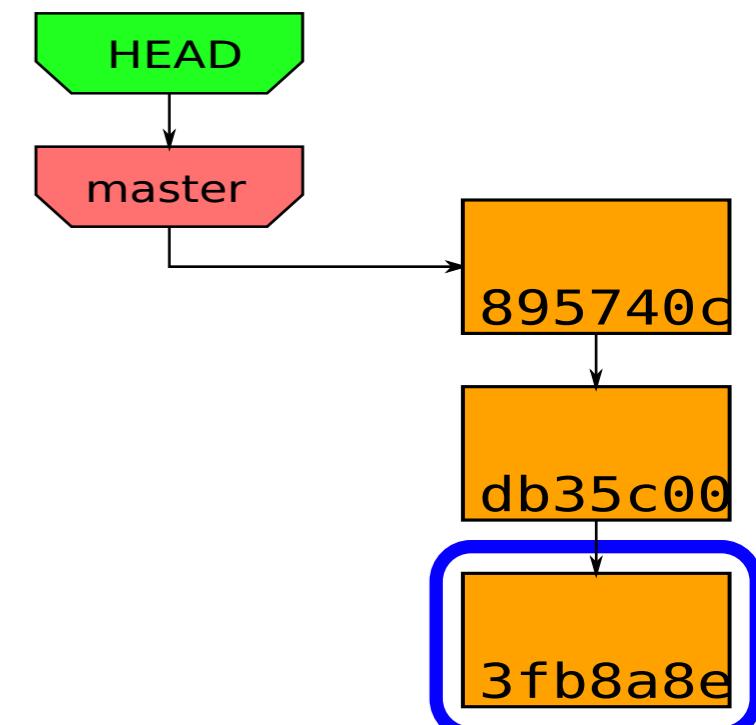
db35c00^

895740c~2

simbólicamente

master^^

HEAD~2



# Especificar revisiones por identificador SHA1

3fb8a8e74cde5324bf670856a8186d9e57942b20

3fb8a8e

db35c00^

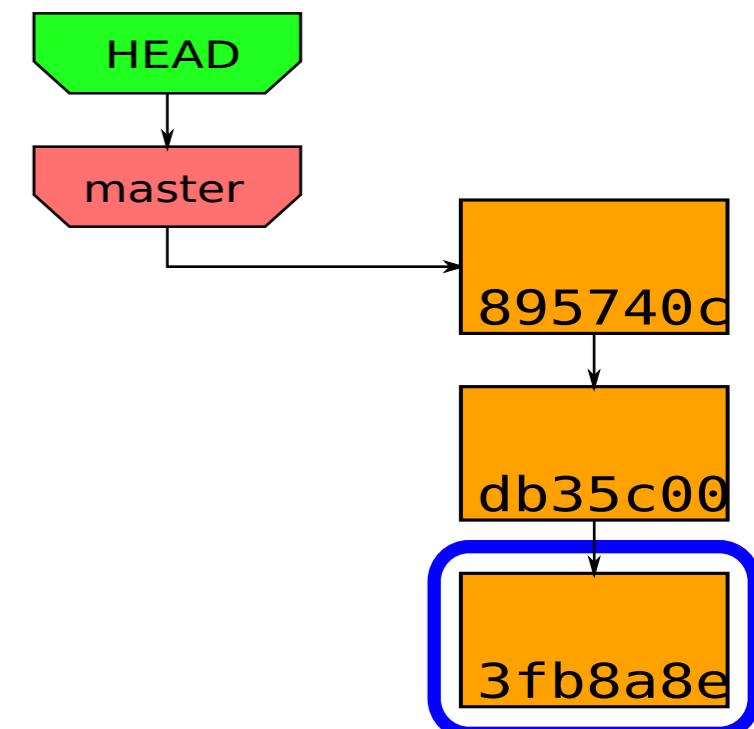
895740c~2

simbólicamente

master^^

HEAD~2

(más en *git-rev-parse(1)*)

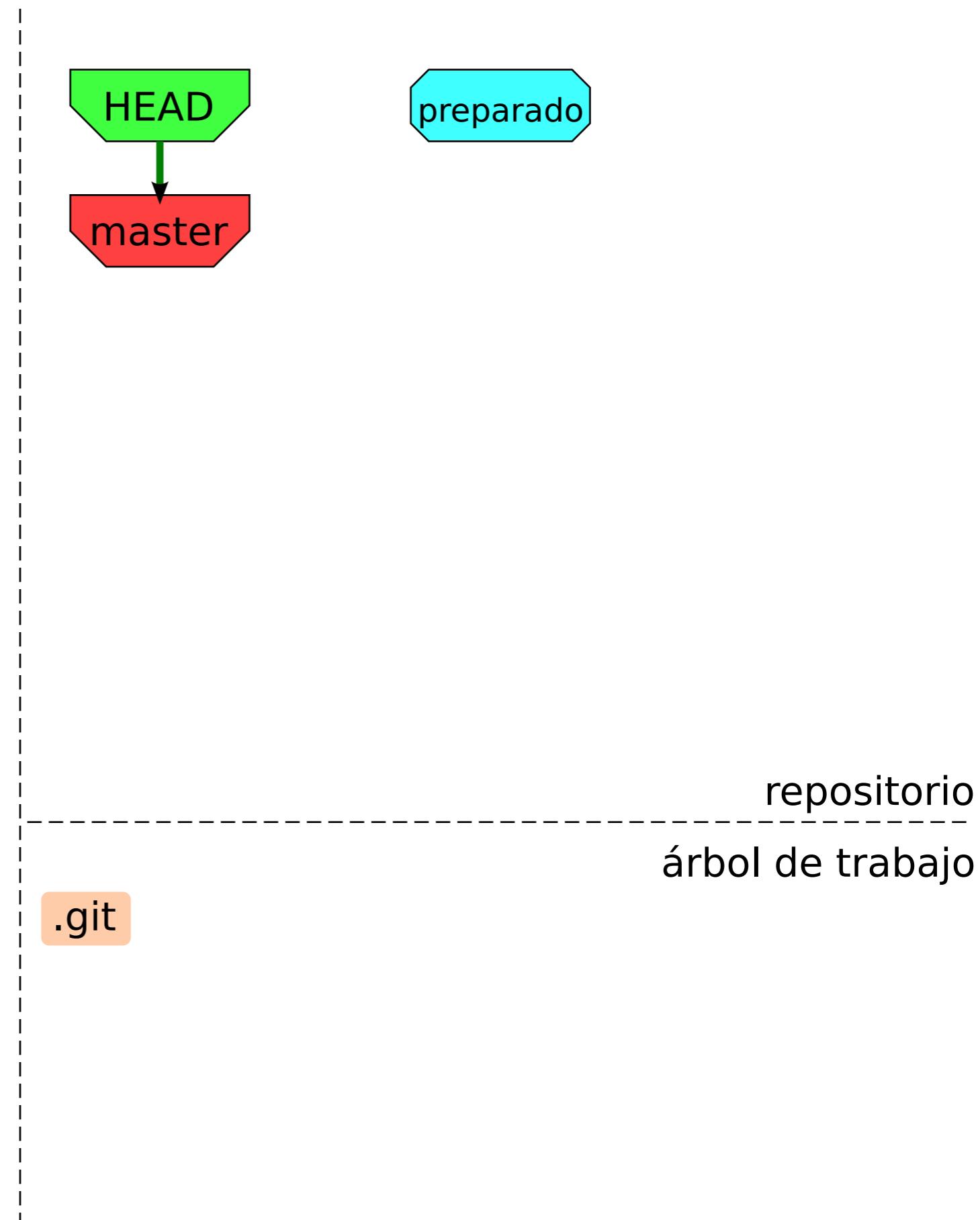


# Ejemplo de preparación y creación de una revisión

```
# mkdir proyecto  
# cd proyecto
```

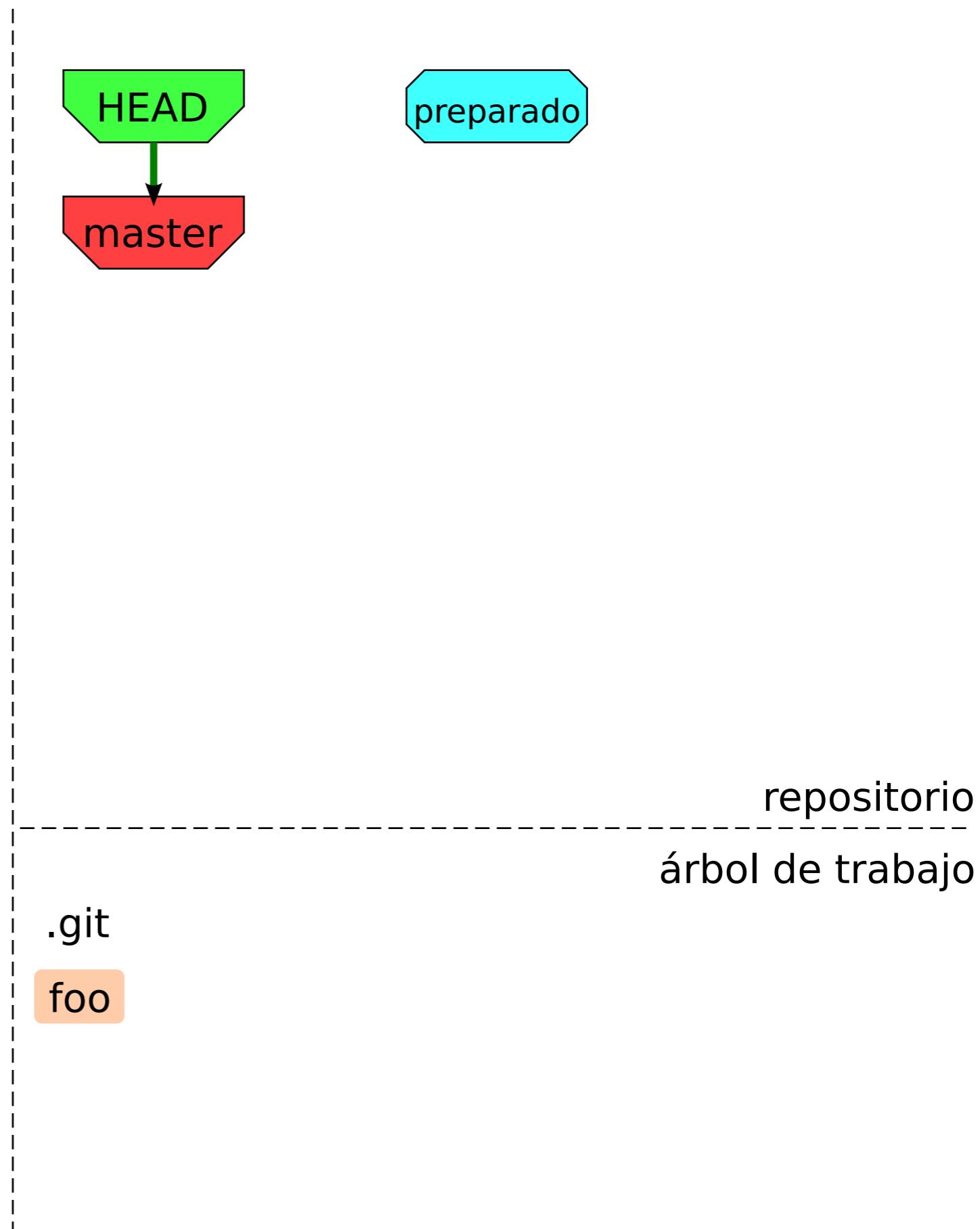
árbol de trabajo

```
# mkdir proyecto  
# cd proyecto  
# git init
```



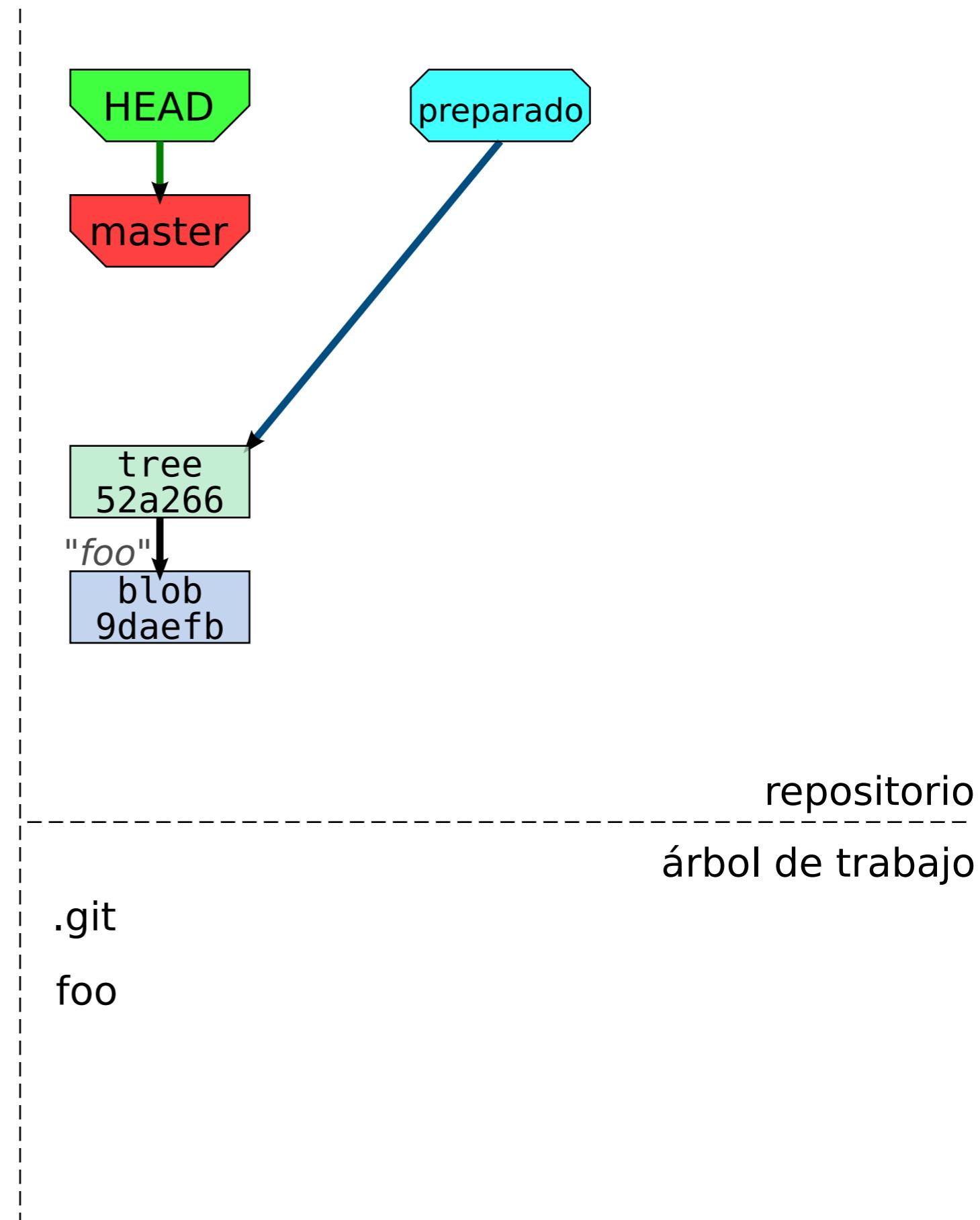
```
# mkdir proyecto  
# cd proyecto  
# git init
```

```
# echo foo > foo
```

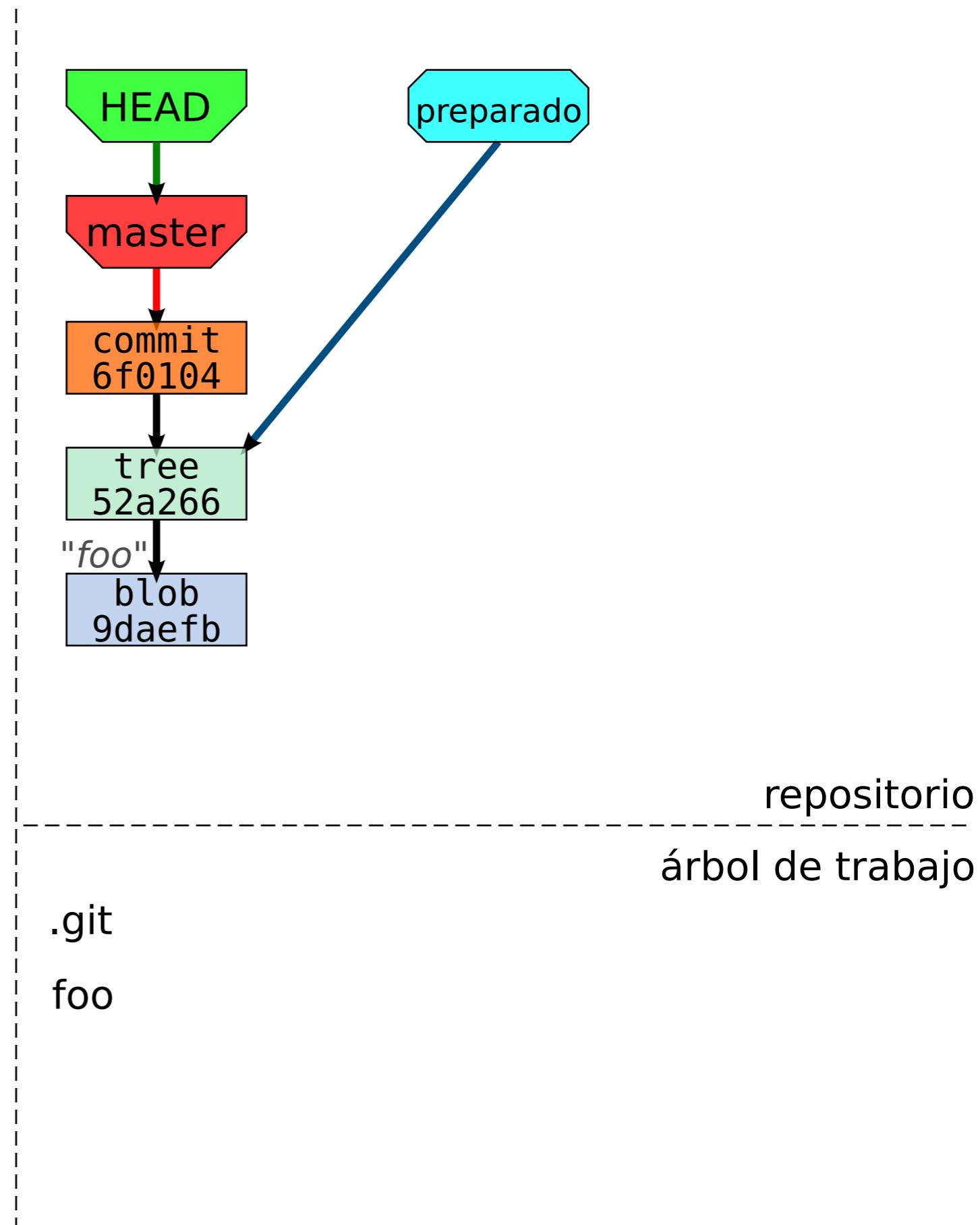


```
# mkdir proyecto  
# cd proyecto  
# git init
```

```
# echo foo > foo  
# git add foo
```



```
# mkdir proyecto  
# cd proyecto  
# git init  
  
# echo foo > foo  
# git add foo  
  
# git commit -m "foo"
```



```

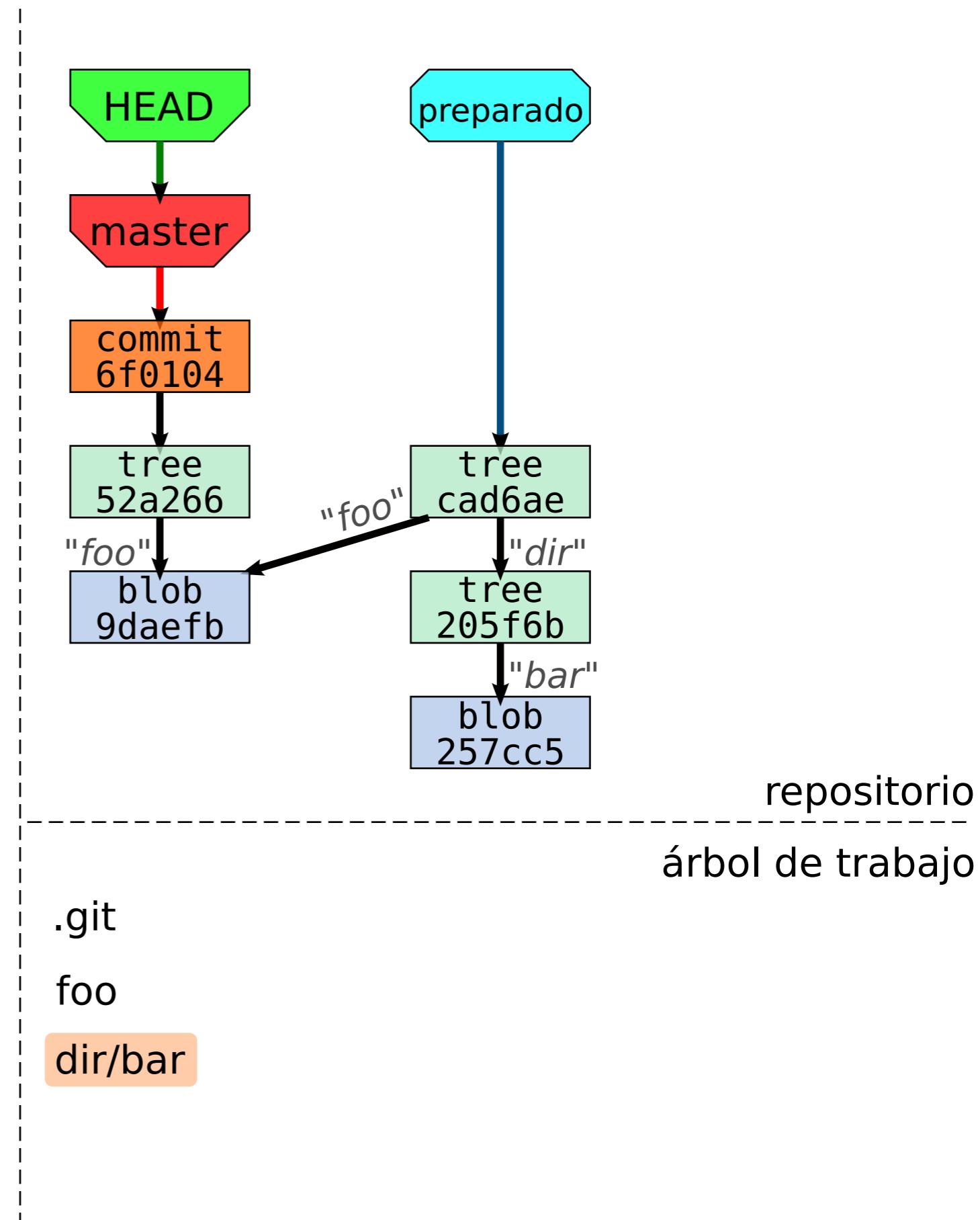
# mkdir proyecto
# cd proyecto
# git init

# echo foo > foo
# git add foo

# git commit -m "foo"

# mkdir dir
# echo bar > dir/bar
# git add dir/bar

```



```

# mkdir proyecto
# cd proyecto
# git init

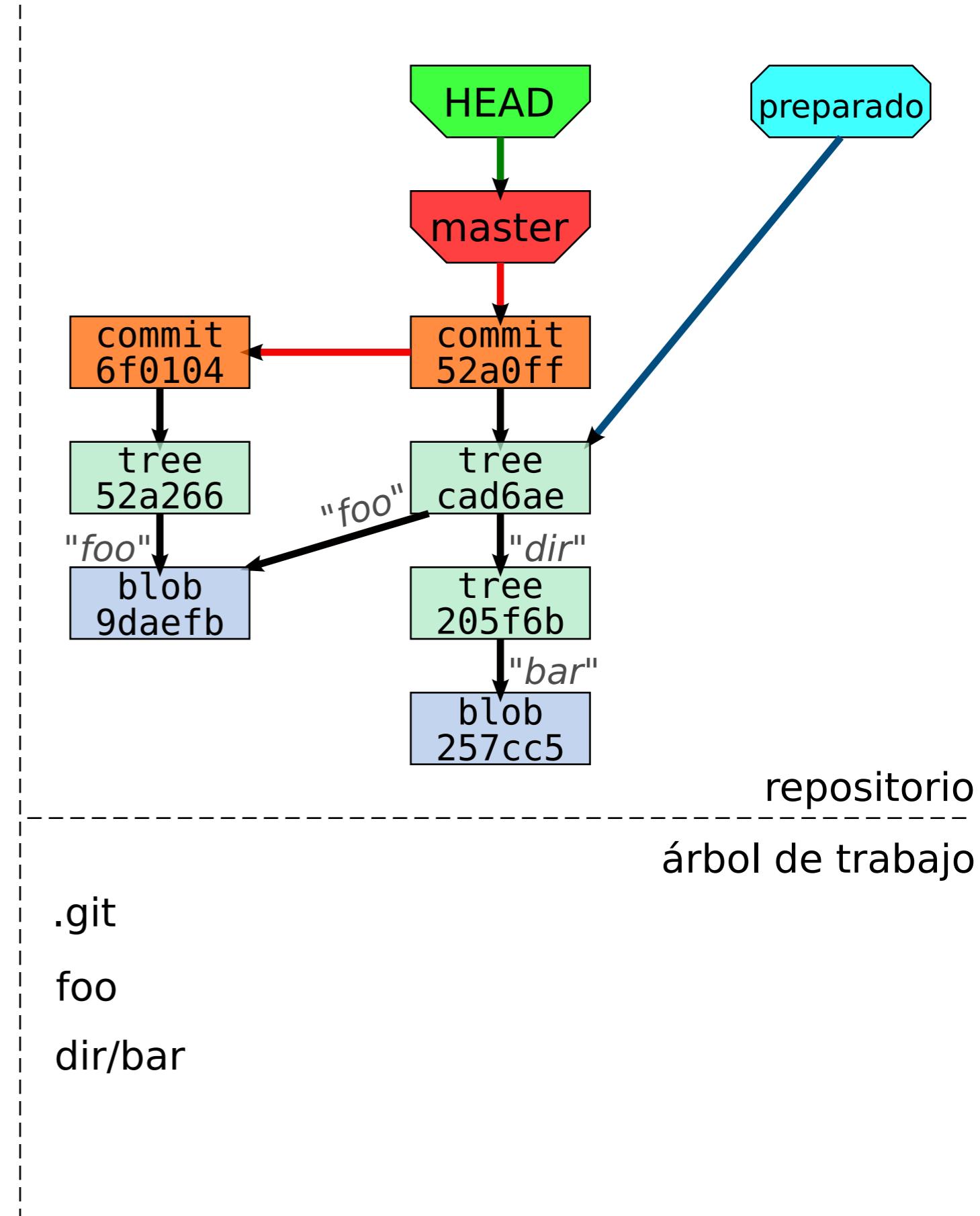
# echo foo > foo
# git add foo

# git commit -m "foo"

# mkdir dir
# echo bar > dir/bar
# git add dir/bar

# git commit -m "bar"

```



# Historial

# Mostrar revisiones

git show

# Mostrar revisiones

`git show`

- muestra la última revisión
  - identificador SHA1
  - autor
  - fecha
  - mensaje
  - diferencias

# Mostrar revisiones

## git show

```
commit 1d33b1b1b7530a1168d7c3adc44063bfe32592eb
Author: Bart Trojanowski <bart@jukie.net>
Date:   Sat Sep 12 15:52:20 2009 -0400
```

arreglado el orden de los meses

```
diff --git a/prueba b/prueba
index a652737..750d793 100644
--- a/prueba
+++ b/prueba
@@ -1,2 +1,3 @@
Enero
+Febrero
Marzo
@@ -9,3 +10,2 @@ Septiembre
Octubre
-Febrero
Noviembre
```

# Mostrar revisiones

## git show

```
commit 1d33b1b1b7530a1168d7c3adc44063bfe32592eb
Author: Bart Trojanowski <bart@jukie.net>
Date:   Sat Sep 12 15:52:20 2009 -0400
```

arreglado el orden de los meses

```
diff --git a/prueba b/prueba
index a652737..750d793 100644
--- a/prueba
+++ b/prueba
@@ -1,2 +1,3 @@
Enero
+Febrero
Marzo
@@ -9,3 +10,2 @@ Septiembre
Octubre
-Febrero
Noviembre
```

# Mostrar revisiones

## git show

```
commit 1d33b1b1b7530a1168d7c3adc44063bfe32592eb
Author: Bart Trojanowski <bart@jukie.net>
Date:   Sat Sep 12 15:52:20 2009 -0400
```

arreglado el orden de los meses

```
diff --git a/prueba b/prueba
index a652737..750d793 100644
--- a/prueba
+++ b/prueba
@@ -1,2 +1,3 @@
Enero
+Febrero
Marzo
@@ -9,3 +10,2 @@ Septiembre
Octubre
-Febrero
Noviembre
```

# Mostrar revisiones

## git show

```
commit 1d33b1b1b7530a1168d7c3adc44063bfe32592eb
Author: Bart Trojanowski <bart@jukie.net>
Date:   Sat Sep 12 15:52:20 2009 -0400
```

arreglado el orden de los meses

```
diff --git a/prueba b/prueba
index a652737..750d793 100644
--- a/prueba
+++ b/prueba
@@ -1,2 +1,3 @@
Enero
+Febrero
Marzo
@@ -9,3 +10,2 @@ Septiembre
Octubre
-Febrero
Noviembre
```

# Mostrar revisiones

## git show

```
commit 1d33b1b1b7530a1168d7c3adc44063bfe32592eb
Author: Bart Trojanowski <bart@jukie.net>
Date:   Sat Sep 12 15:52:20 2009 -0400
```

arreglado el orden de los meses

```
diff --git a/prueba b/prueba
index a652737..750d793 100644
--- a/prueba
+++ b/prueba
@@ -1,2 +1,3 @@
Enero
+Febrero
Marzo
@@ -9,3 +10,2 @@ Septiembre
Octubre
-Febrero
Noviembre
```

# Mostrar revisiones (diffstat)

`git show --stat`

# Mostrar revisiones (diffstat)

```
git show --stat
```

```
commit 3ff323f89075624b6891e7c428edb8e8a35be13c
Merge: 755ae76 a54775c
Author: Linus Torvalds <torvalds@linux-foundation.org>
Date:   Mon Sep 7 11:42:25 2009 -0700
```

Merge branch 'drm-fixes'

```
r300.c           2 +-
radeon_asic.h    6 ++-- 
rs600.c          65 ++++++-----+
rs690.c          64 -----.
rv515.c          2 +-
5 files changed, 70 insertions(+), 69 deletions(-)
```

# Más usos de 'show'

# Más usos de 'show'

`git show <revisión>`  
- revisión anterior

# Más usos de 'show'

`git show <revisión>`

- revisión anterior

`git show <revisión>:<fichero>`

- contenidos de un fichero
- se utiliza HEAD por omisión

# Registros

git log

# Registros

`git log`

- muestra registros de una revisión
- como `show`, pero muchas revisiones

# Registros

`git log <rama>`

- muestra registros de una revisión
- como `show`, pero muchas revisiones

# Registros

git log

```
commit 1d33b1b1b7530a1168d7c3adc44063bfe32592eb
Author: Bart Trojanowski <bart@jukie.net>
Date:   Sat Sep 12 15:52:20 2009 -0400
```

tercer mensaje

```
commit 895740c342fdb03ce26b6417051ab6f2188a6d0
Author: Bart Trojanowski <bart@jukie.net>
Date:   Sat Sep 12 14:53:20 2009 -0400
```

segundo mensaje

```
commit db35c00c63a7e98c4a89e180d317c9fb08e40f4b
Author: Bart Trojanowski <bart@jukie.net>
Date:   Sat Sep 12 10:08:20 2009 -0400
```

primera revisión!

# Registros (con diffstat)

git log --stat

# Registros (con diffstat)

`git log --stat`

```
commit 895740c342fdb03ce26b6417051ab6f2188a6d0
Author: Bart Trojanowski <bart@jukie.net>
Date:   Sat Sep 12 14:53:20 2009 -0400

segundo mensaje
```

```
prueba | 15 ++++++++-----
1 files changed, 12 insertions(+), 3 deletions(-)
```

```
commit db35c00c63a7e98c4a89e180d317c9fb08e40f4b
Author: Bart Trojanowski <bart@jukie.net>
Date:   Sat Sep 12 10:08:20 2009 -0400
```

i primera revisión!

```
prueba | 1 +
1 files changed, 1 insertions(+), 0 deletions(-)
```

# Registros (menos detalle)

`git log --oneline`

# Registros (menos detalle)

`git log --oneline`

```
1d33b1b quinto mensaje
91f3a84 cuarto mensaje
895740c tercer mensaje
db35c00 segundo mensaje
3fb8a8e i primera revisión!
```

# Registros (limitados)

git log -10

# Registros (limitados)

`git log -10`

- limitado a 10 revisiones

# Registros (grafo de rev.)

`git log --graph`

# Registros (grafo de rev.)

git log --graph

```
# git log --graph --oneline
* cf843d3 Merge commit '6727836'
  \
   * 6727836 Merge commit 'b335301'
   |
   | \
   |  * b335301 rama izquierda
   | *
   |  * 68f464c rama antiaua
   *
   |  * 9923d93 rama derecha
   |
   | /
   | /
   * 1d33b1b arreglado orden de los meses
   |
   * 91f3a84 estropeado orden de los meses
   * 895740c más revisiones
   * db35c00 otra revisión
   * 3fb8a8e mi mensaje de revisión
```

# Registros (otras cosas)

# Registros (otras cosas)

- por fecha

```
git log @{yesterday}  
git log --since="May 1" \  
--until="June 1"
```

# Registros (otras cosas)

- por fecha

```
git log @{yesterday}  
git log --since="May 1" \  
--until="June 1"
```

- por persona

```
git log --author=fred  
git log --committer=joe
```

# Registros (otras cosas)

- por fecha

```
git log @{yesterday}  
git log --since="May 1" \  
--until="June 1"
```

- por persona

```
git log --author=fred  
git log --committer=joe
```

- por mensaje

```
git log --grep="una.*regexp"
```

# Registros (otras cosas)

- por fecha

```
git log @{yesterday}  
git log --since="May 1" \  
--until="June 1"
```

- por persona

```
git log --author=fred  
git log --committer=joe
```

- por mensaje

```
git log --grep="una.*regexp"
```

# Registros (otras cosas)

- por cambios hechos  
`git log -Sfuncion`

# Registros (otras cosas)

- por cambios hechos  
`git log -Sfuncion`
- con parches  
`git log -p`

# Registros (otras cosas)

- por cambios hechos

`git log -Sfuncion`

- con parches

`git log -p`

- con interfaz gráfica

`gitk <opciones de "git log">`

# Referencias

# Referencias

("refs")

- señalan a revisiones

- señalan a revisiones
- ligeras

- señalan a revisiones
- ligeras
- mutables

- señalan a revisiones
- ligeras
- mutables
- se pueden eliminar

3 tipos:

3 tipos:

ramas

master



895740c342fdbcb03ce26b6417051ab6f2188a6d0

3 tipos:

ramas

etiquetas ("tags")

master

tag

```
graph TD; master --> hash[895740c342fdb03ce26b6417051ab6f2188a6d0]; tag --> hash;
```

895740c342fdb03ce26b6417051ab6f2188a6d0

3 tipos:

ramas

etiquetas ("tags")

HEAD

HEAD

master

tag

895740c342fdbcb03ce26b6417051ab6f2188a6d0

# Ramas

git branch

# Ramas

```
git clone git://github.com/git/git.git  
cd git  
git branch
```

# Ramas

```
git clone git://github.com/git/git.git
```

```
cd git
```

```
git branch
```

```
* master
```

# Ramas

```
git clone git://github.com/git/git.git
```

```
cd git
```

```
git branch
```

```
* master
```

```
git branch -r
```

# Ramas

```
git clone git://github.com/git/git.git
cd git
git branch
```

```
* master
```

```
git branch -r
```

```
origin/HEAD -> origin/master
origin/html
origin/maint
origin/man
origin/master
origin/next
origin/pu
origin/todo
```

# Ramas

git branch

# Ramas

git branch

```
defecto-123
funcion-123
* master
trabajando-en-funcion-x
```

# Ramas

`git branch <nombre>`

# Ramas

`git branch <nombre>`

- crea una nueva rama

# Ramas

`git branch <nombre> <ref>`  
- crea una nueva rama

# Ramas

`git branch <nombre> <ref>`

- crea una nueva rama

`git checkout <nombre>`

- cambia a una rama

# Ramas

```
git branch <nombre> <ref>
```

- crea una nueva rama

```
git checkout <nombre>
```

- cambia a una rama

# Ramas

```
git branch <nombre> <ref>
```

- crea una nueva rama

```
git checkout <nombre>
```

- cambia a una rama

```
git checkout -b <nombre> <ref>
```

# Etiquetas ("tags")

git tag

# Etiquetas ("tags")

git tag

- muestra las etiquetas

# Etiquetas ("tags")

`git tag`

- muestra las etiquetas

`git tag <nombre>`

- crea una etiqueta privada

# Etiquetas ("tags")

`git tag`

- muestra las etiquetas

`git tag <nombre>`

- crea una etiqueta privada

`git tag -a <nombre>`

- crea un objeto de tipo "tag"
- se puede compartir con otros

# Reuniones ("merges")

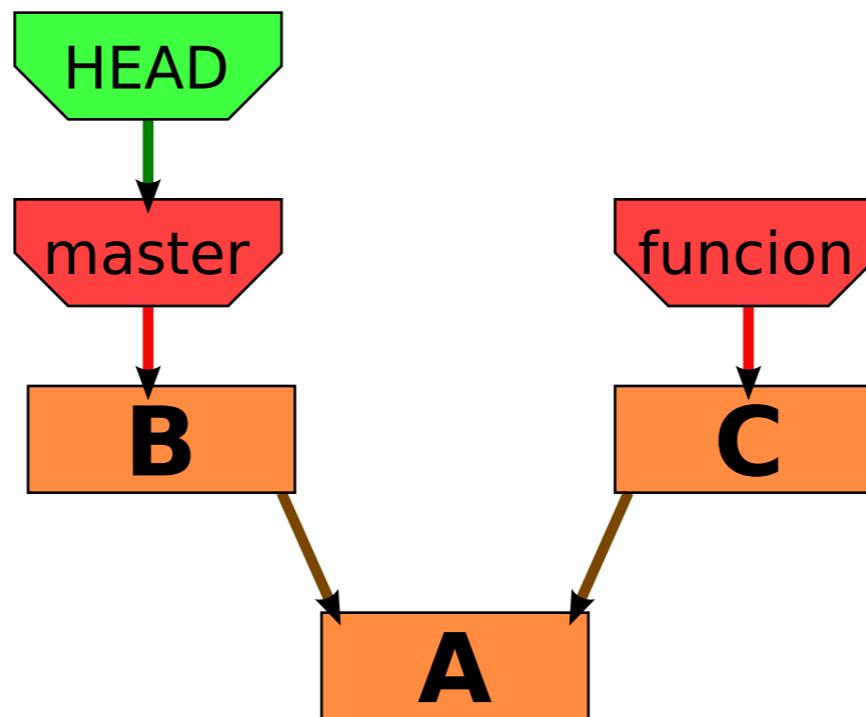
```
git merge <rama>
```

# Reuniones ("merges")

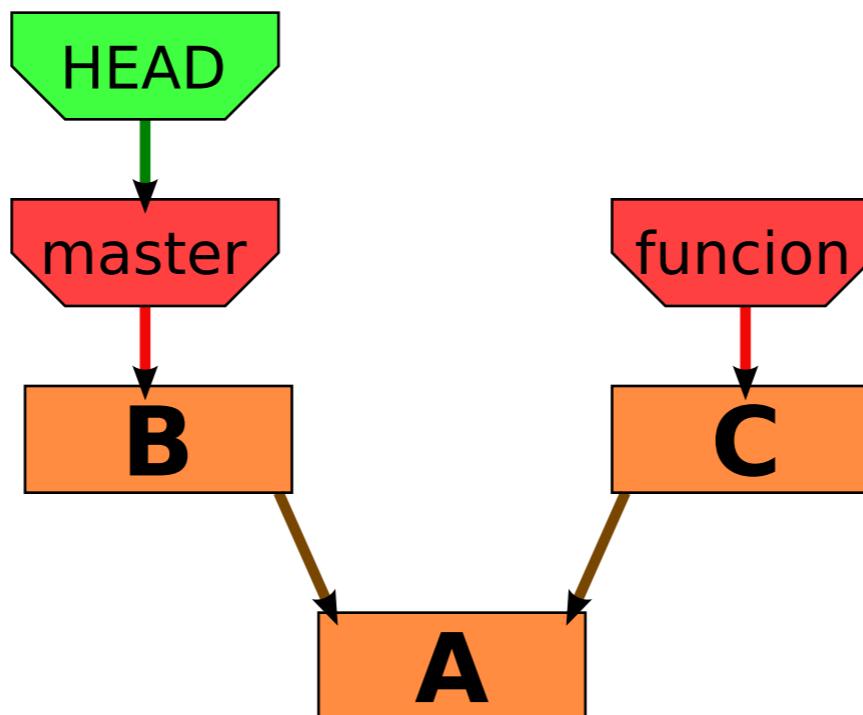
`git merge <rama>`

- reúne `<rama>` con `HEAD`
- crea revisión nueva en `HEAD`
- a veces no hace falta crear una:  
`fast-forward`

# Reuniones ("merges")

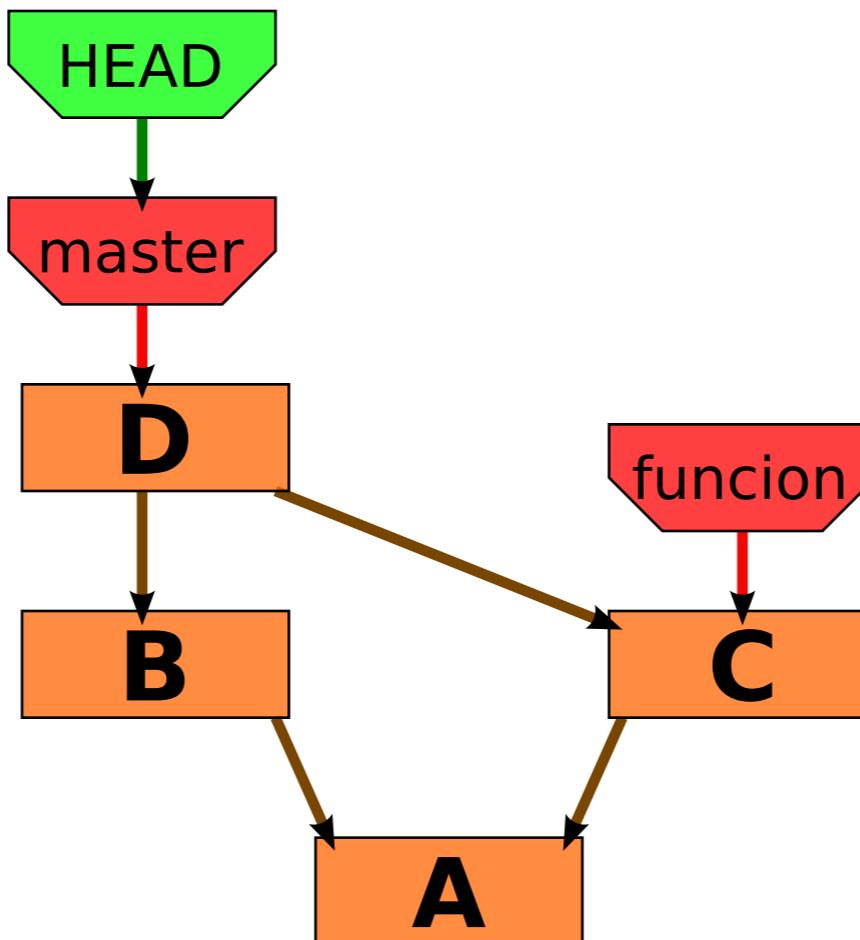


# Reuniones ("merges")



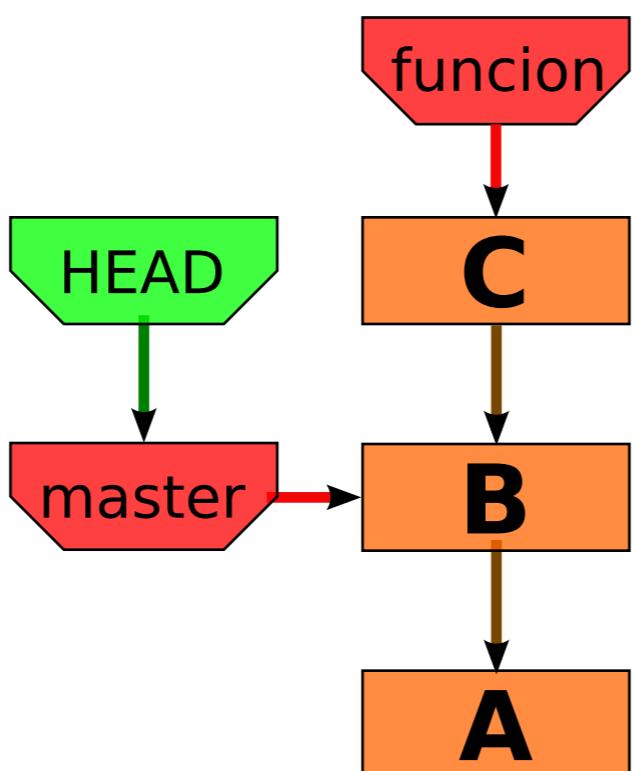
git merge **funcion**

# Reuniones ("merges")

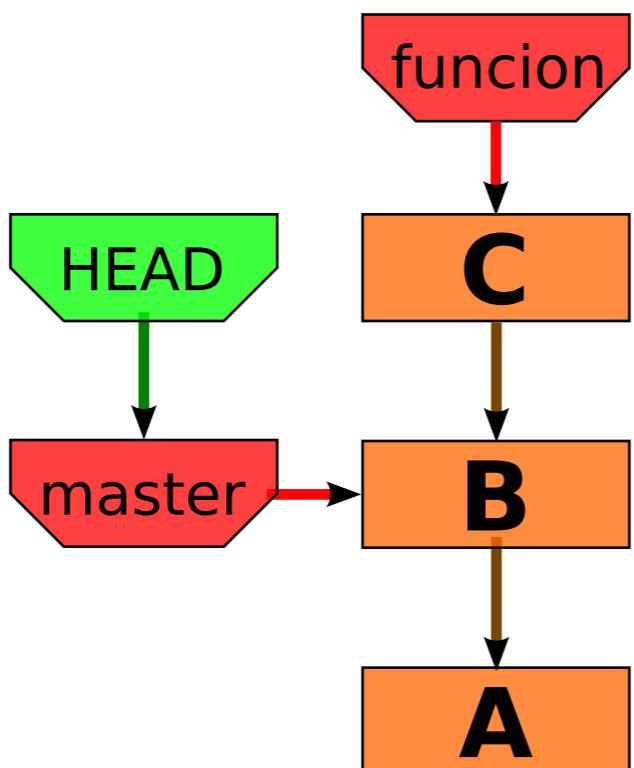


git merge **funcion**

# Reuniones ("merges")

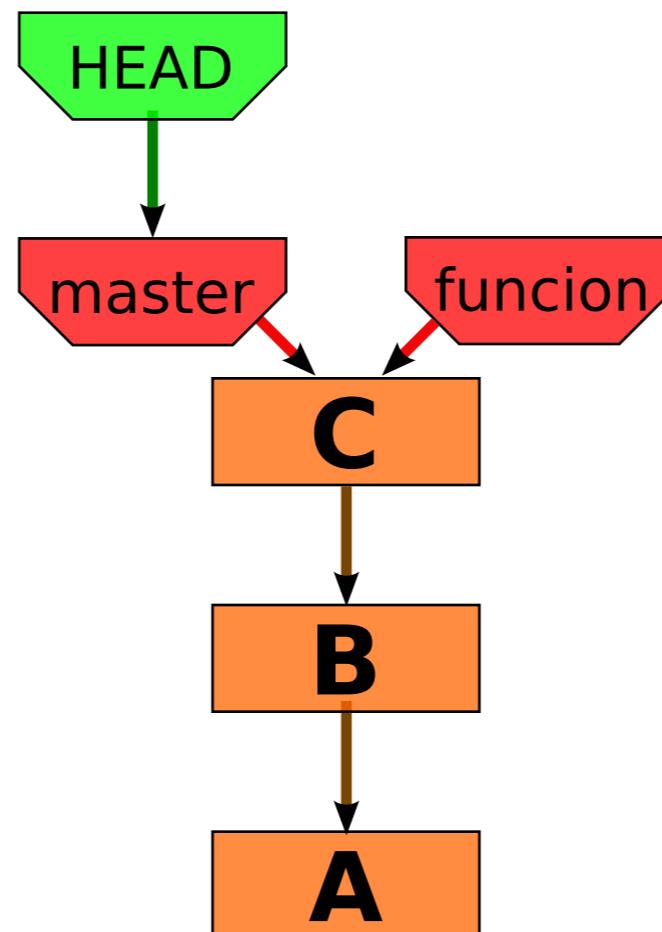


# Reuniones ("merges")



`git merge funcion`

# Reuniones ("merges")



`git merge funcion`

# Replantear ("rebase")

```
git rebase <rama>
```

# Replantear ("rebase")

`git rebase <rama>`

- mueve **HEAD** a otra revisión base

# Replantear ("rebase")

`git rebase <rama>`

- mueve **HEAD** a otra revisión base

## ¡AVISO!

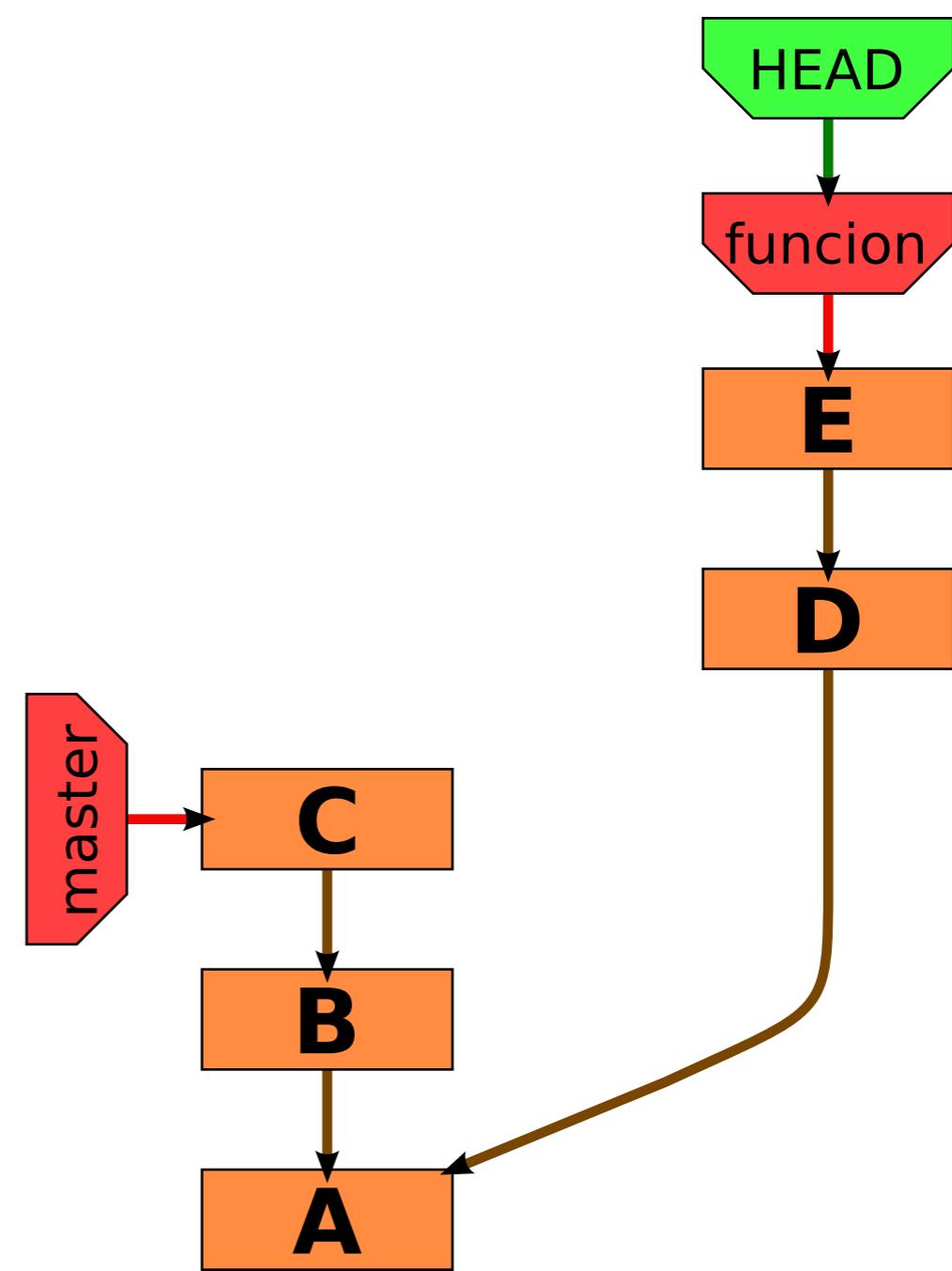
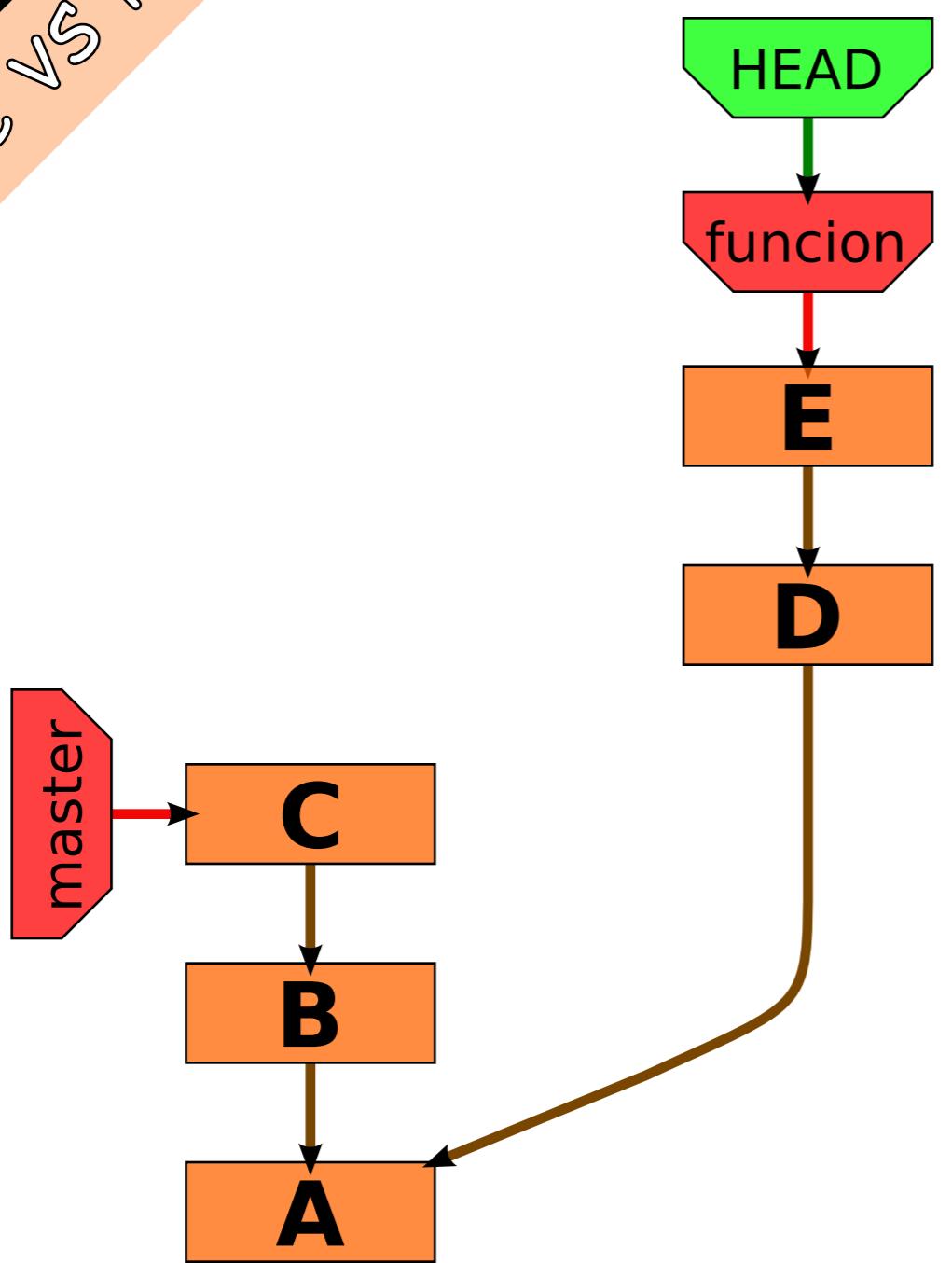
**rebase** cambia el historial

evitad **rebase** en ramas compartidas

# Merge vs Rebase

# Uso de Git

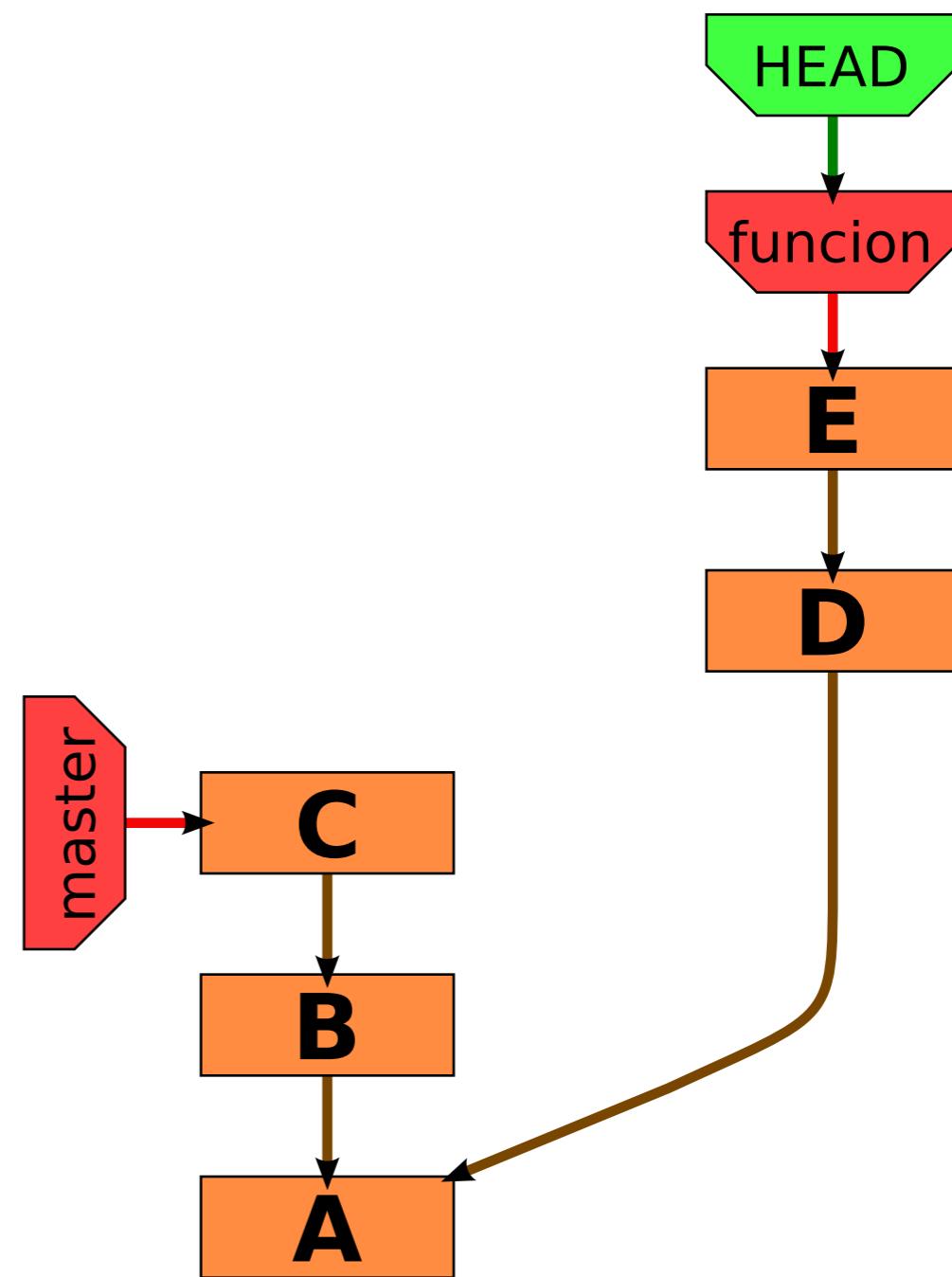
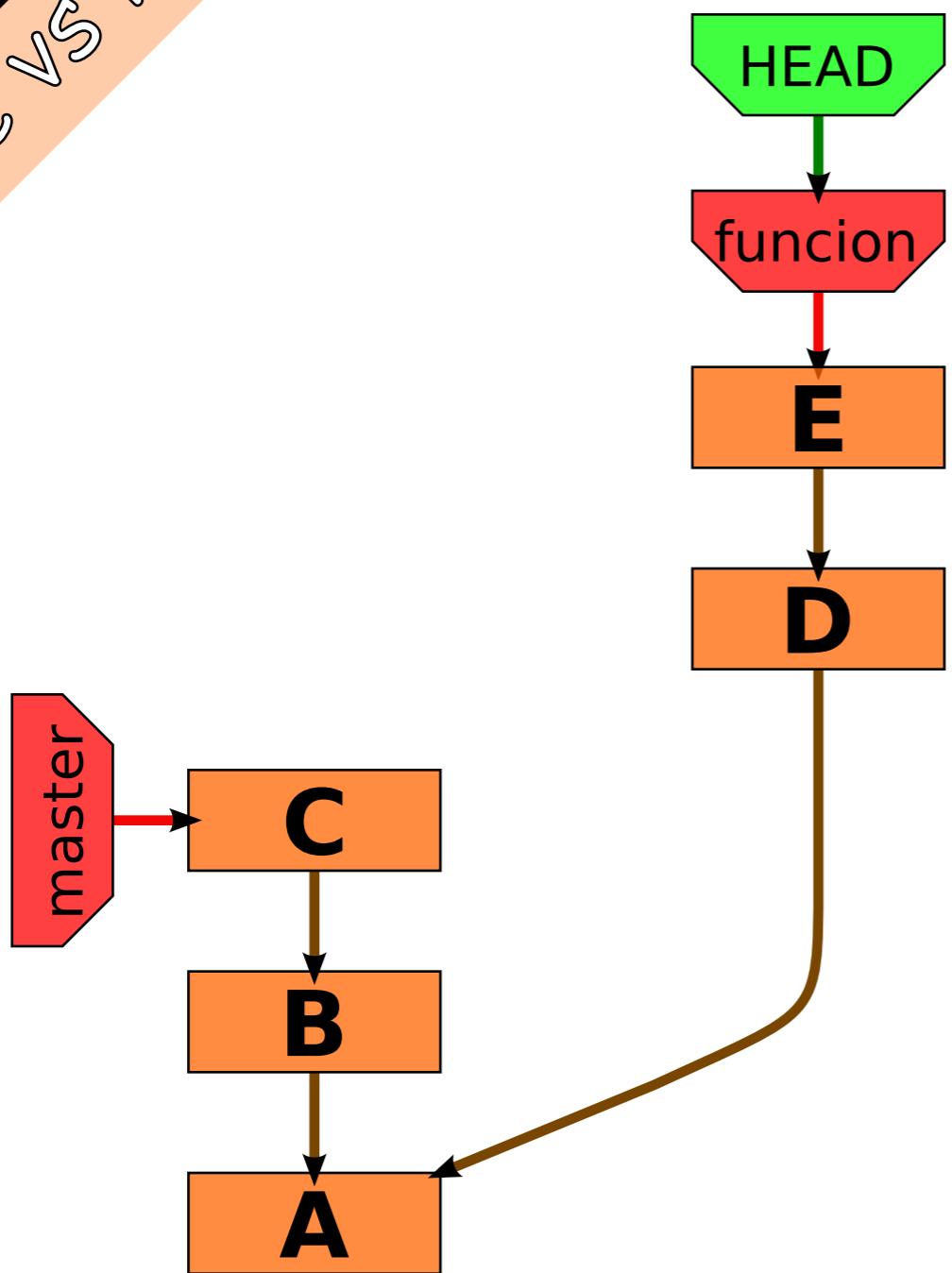
## merge vs rebase



dos grafos idénticos

# Uso de Git

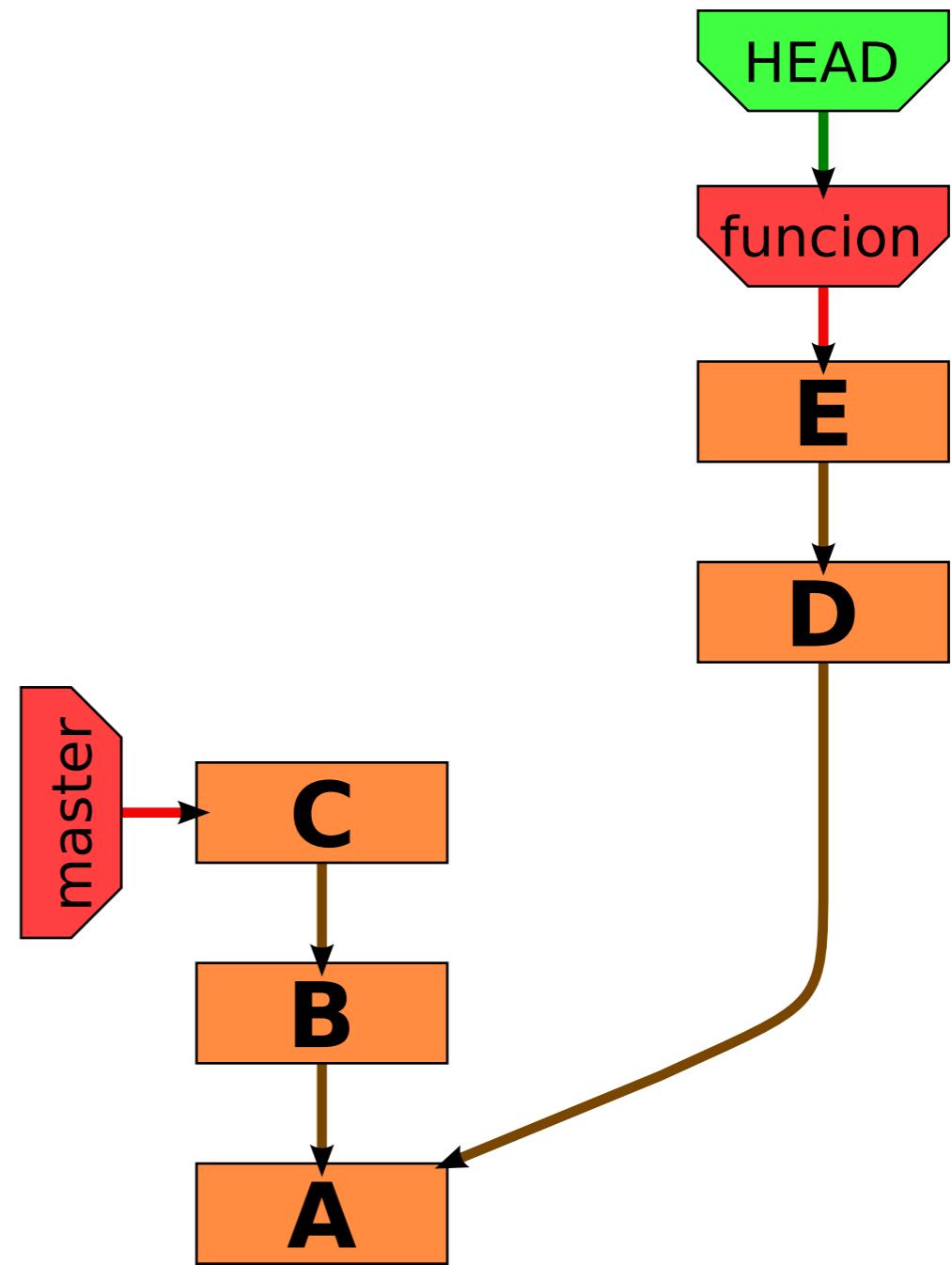
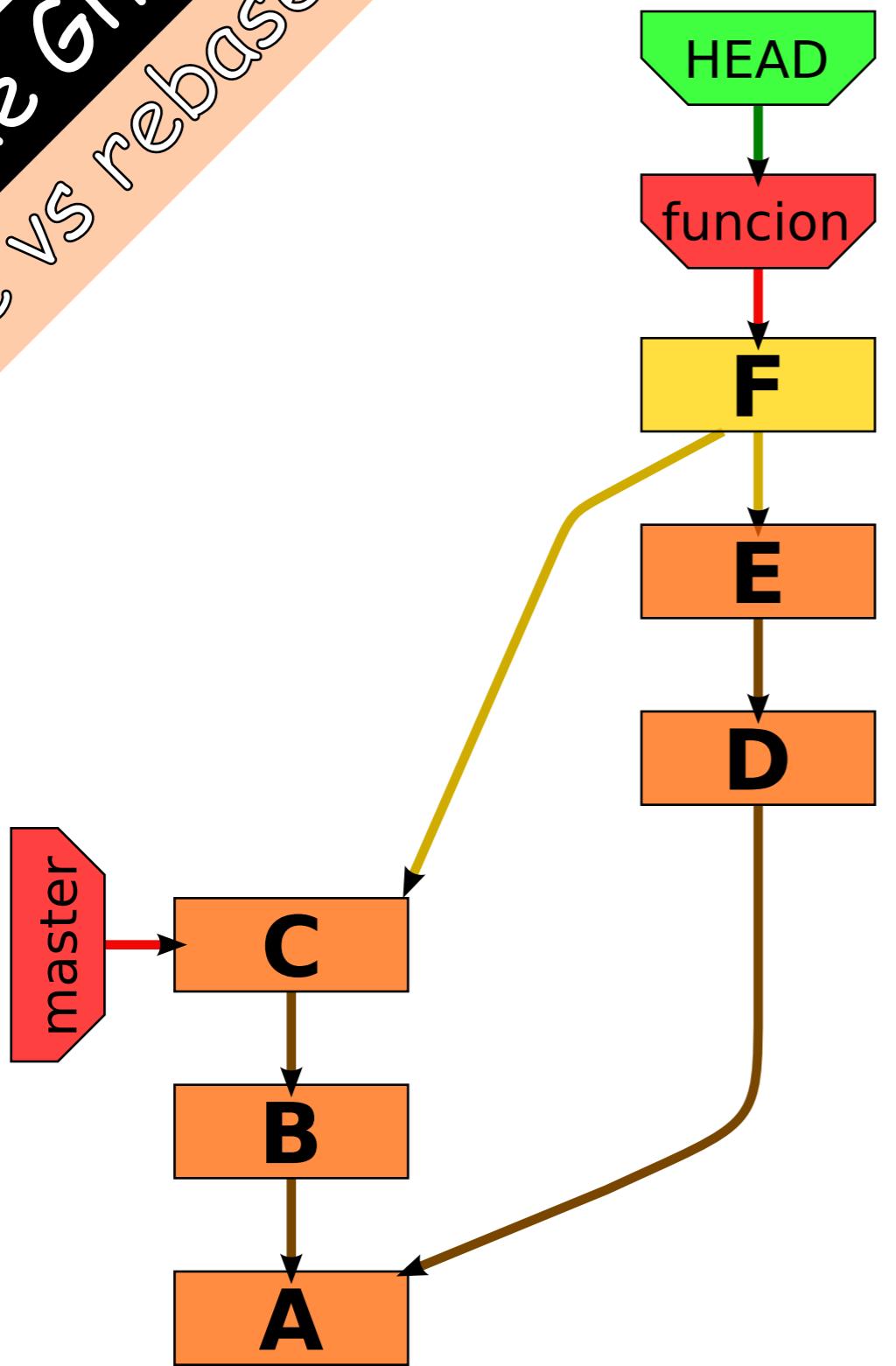
## merge vs rebase



git merge master

# Uso de Git

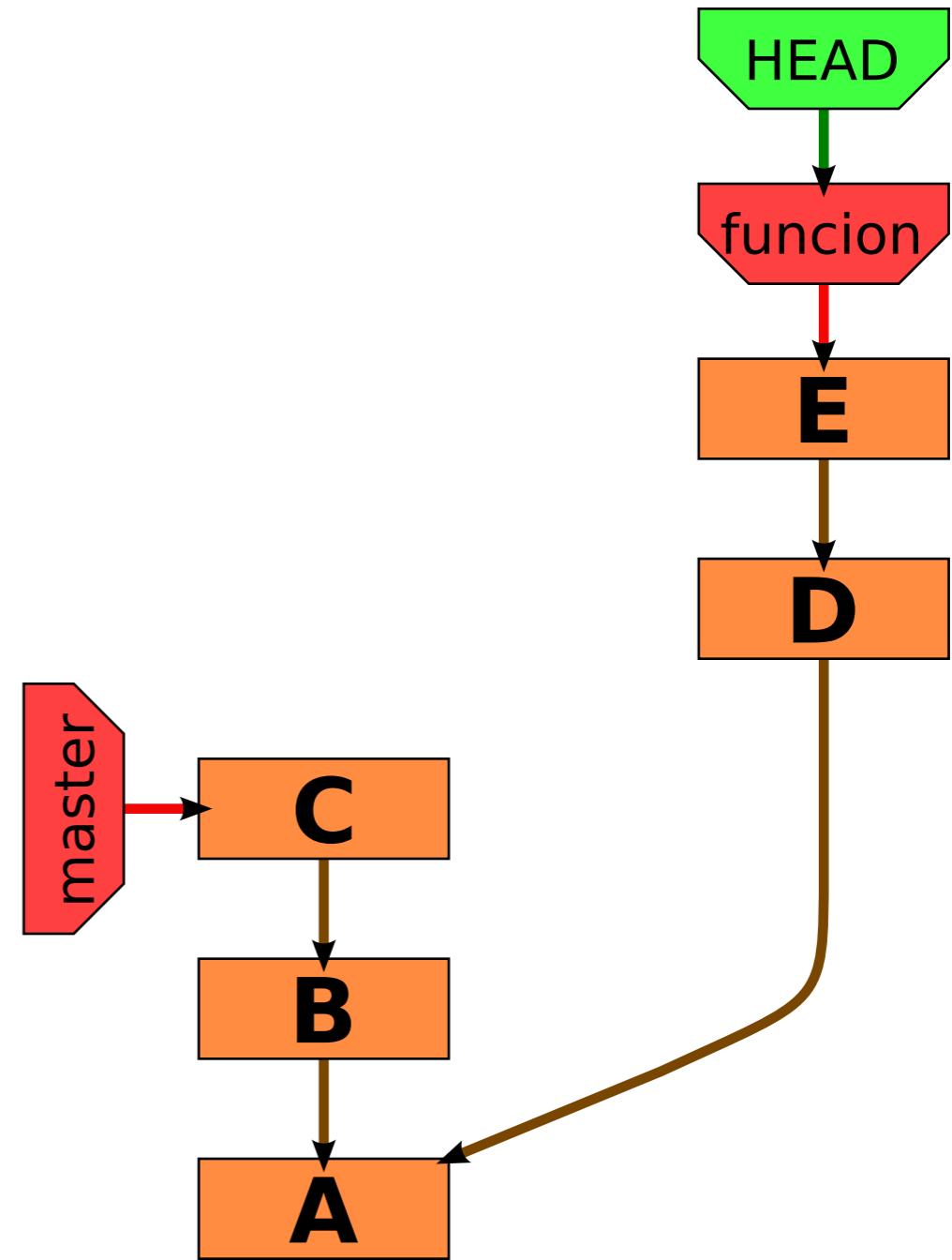
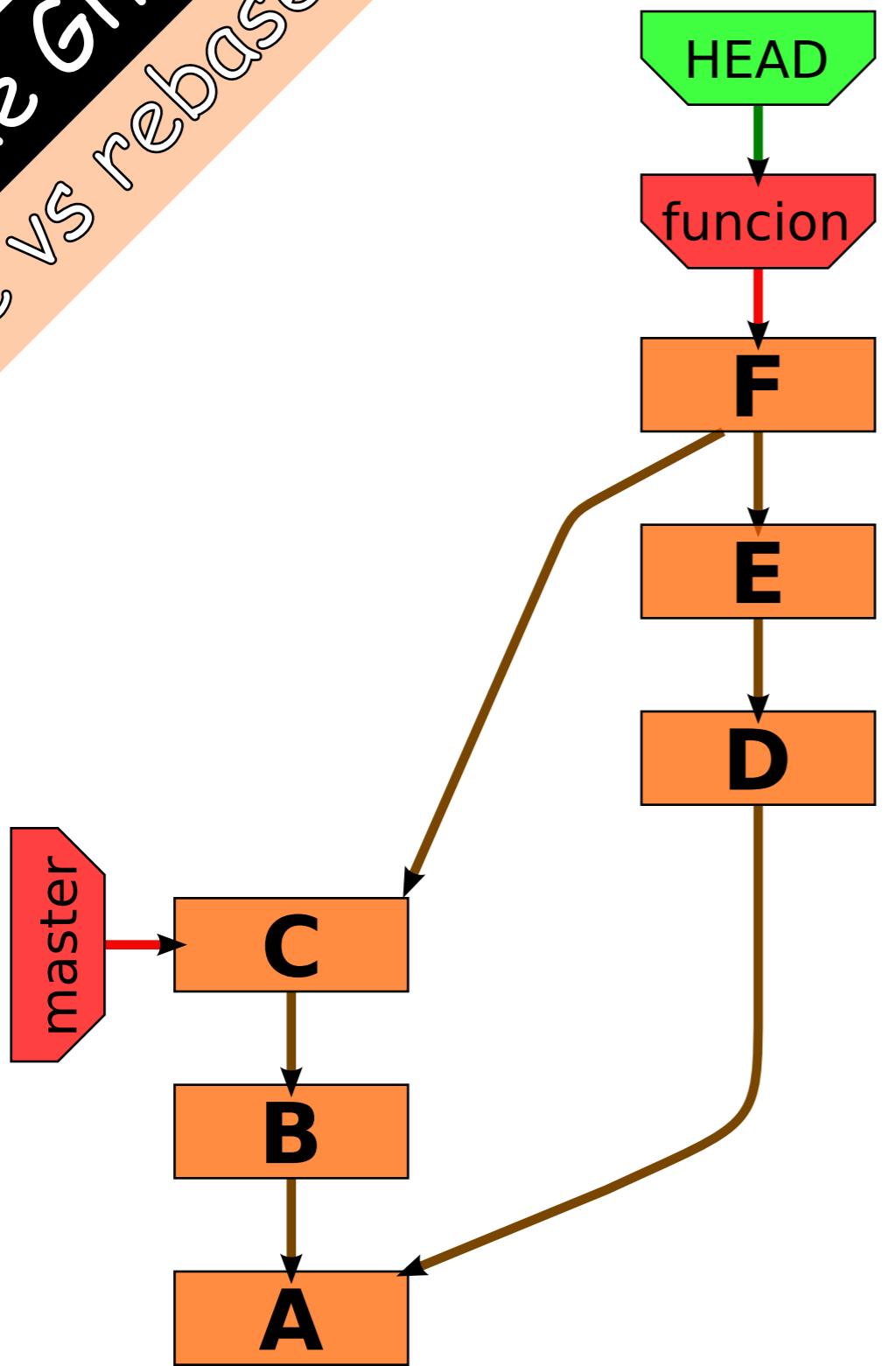
## merge vs rebase



git merge master

# Uso de Git

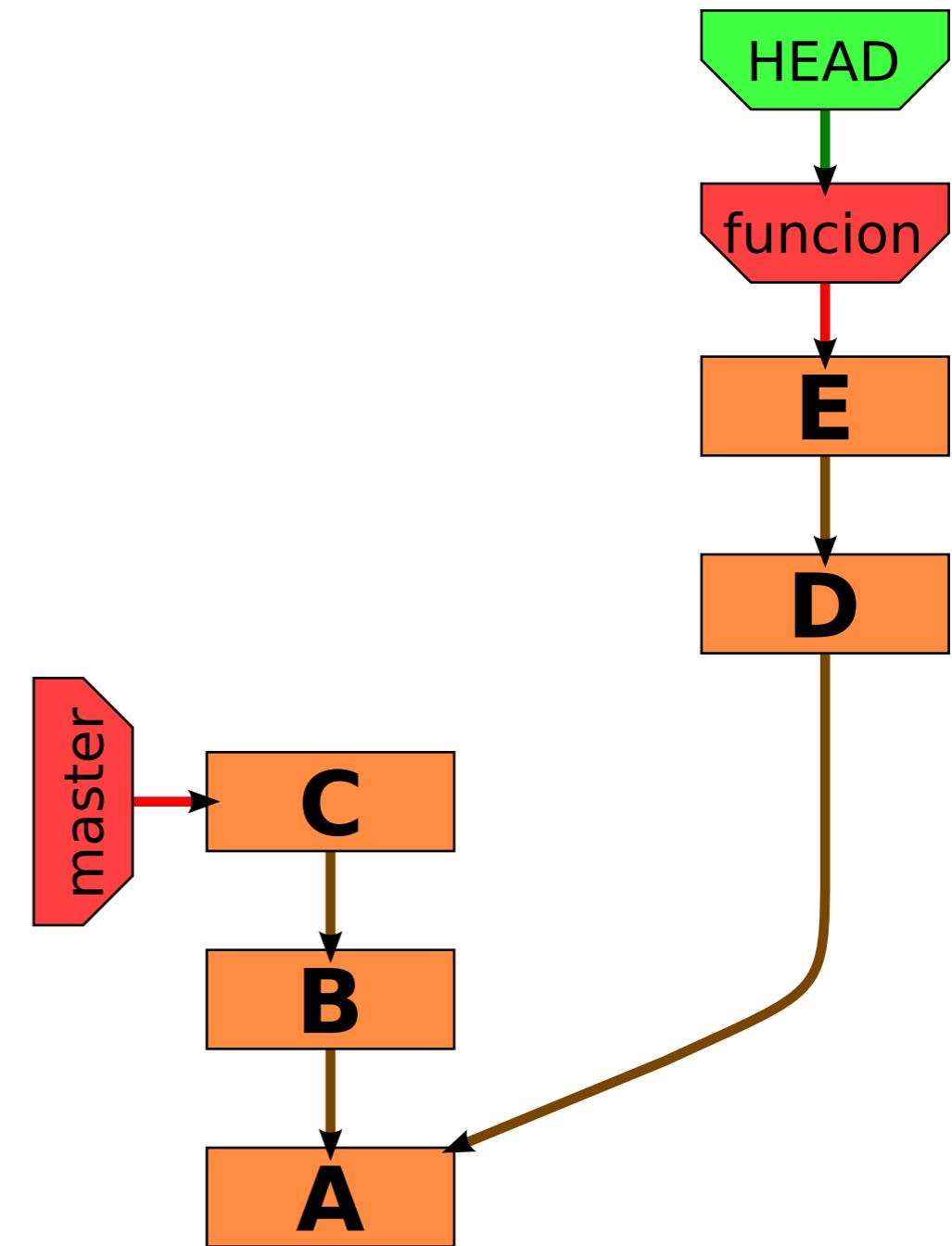
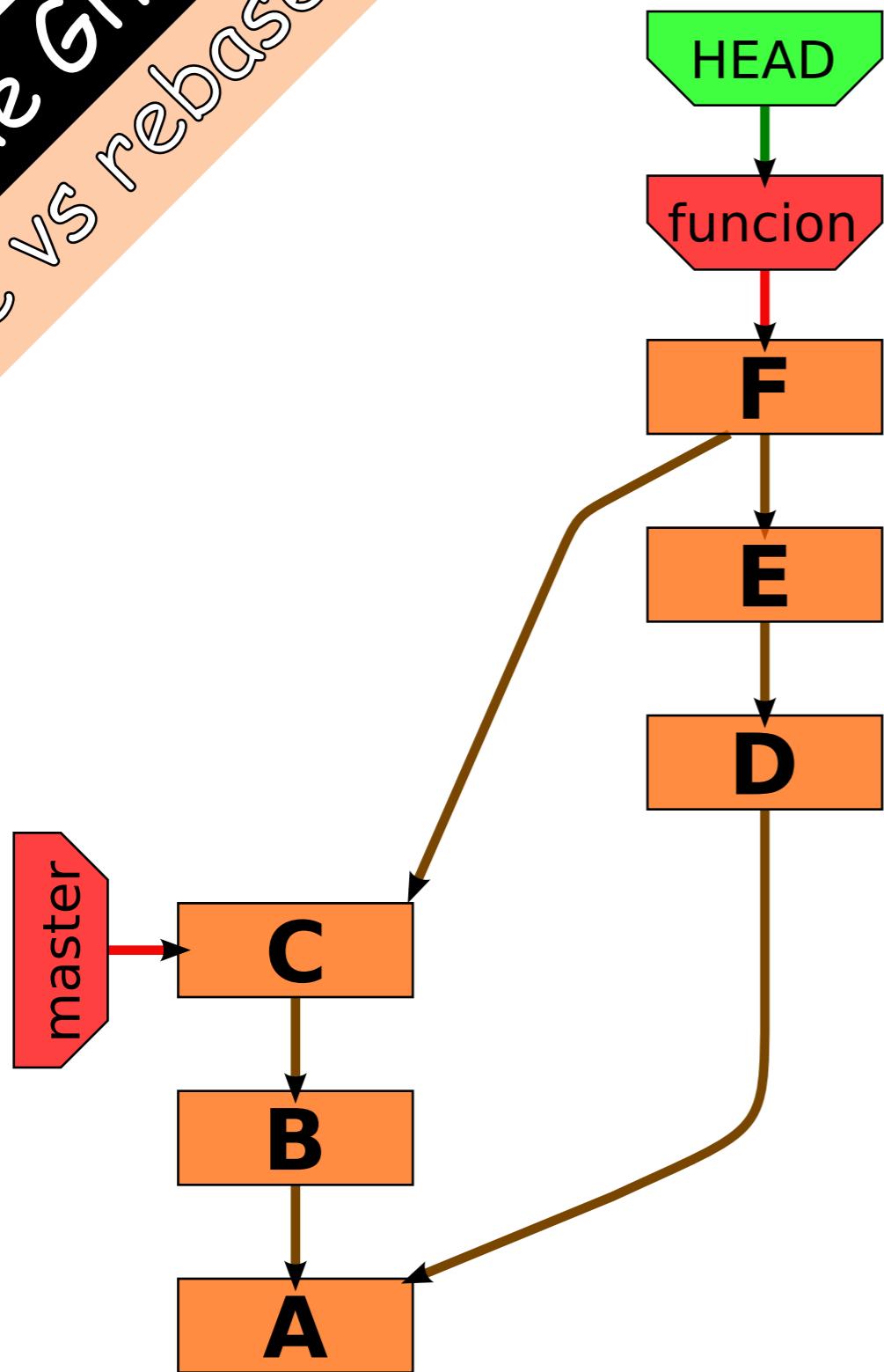
## merge vs rebase



ies fácil!

# Uso de Git

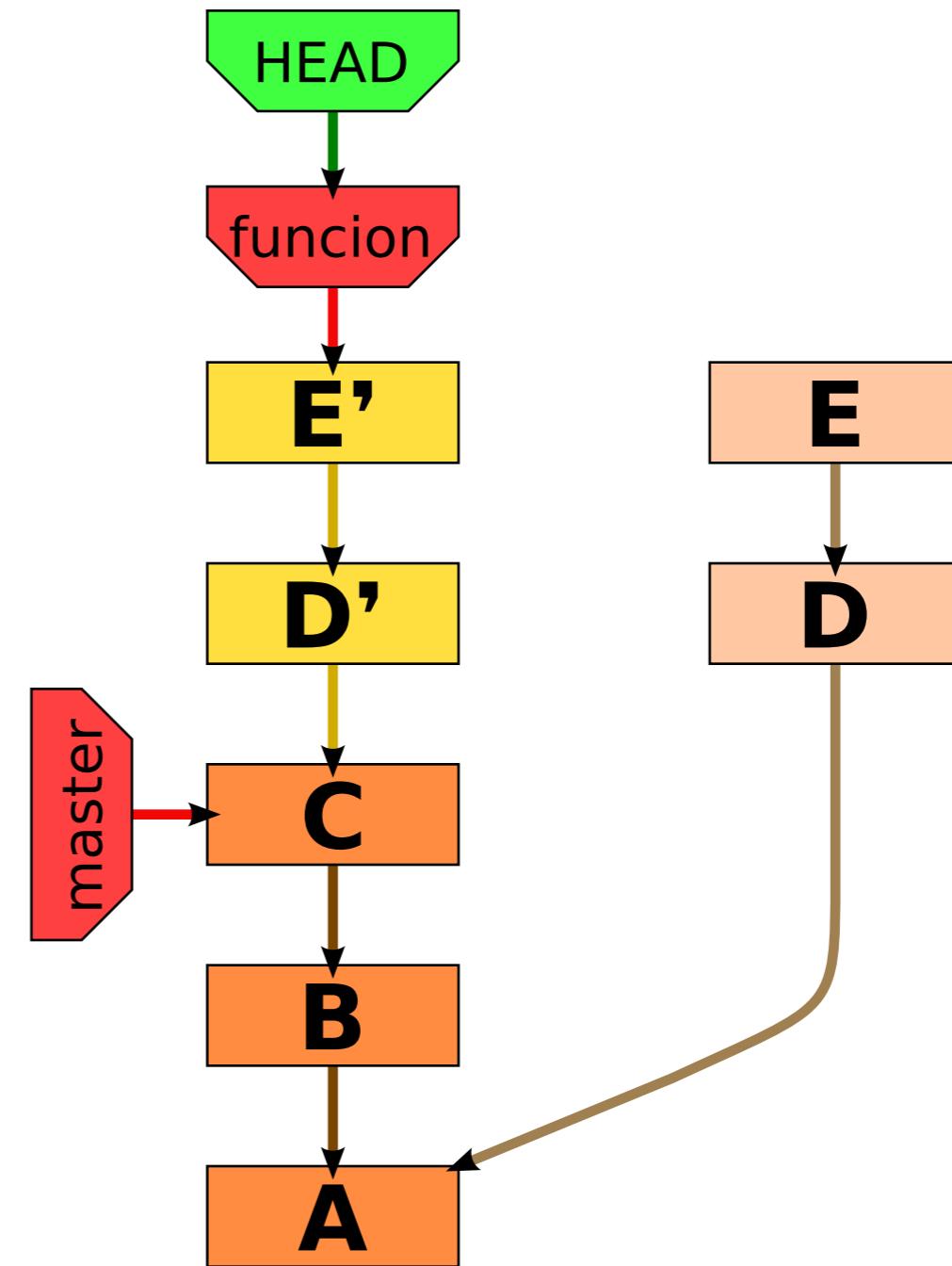
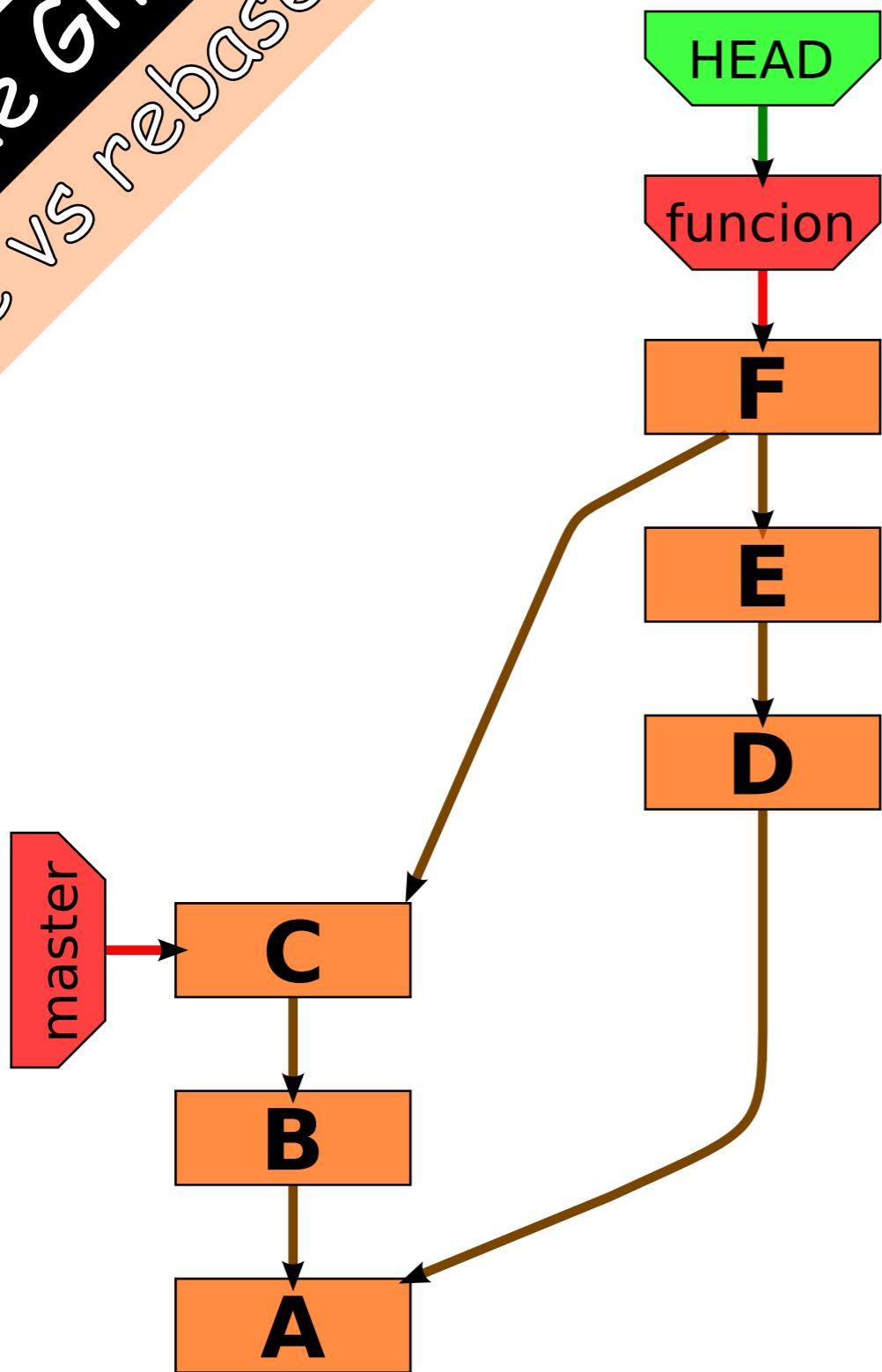
## merge vs rebase



git rebase master

# Uso de Git

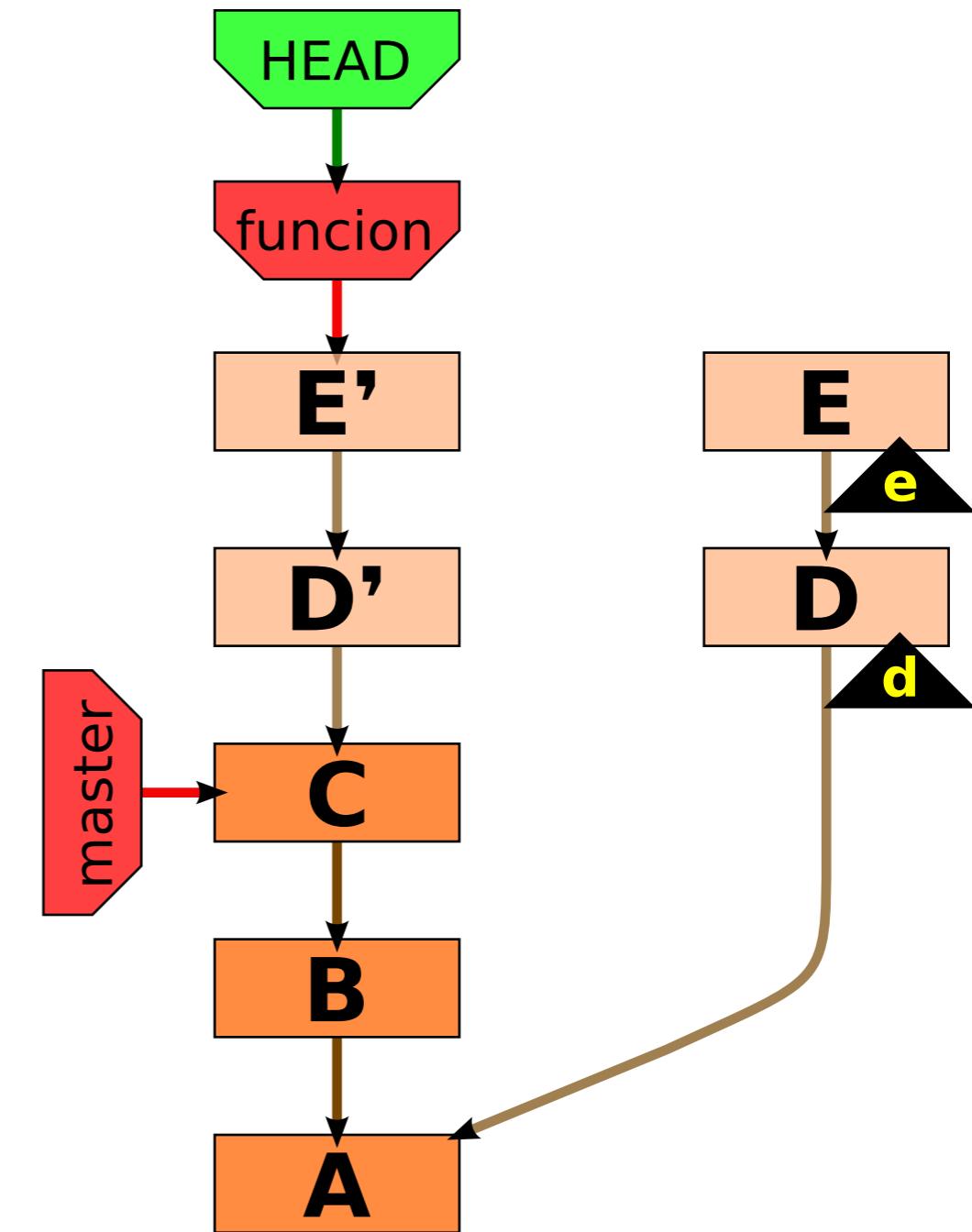
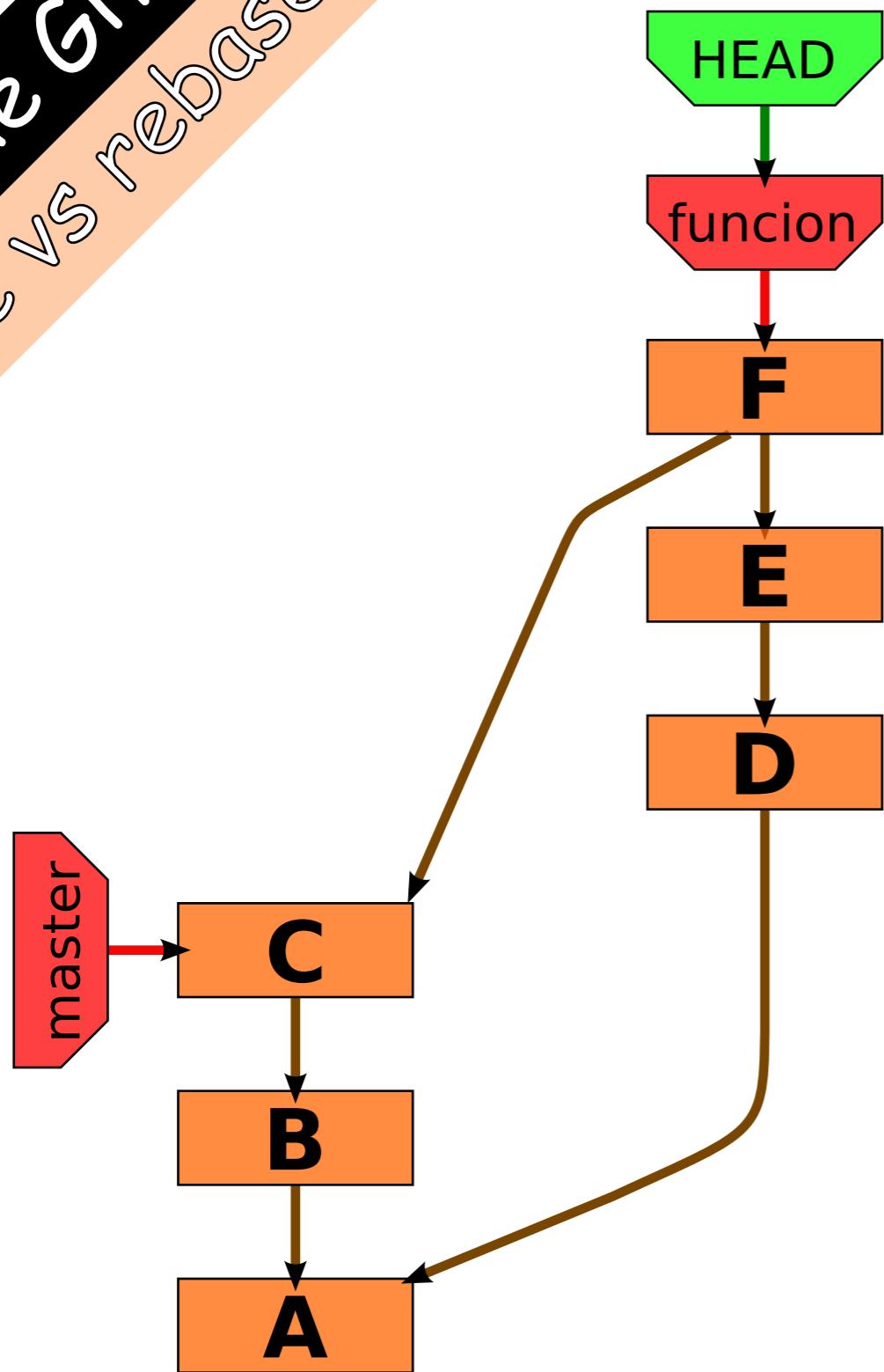
## merge vs rebase



git rebase master

# Uso de Git

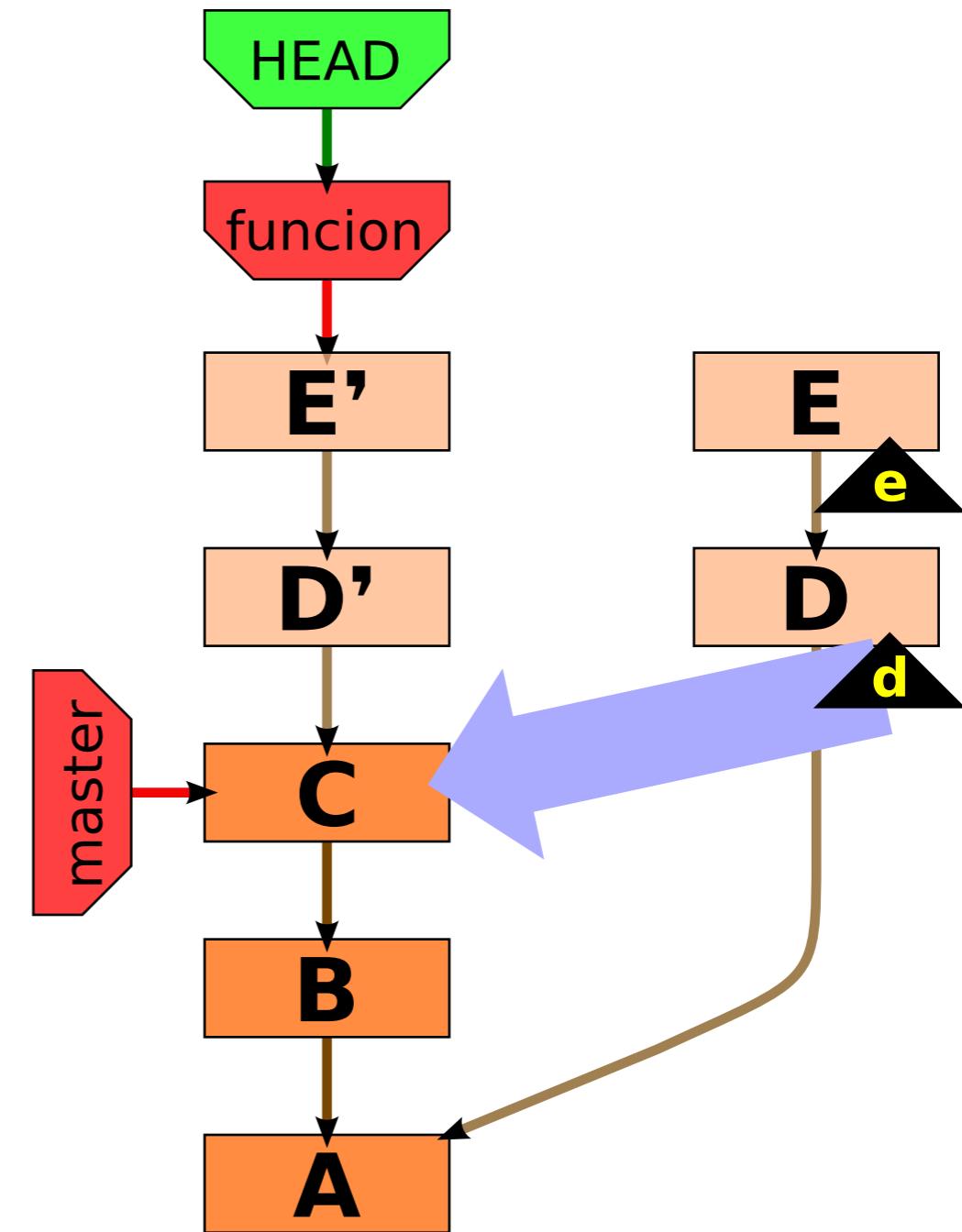
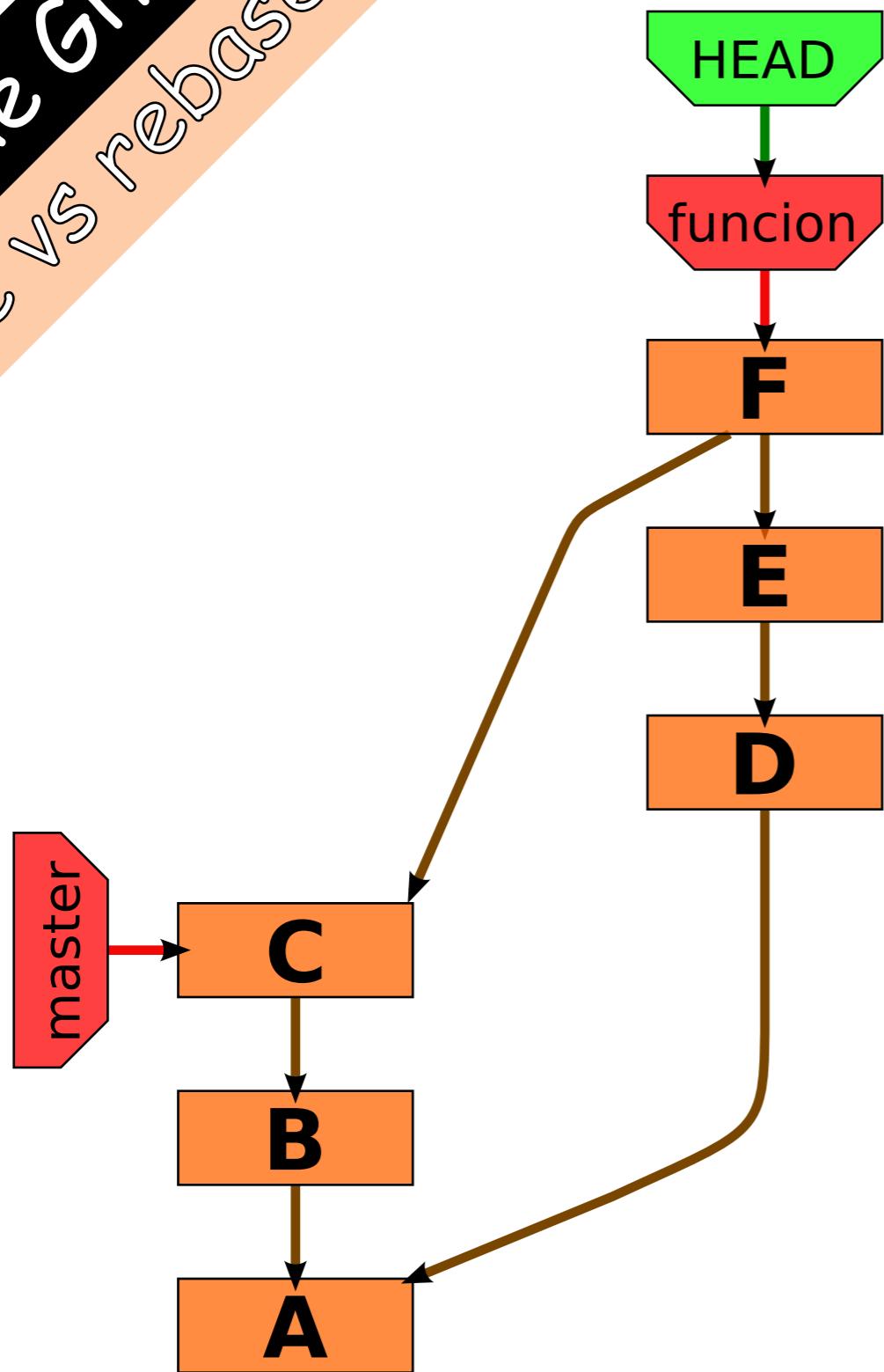
## merge vs rebase



git rebase master

# Uso de Git

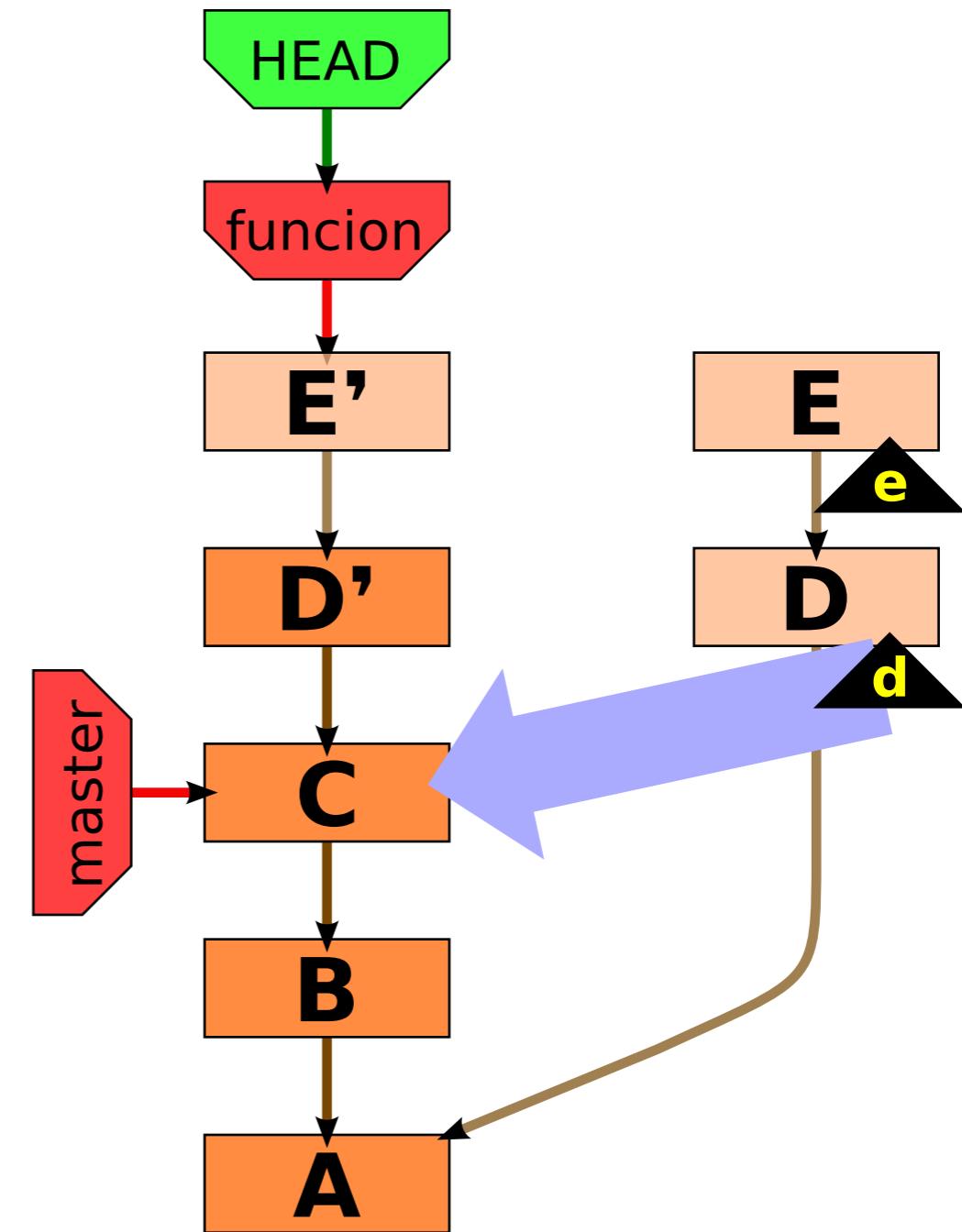
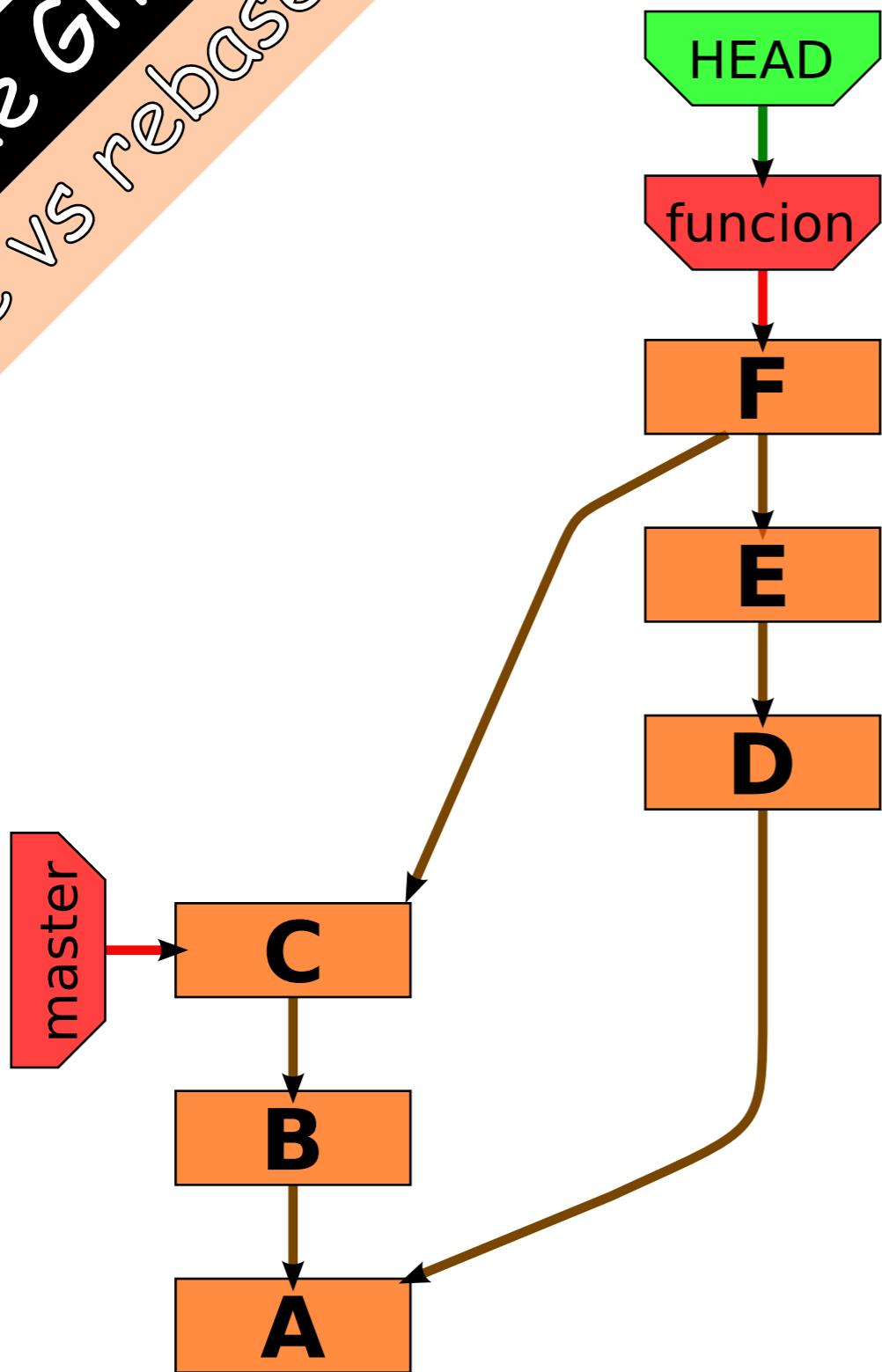
## merge vs rebase



git rebase master

# Uso de Git

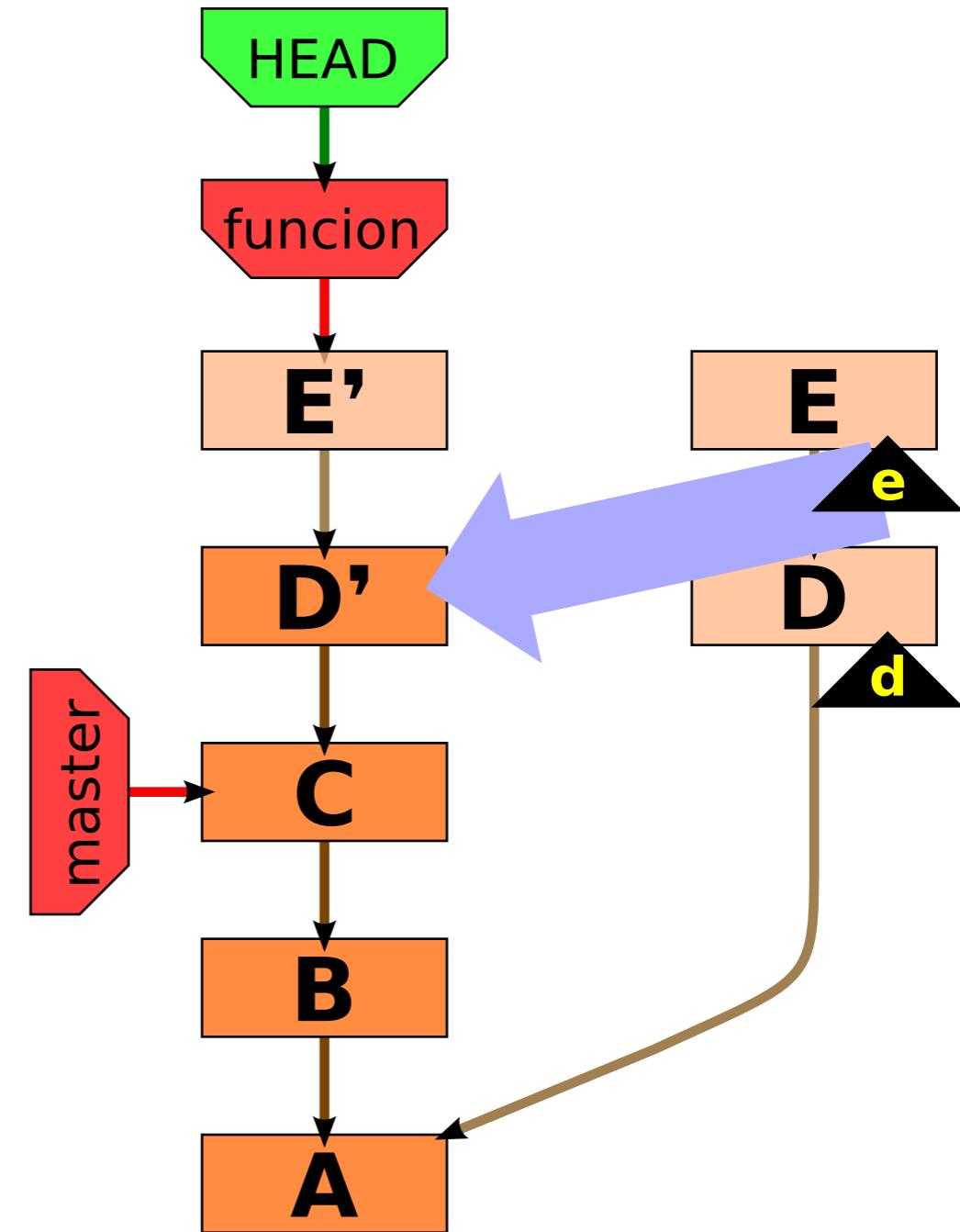
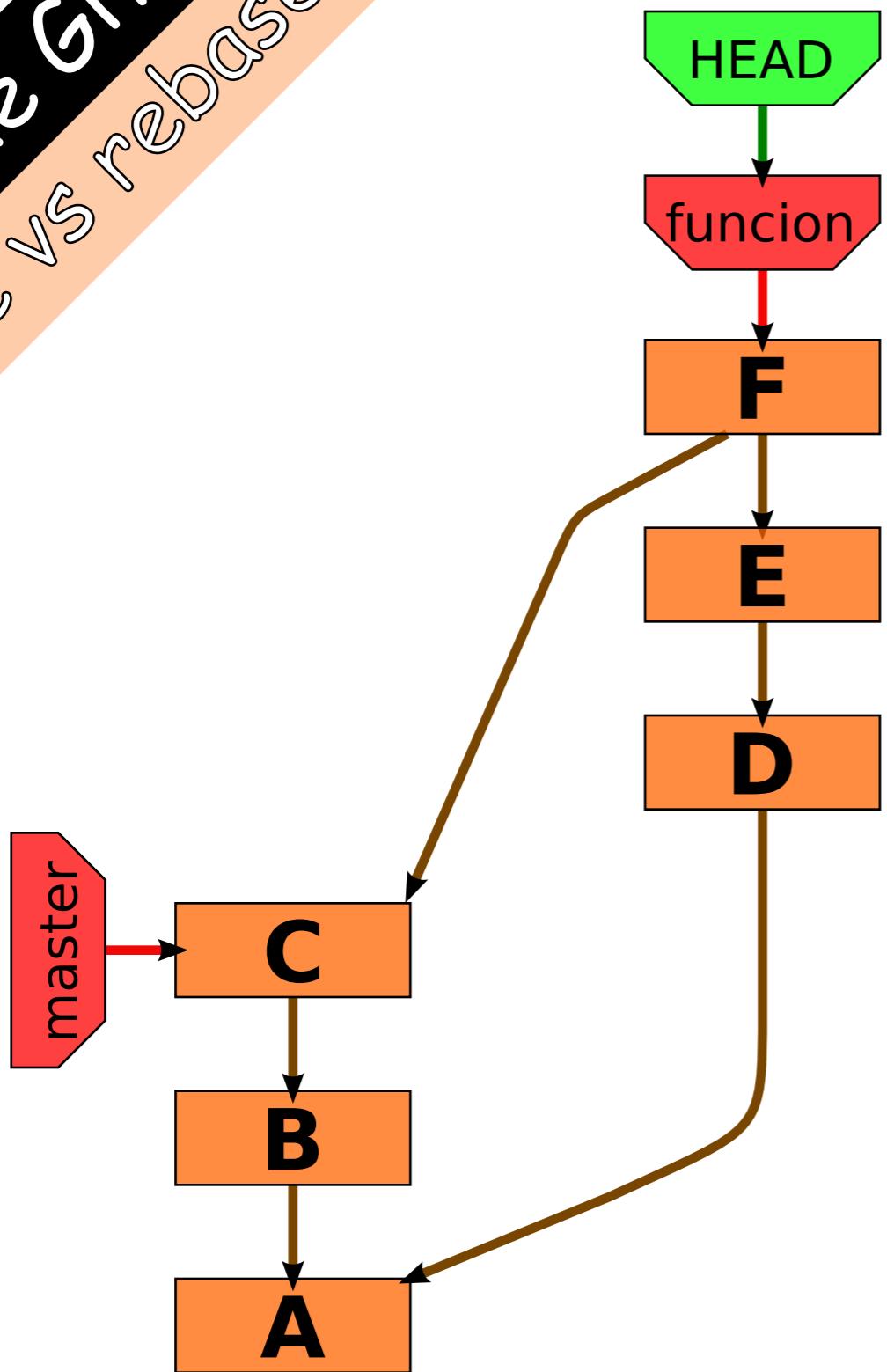
## merge vs rebase



git rebase master

# Uso de Git

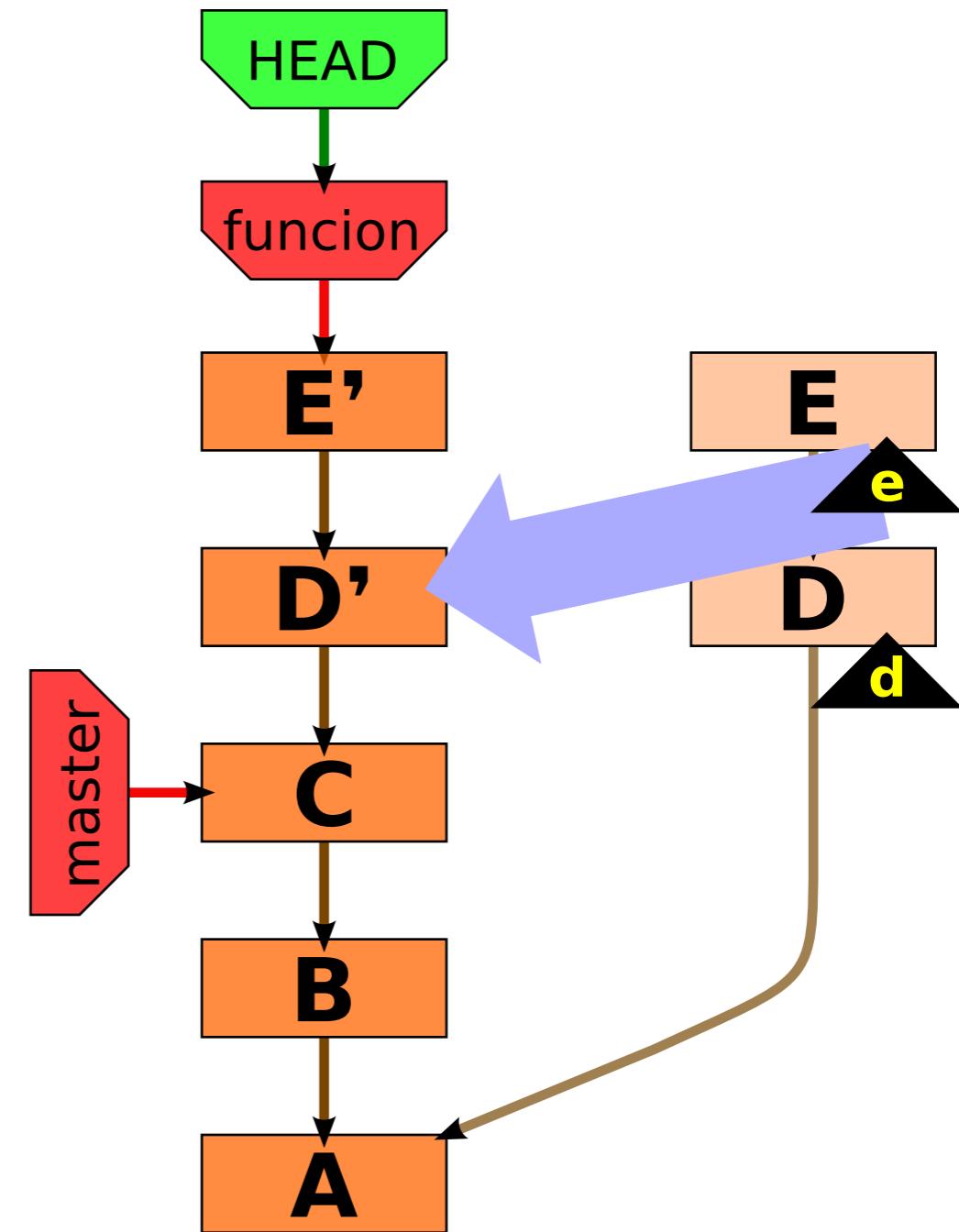
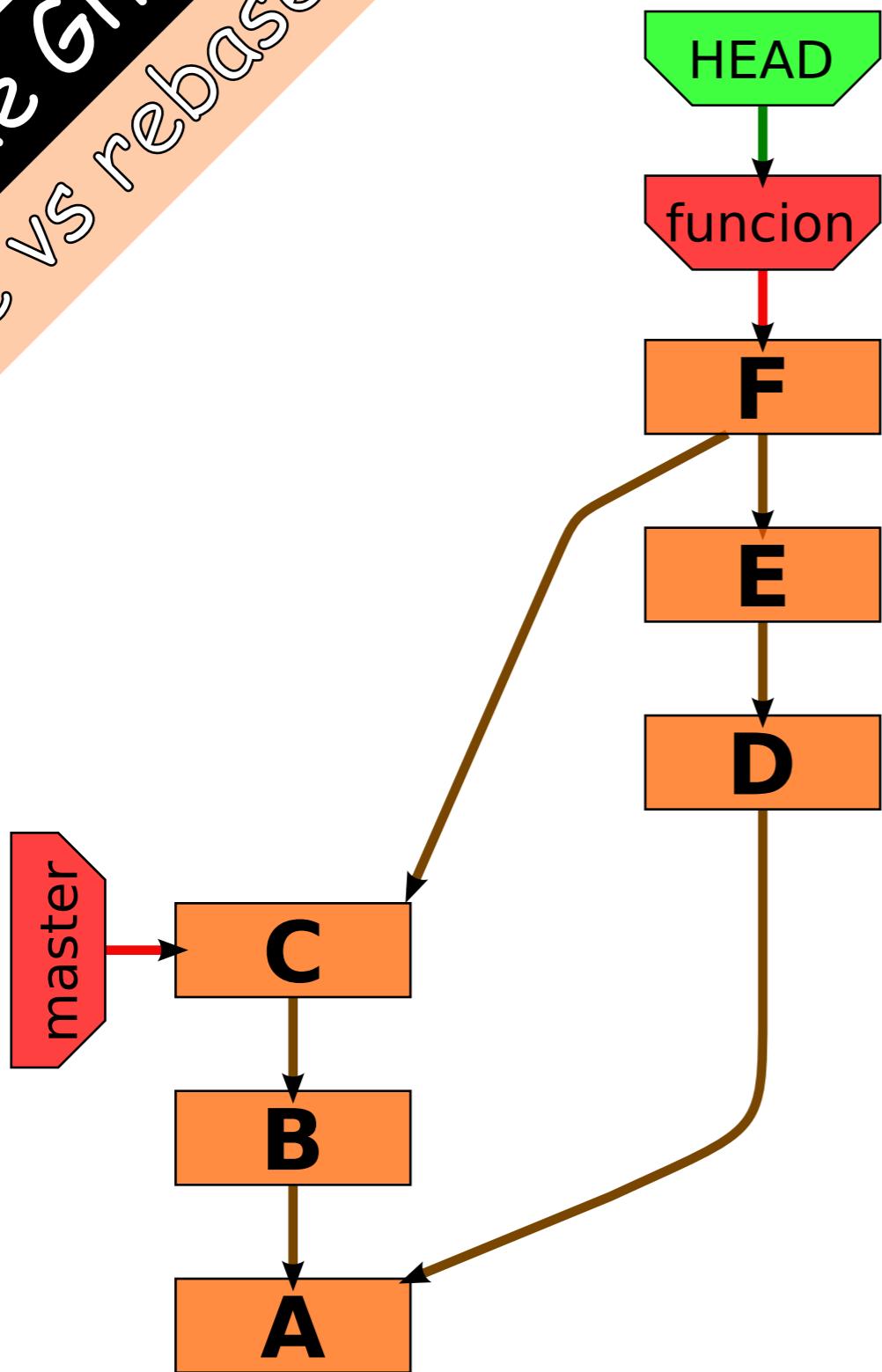
## merge vs rebase



git rebase master

# Uso de Git

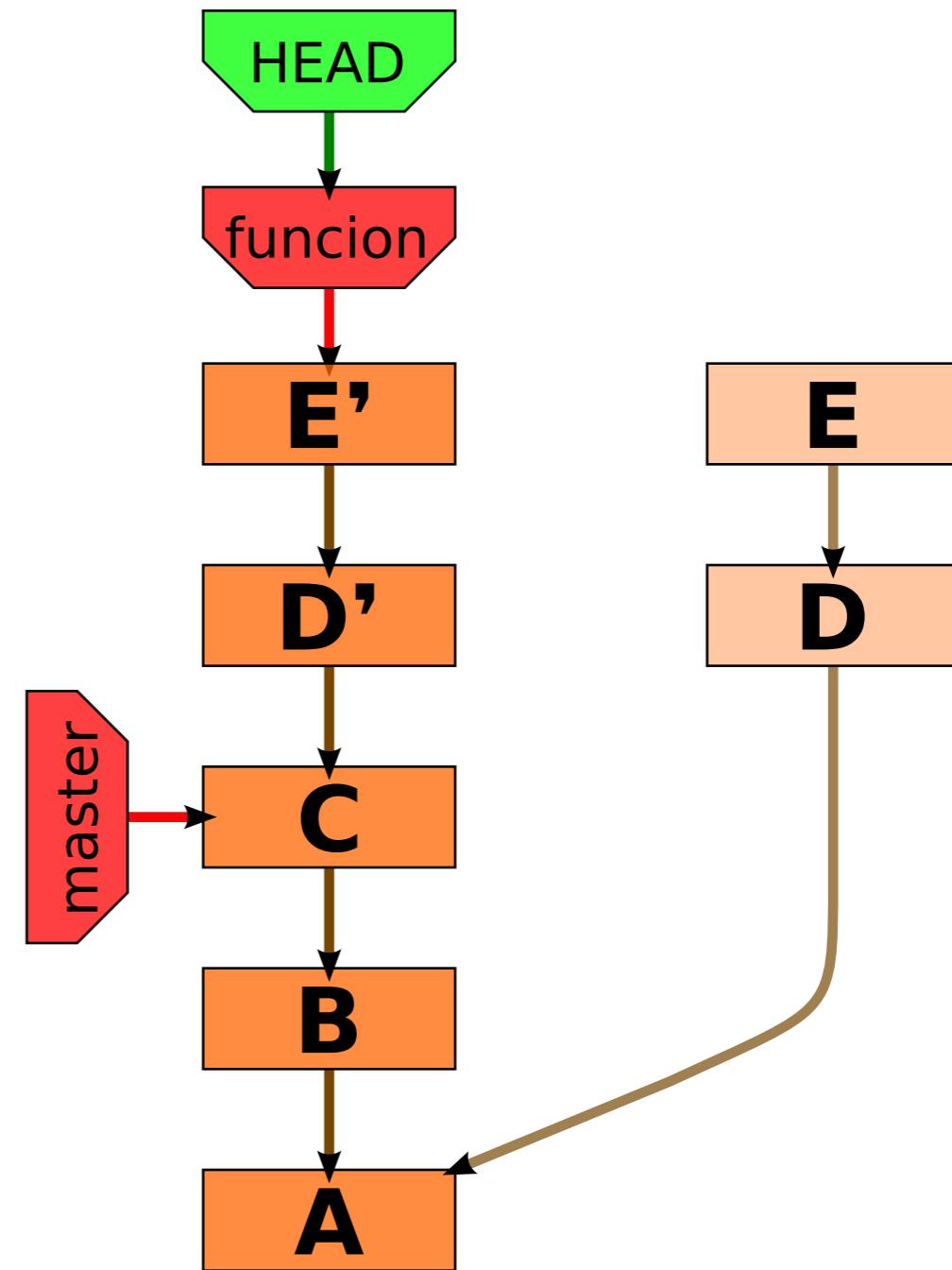
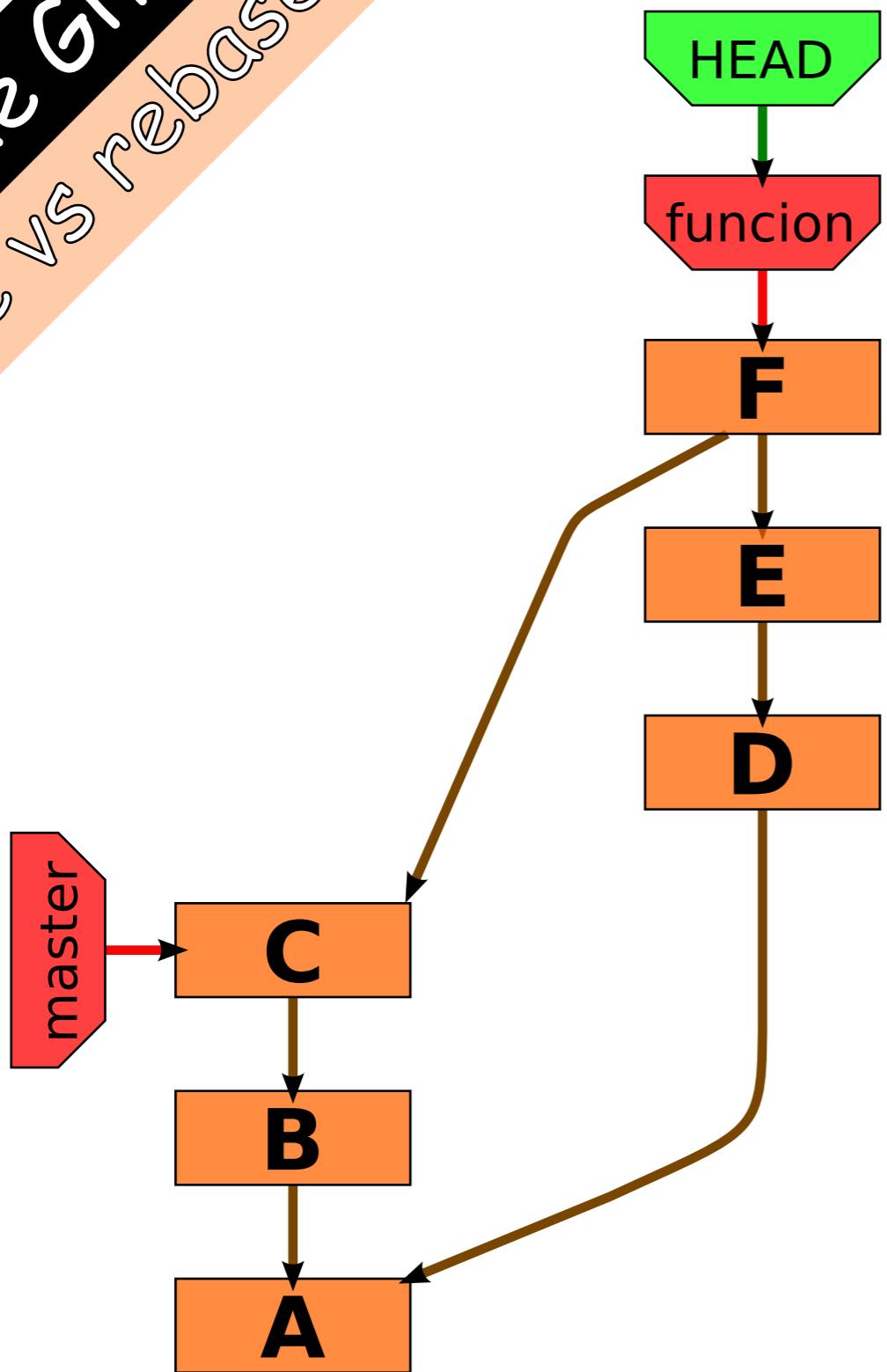
## merge vs rebase



git rebase master

# Uso de Git

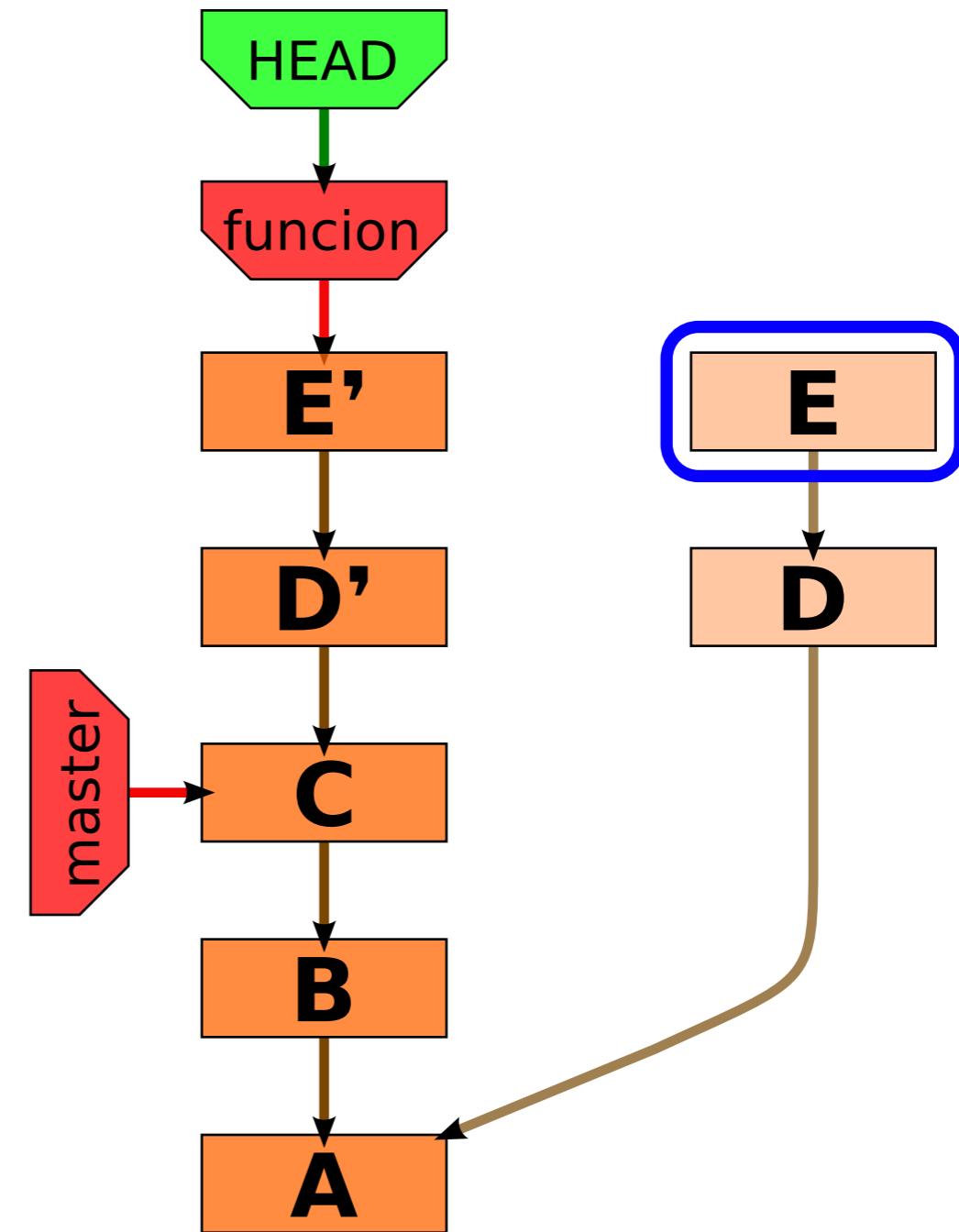
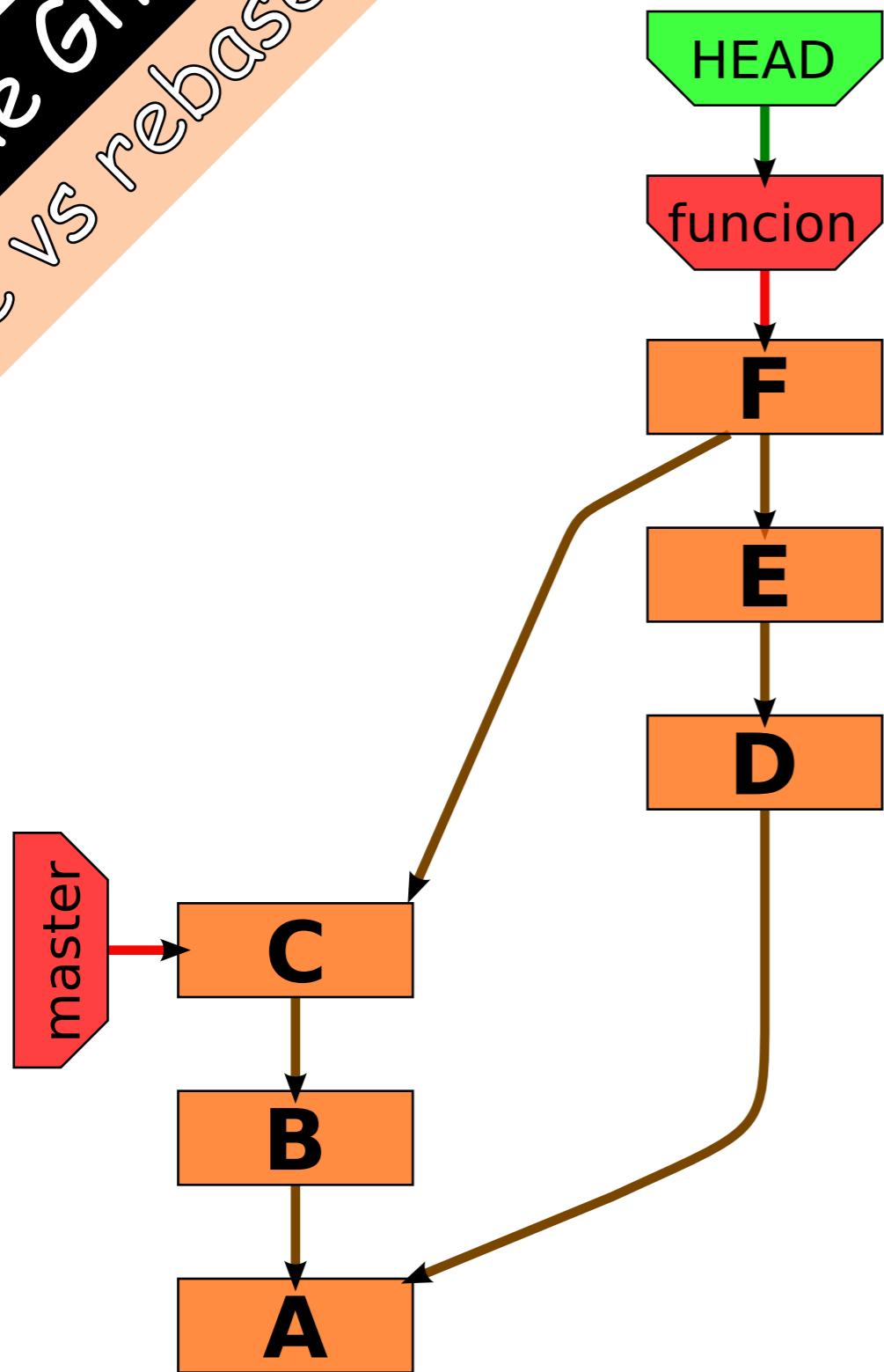
## merge vs rebase



git rebase master

# Uso de Git

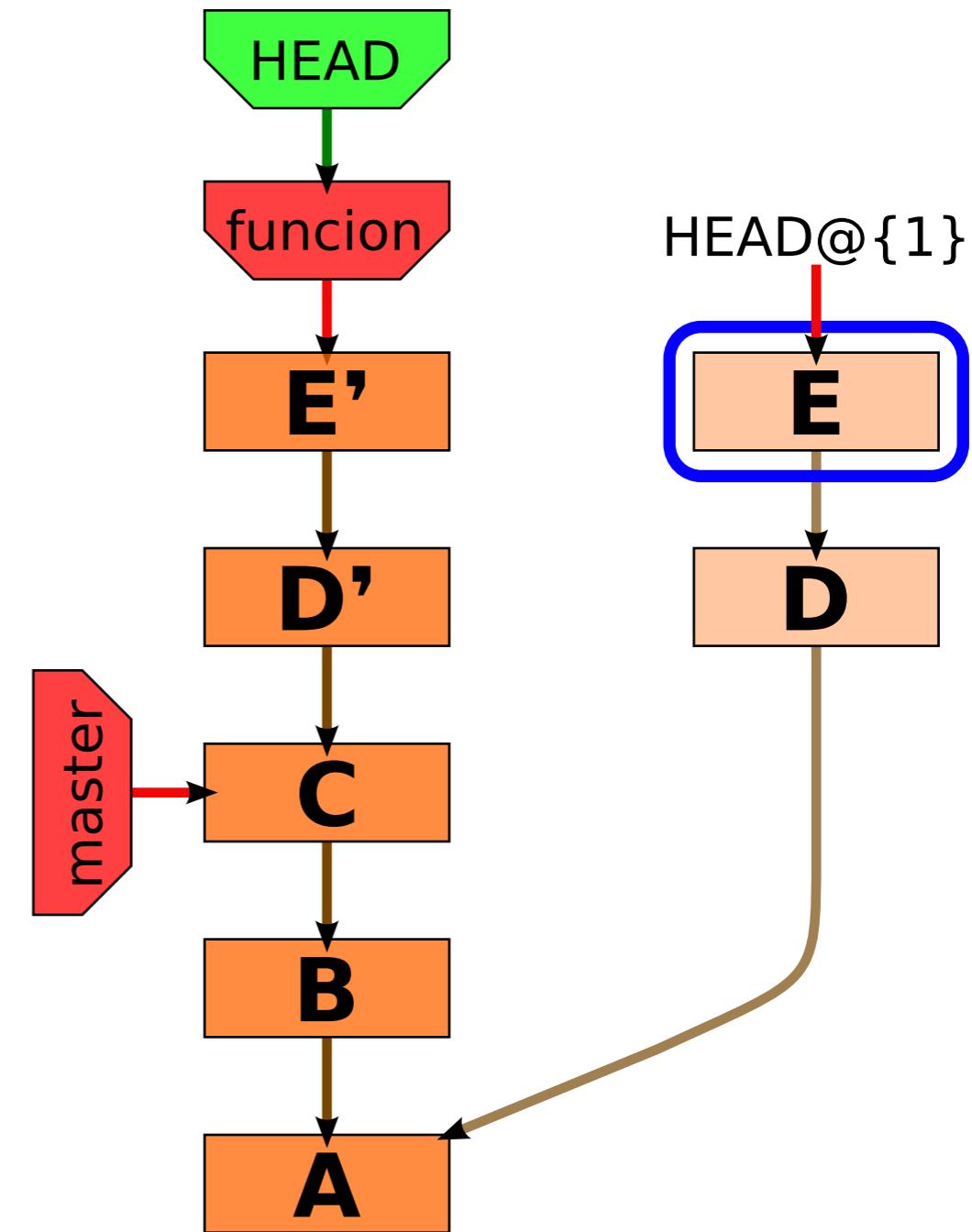
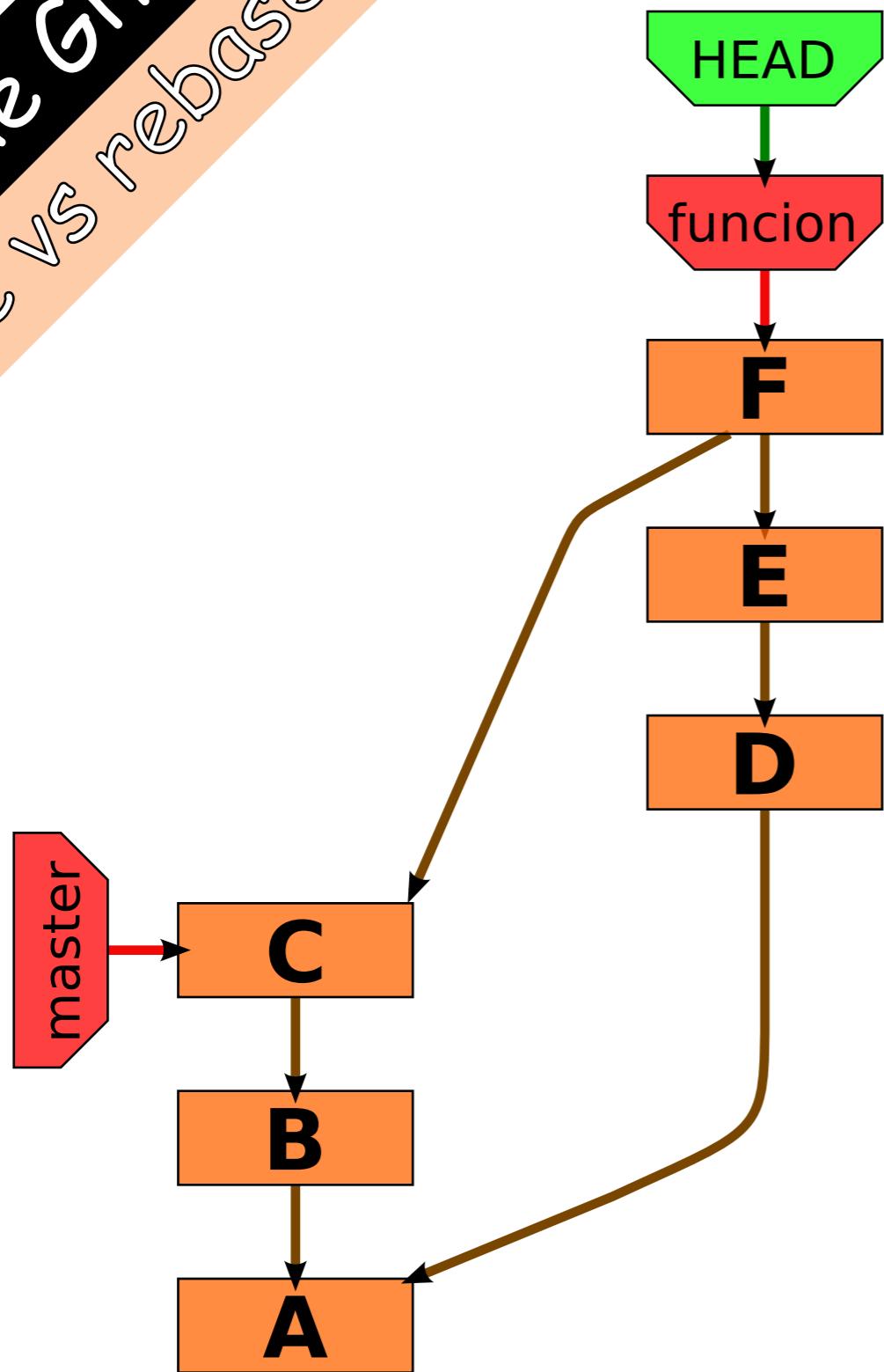
## merge vs rebase



git rebase master

# Uso de Git

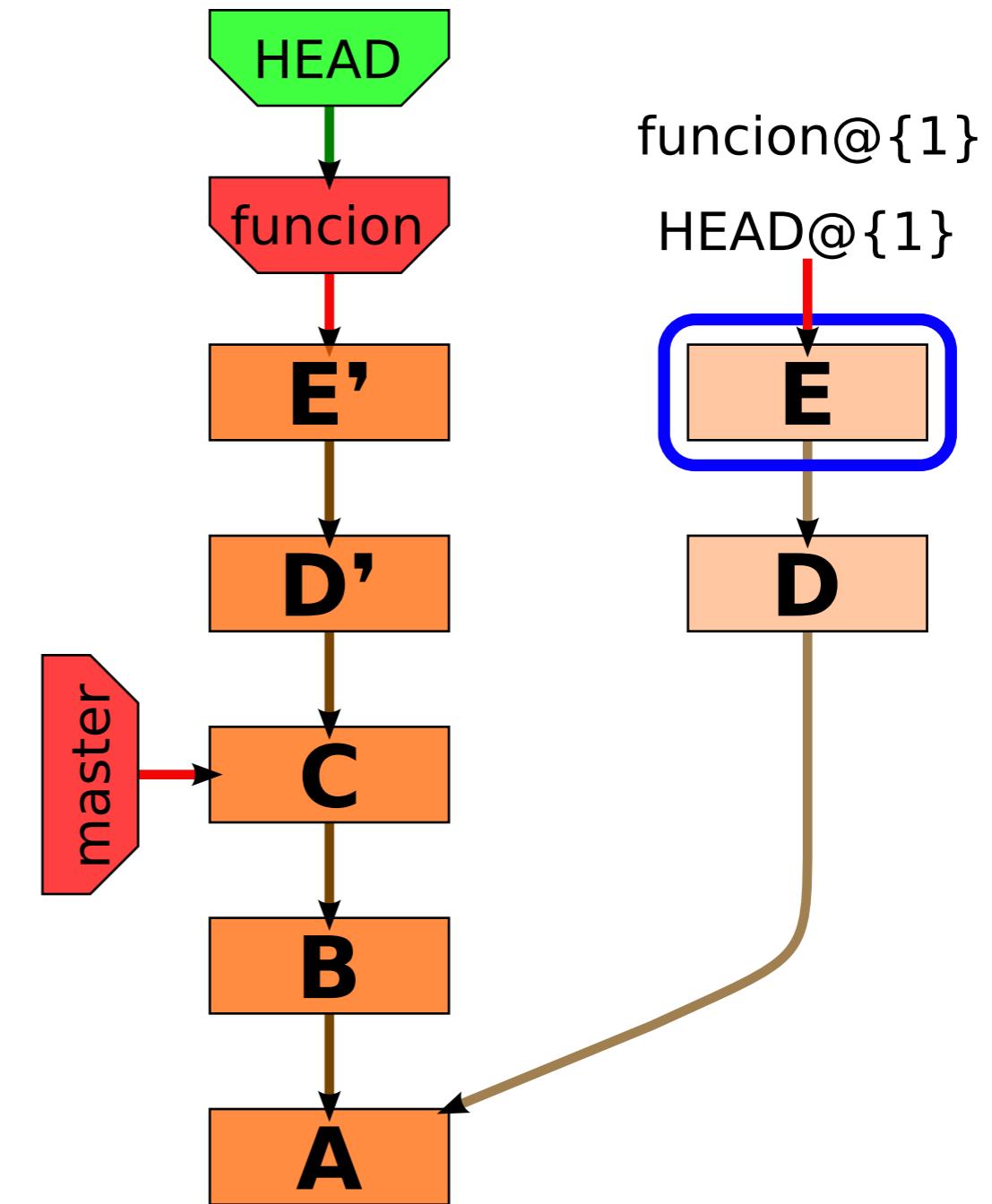
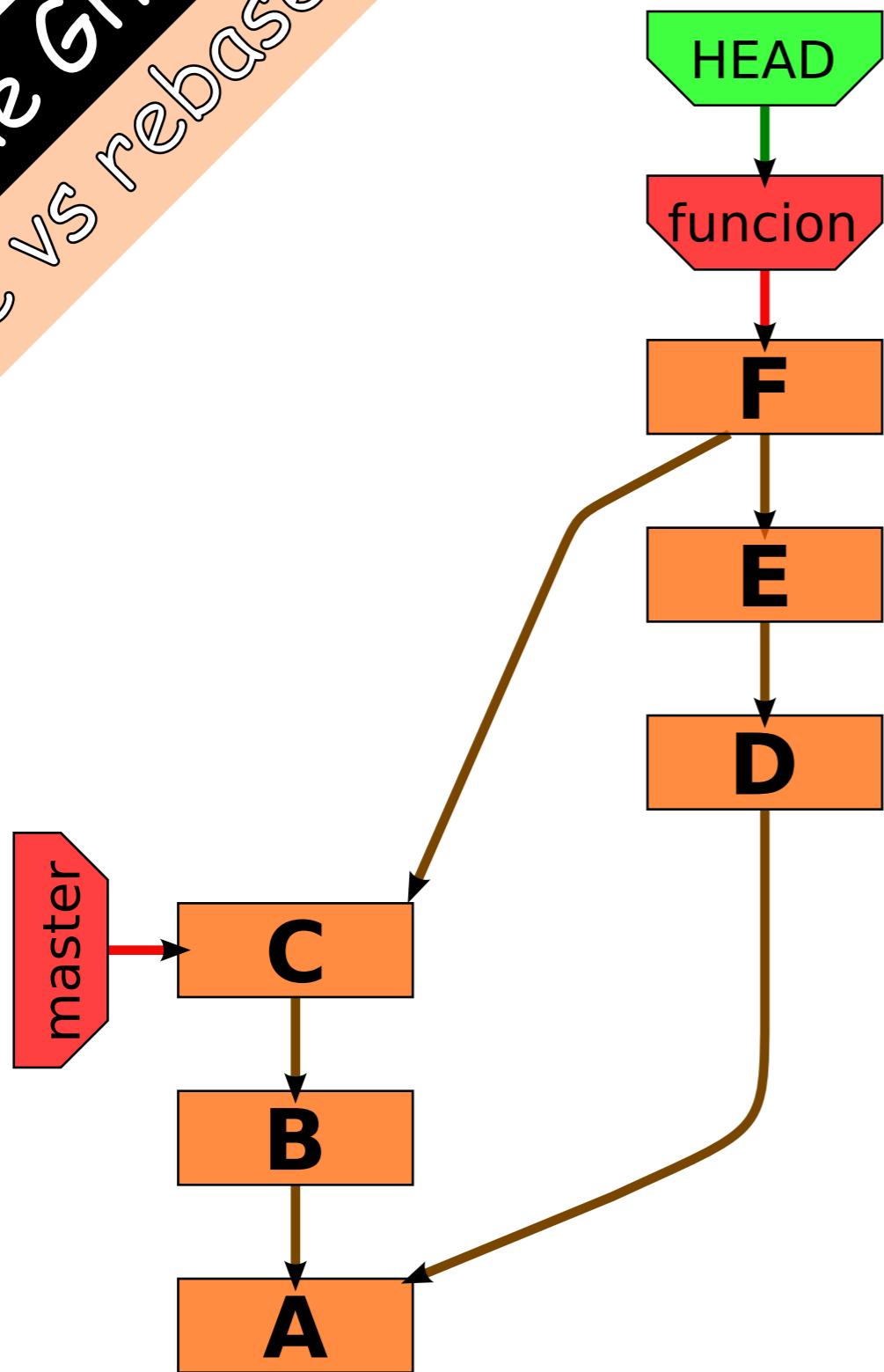
## merge vs rebase



git rebase master

# Uso de Git

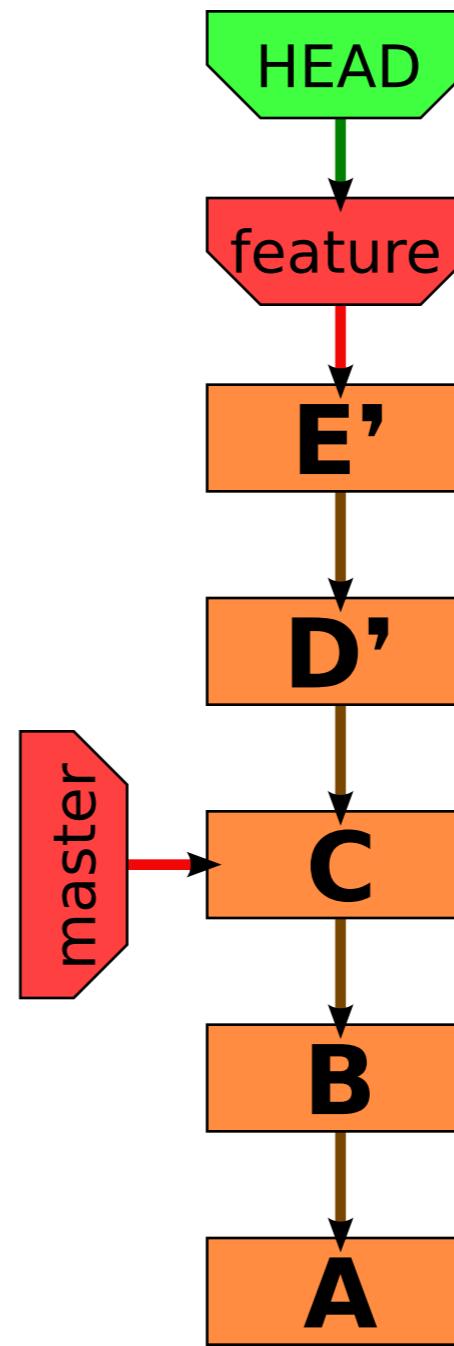
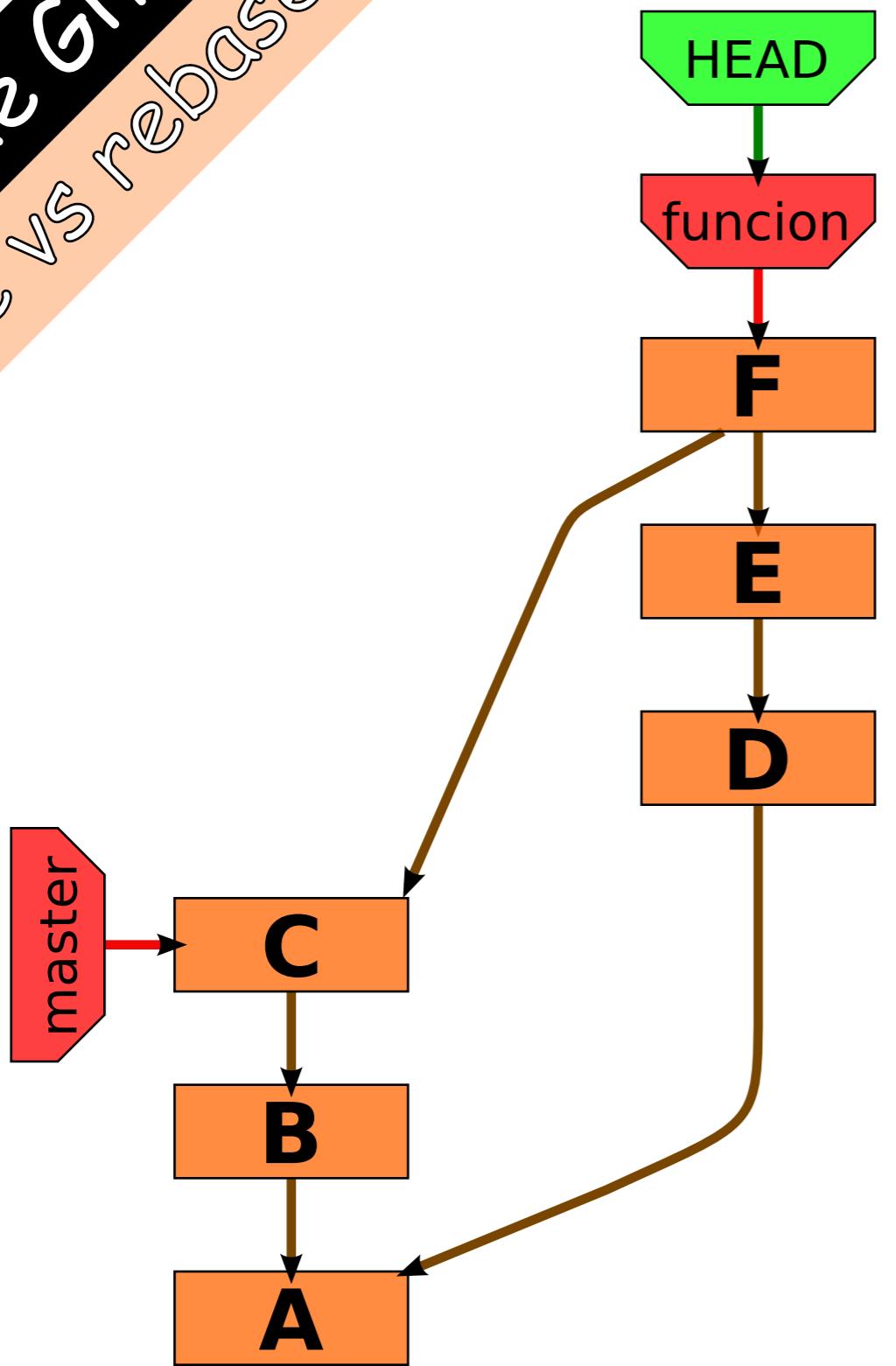
## merge vs rebase



git rebase master

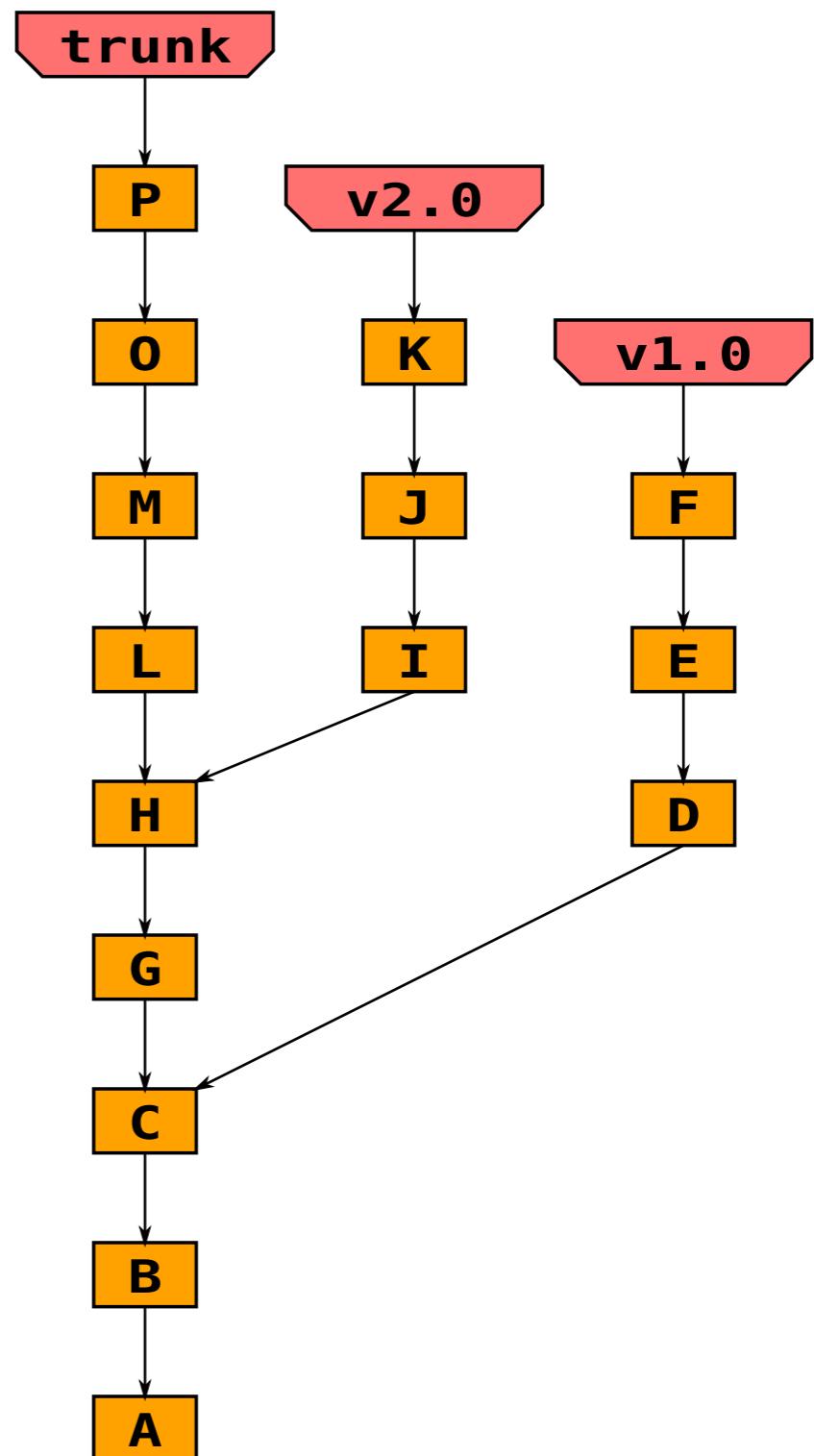
# Uso de Git

## merge vs rebase



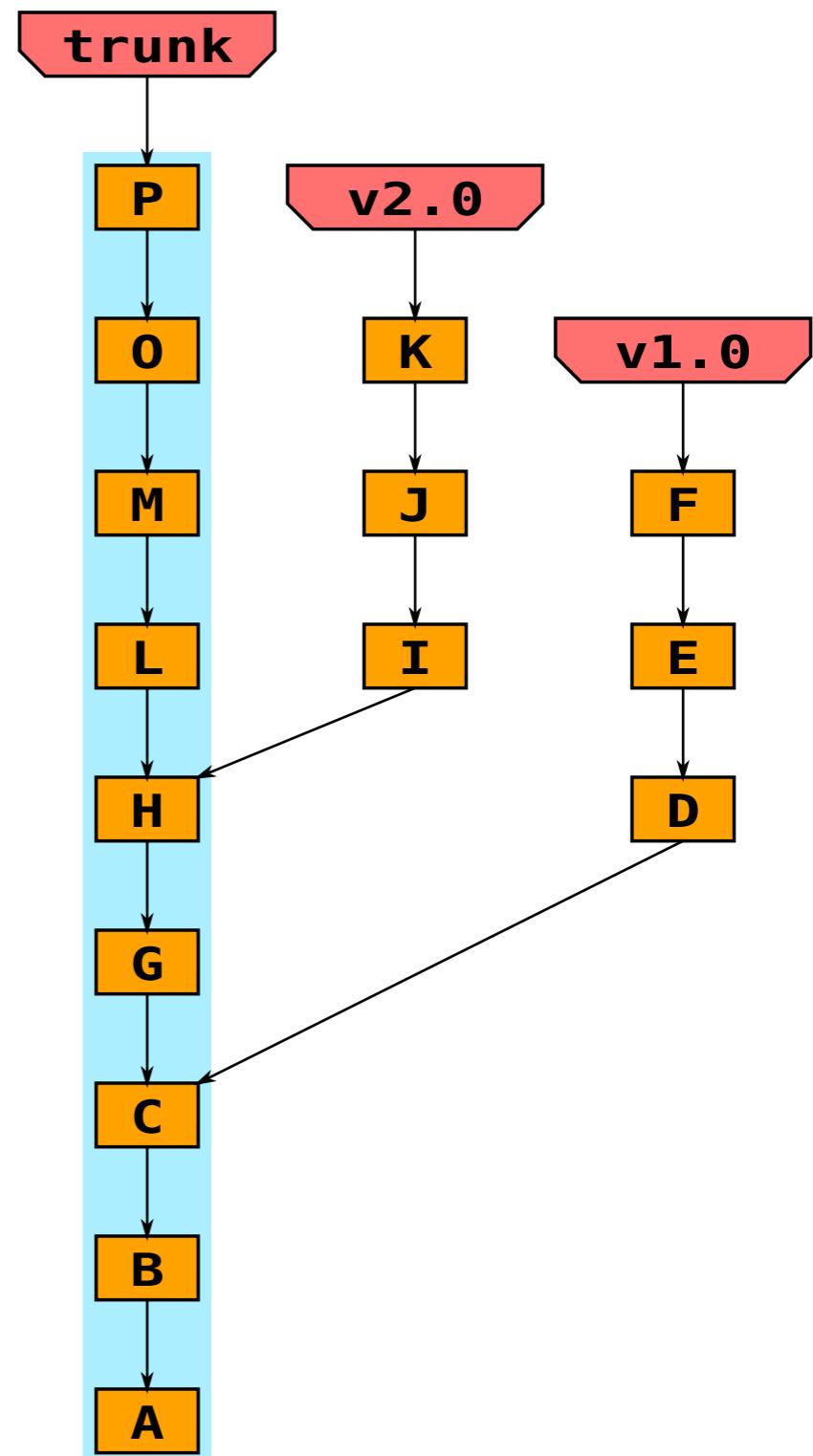
# Modelos de ramas

# Tradicional



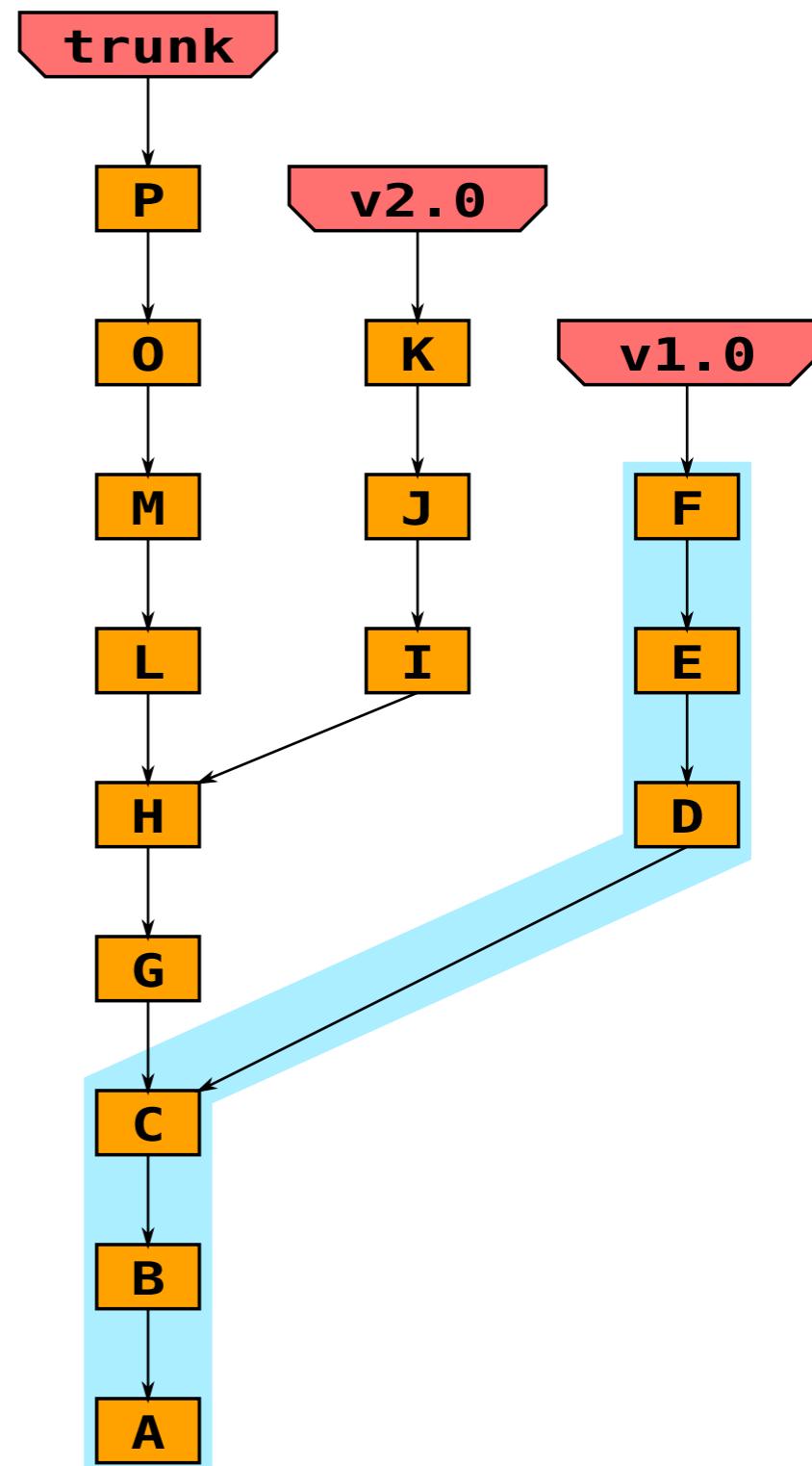
# Tradicional

Línea principal de desarrollo



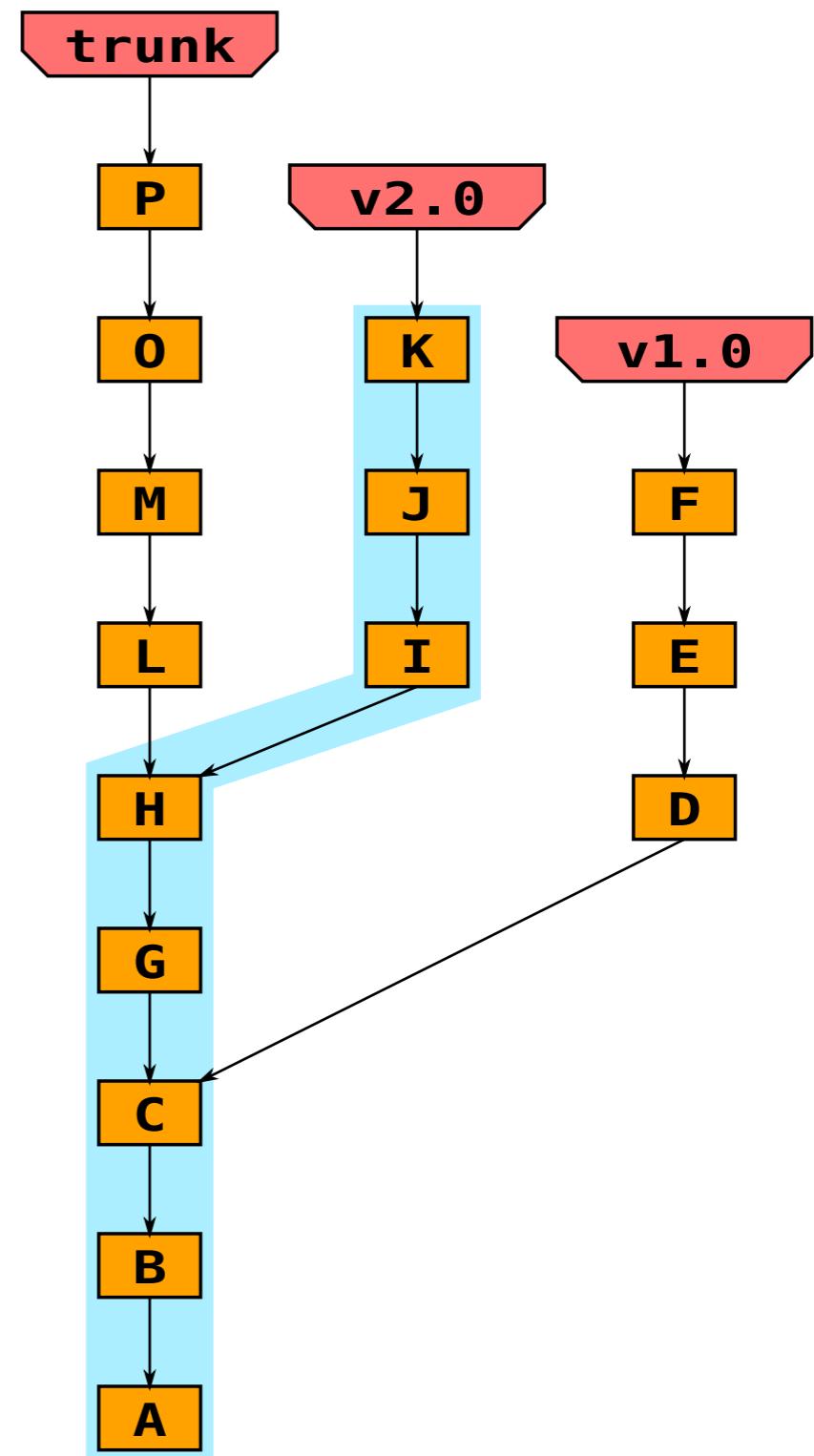
# Tradicional

versiones históricas



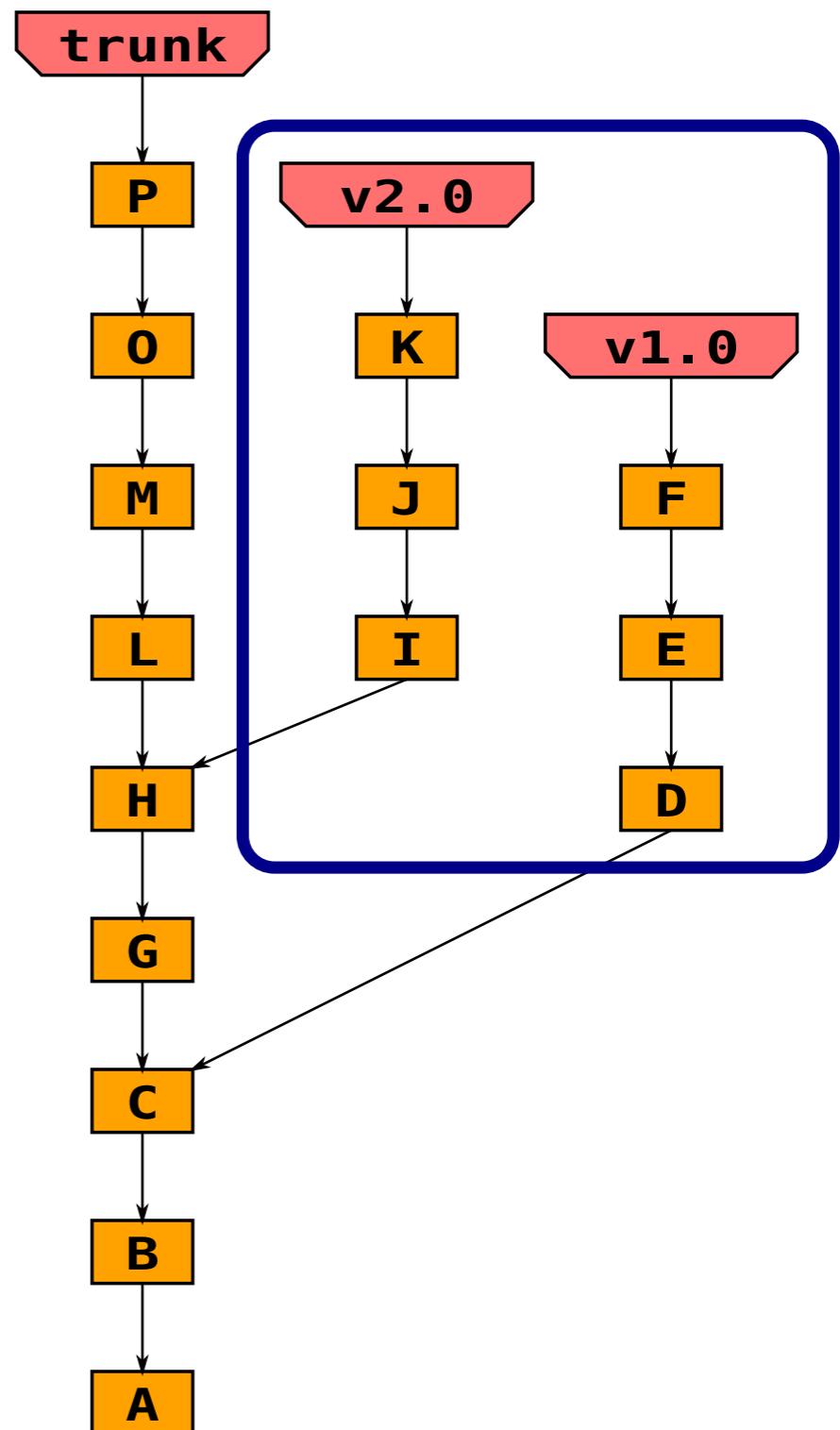
# Tradicional

versiones históricas



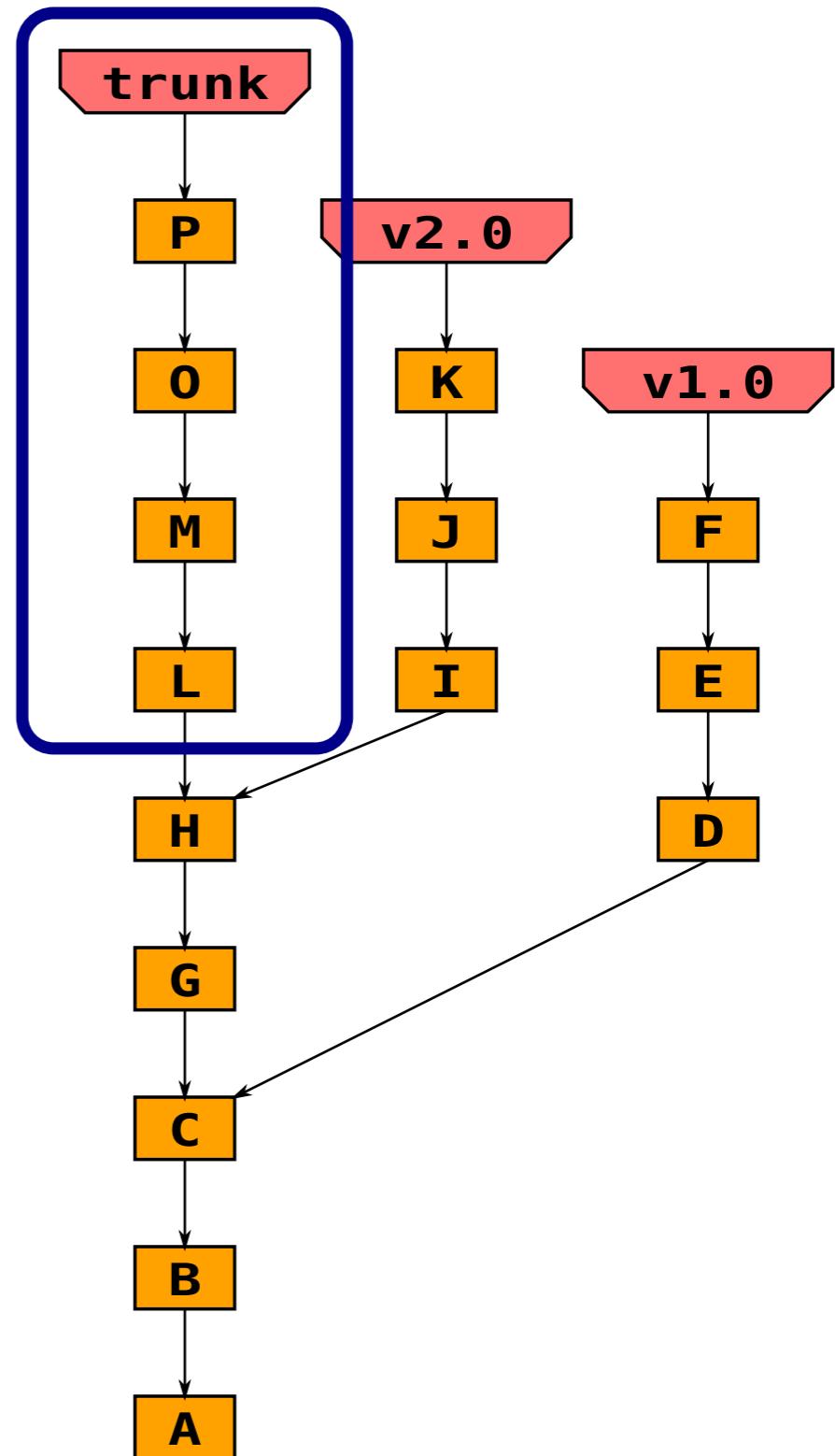
# Tradicional

sólo arreglos de defectos



# Tradicional

añadidos y correcciones



# Ramas temáticas

- una rama por tema

# Ramas temáticas

- una rama por tema
- se crean revisiones a menudo

# Ramas temáticas

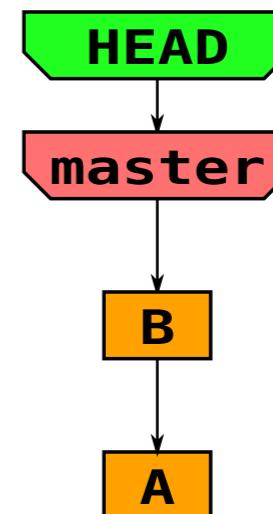
- una rama por tema
- se crean revisiones a menudo
- duran poco

# Ramas temáticas

- una rama por tema
- se crean revisiones a menudo
- duran poco
- el trabajo realizado se reúne en **master**

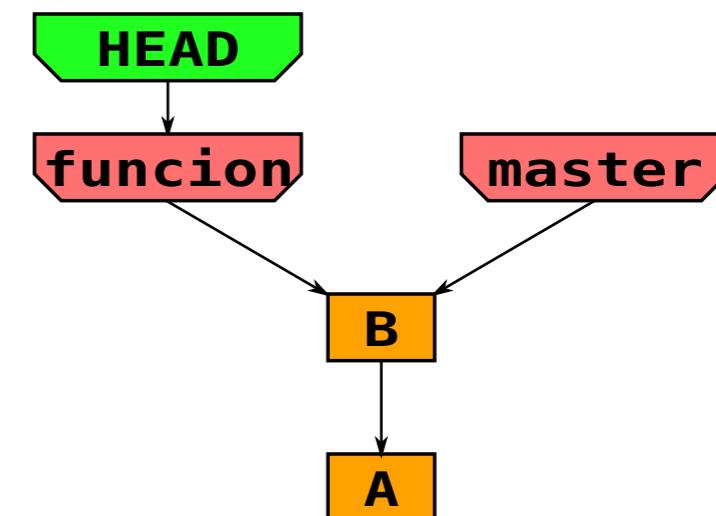
# Ramas temáticas

un repositorio existente



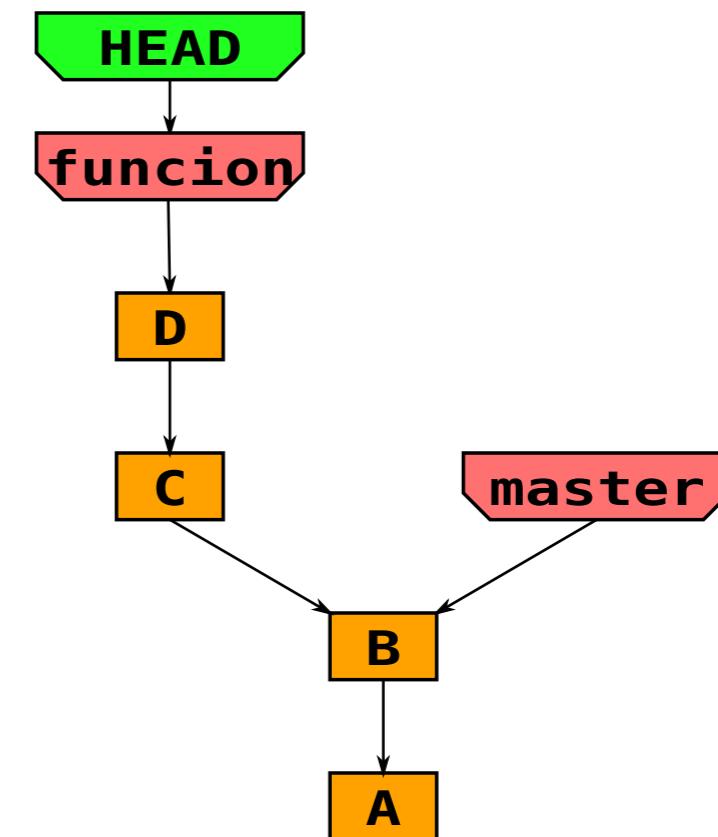
# Ramas temáticas

ramificamos para una **función**



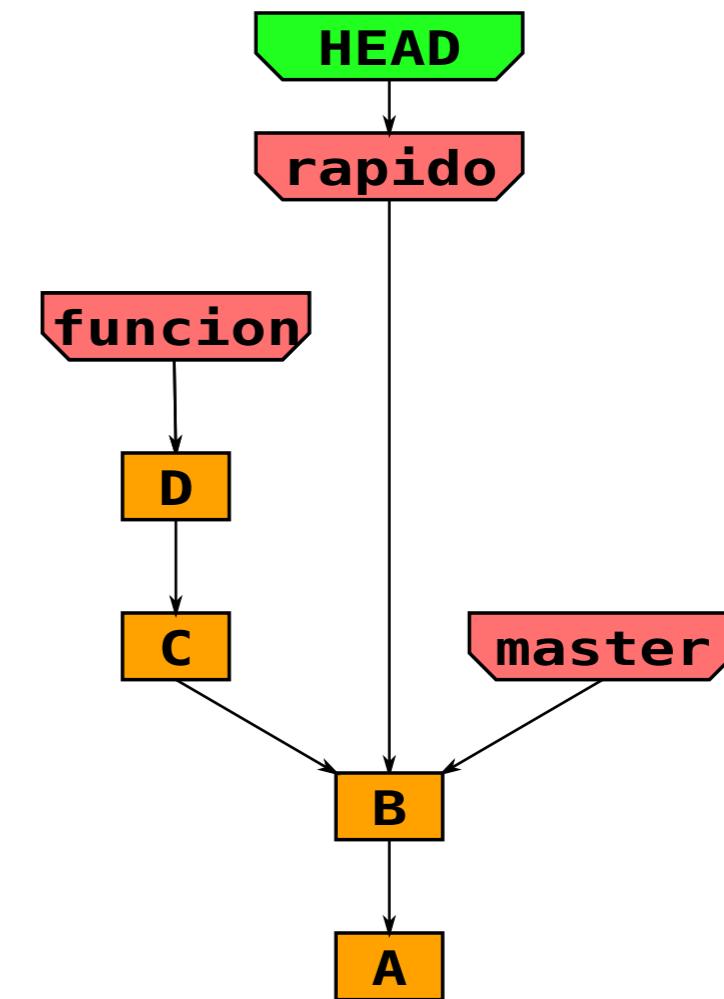
# Ramas temáticas

desarrollamos la función



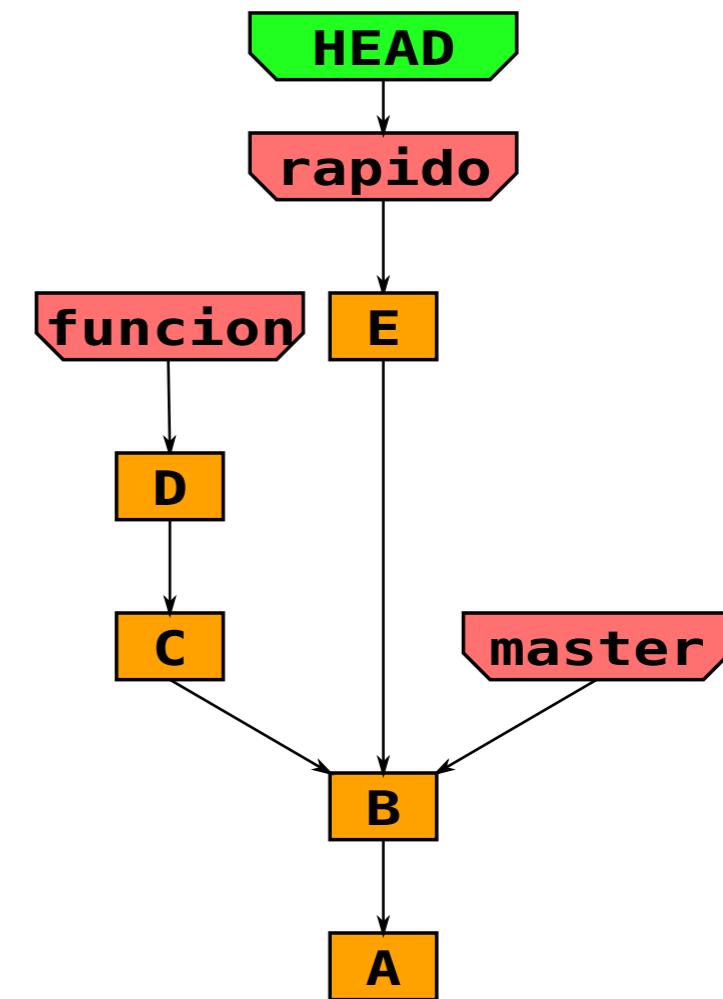
# Ramas temáticas

vemos un fallo de arreglo rápido



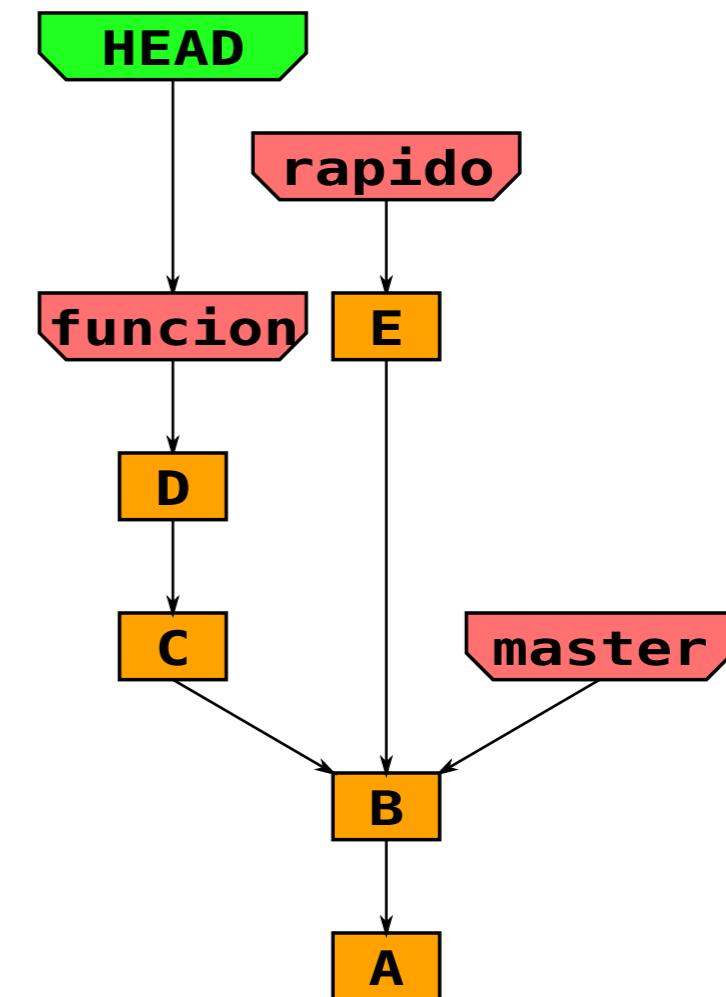
# Ramas temáticas

arreglamos el defecto rápidamente



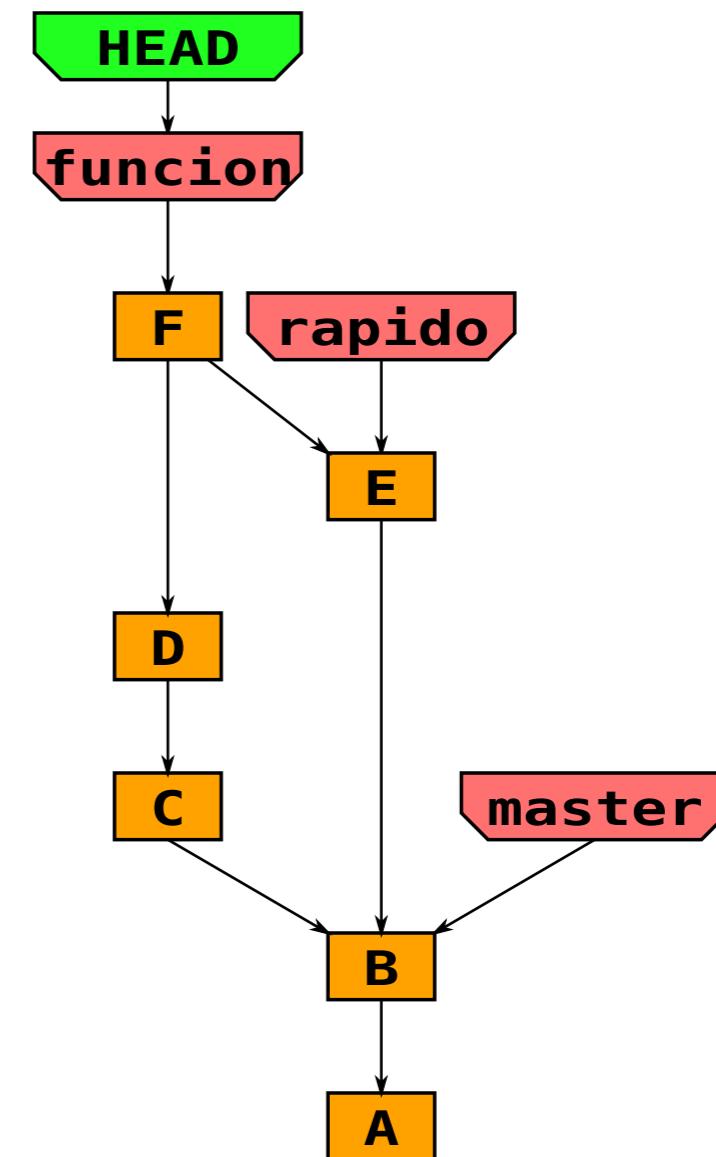
# Ramas temáticas

volvemos a la nueva función



# Ramas temáticas

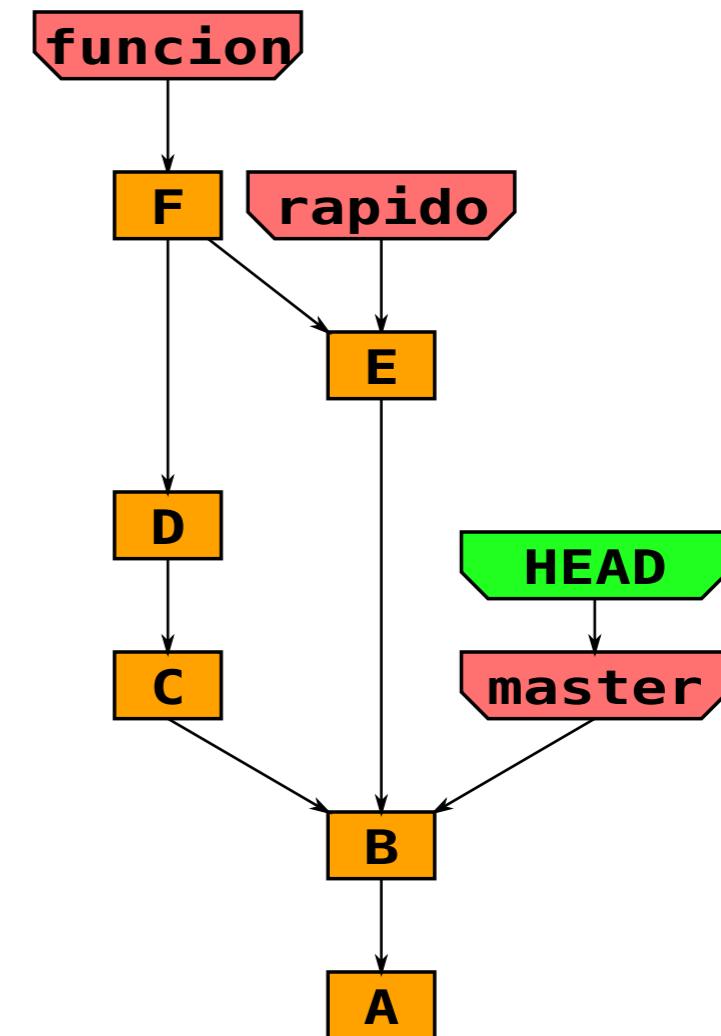
reunimos el arreglo rápido



# Ramas temáticas

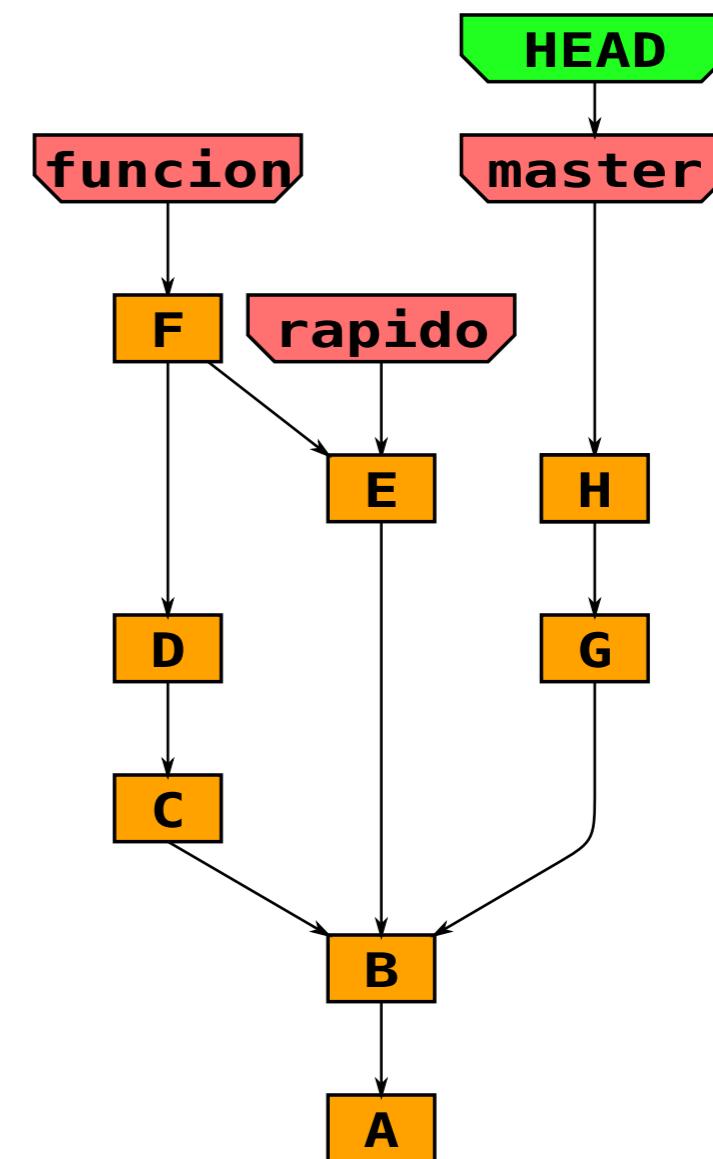
*iUps!*

Tenemos que arreglar una  
*caída originada río arriba*



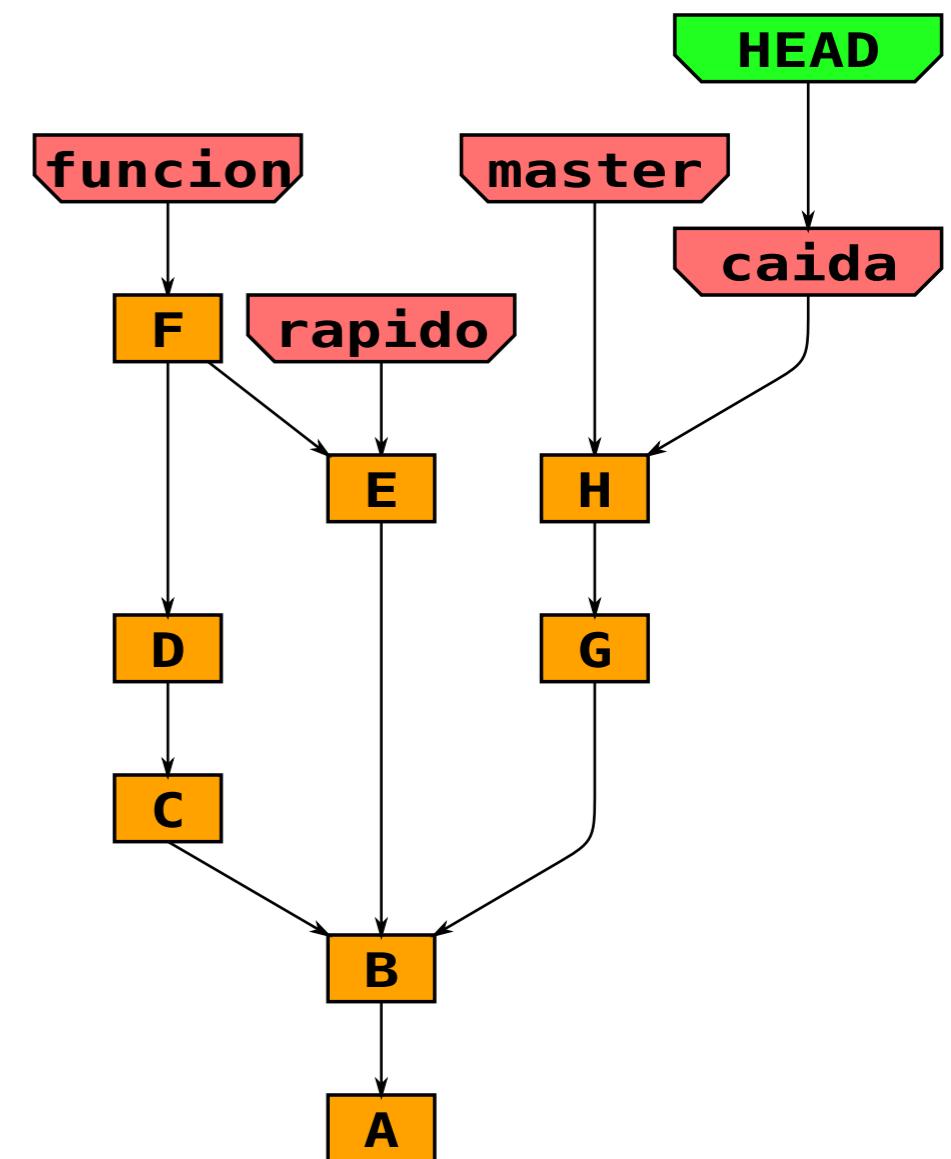
# Ramas temáticas

primero, actualizamos **master** con el código de la página oficial



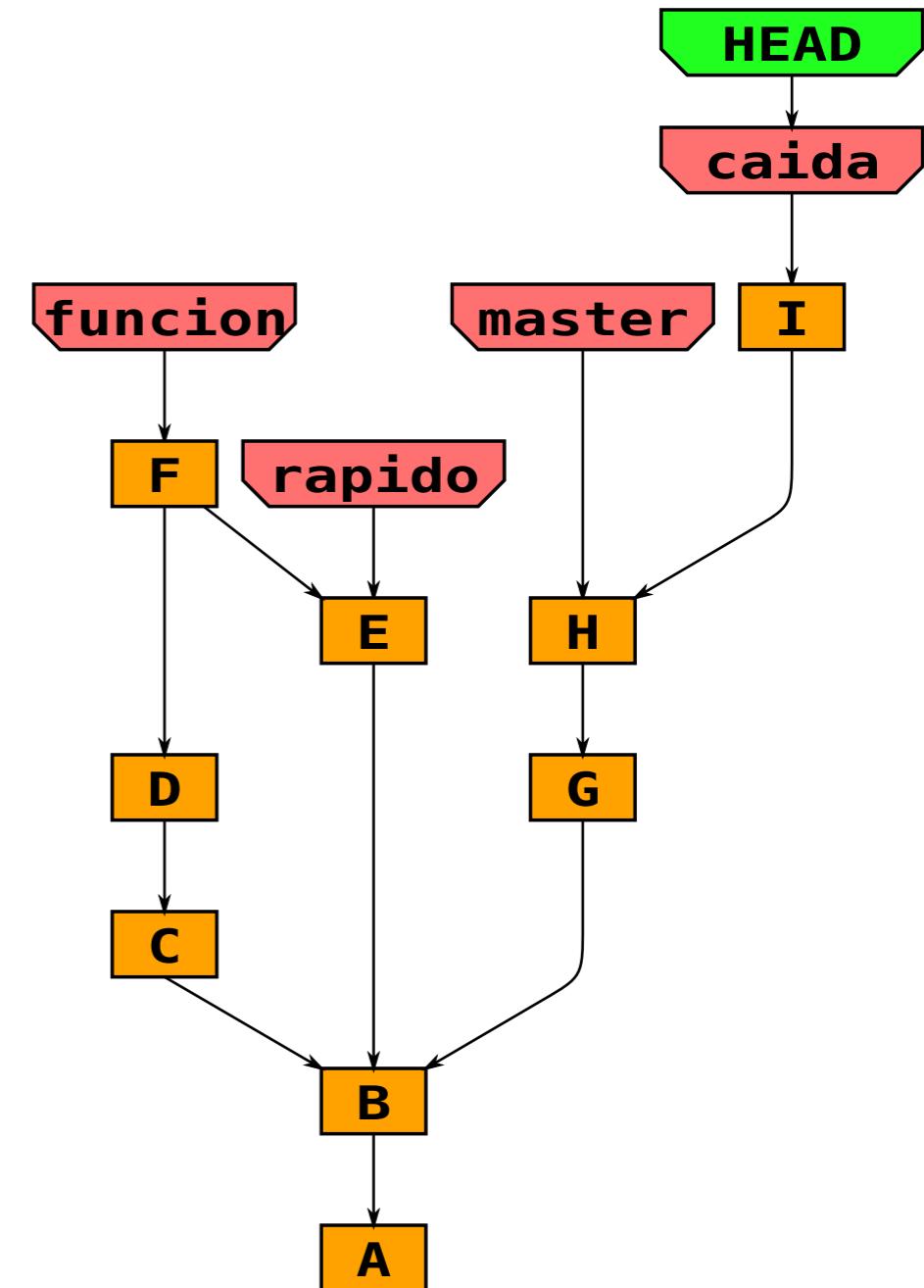
# Ramas temáticas

creamos una rama para arreglar  
la **caída**



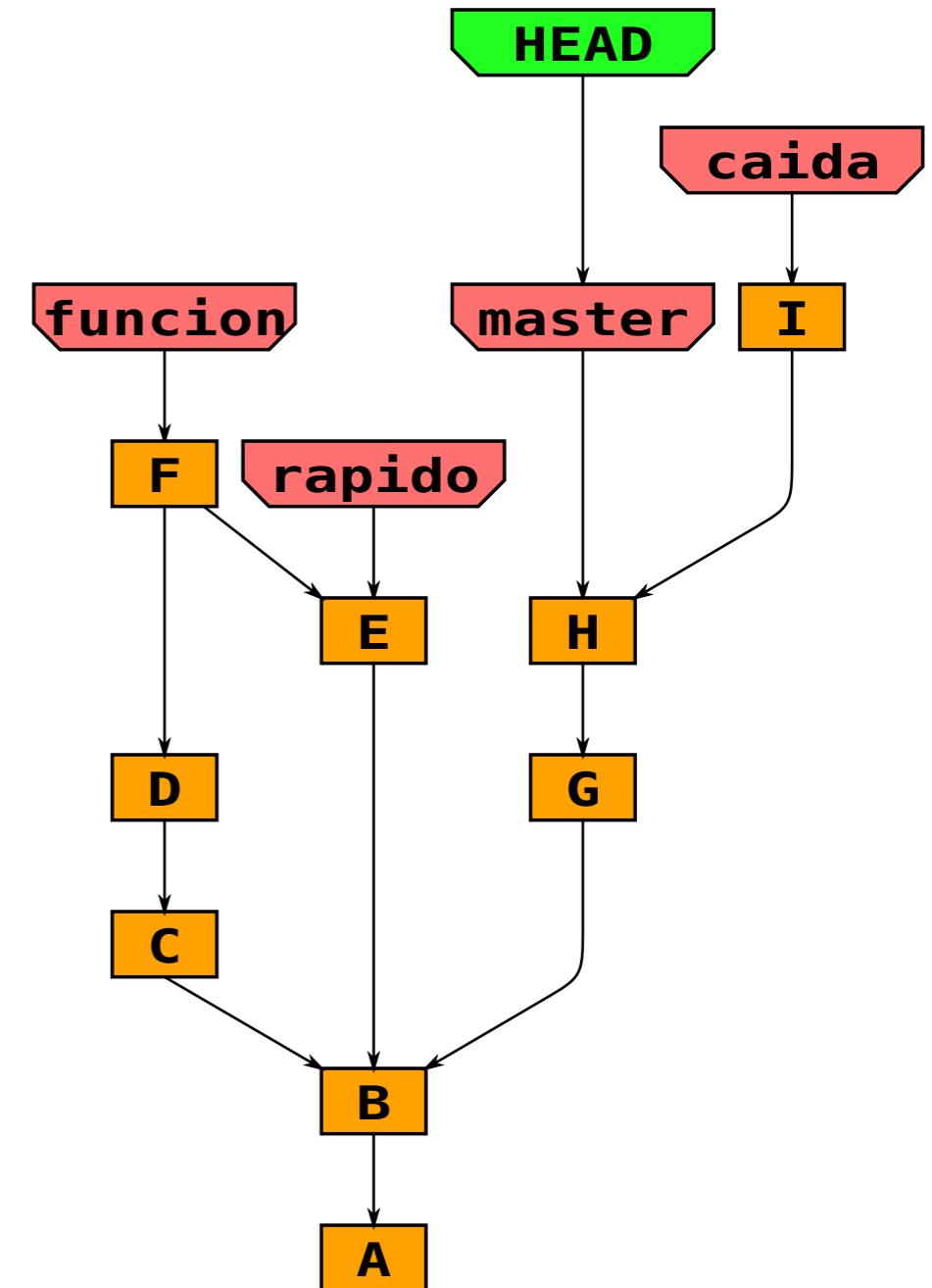
# Ramas temáticas

la **caída** se arregla fácilmente



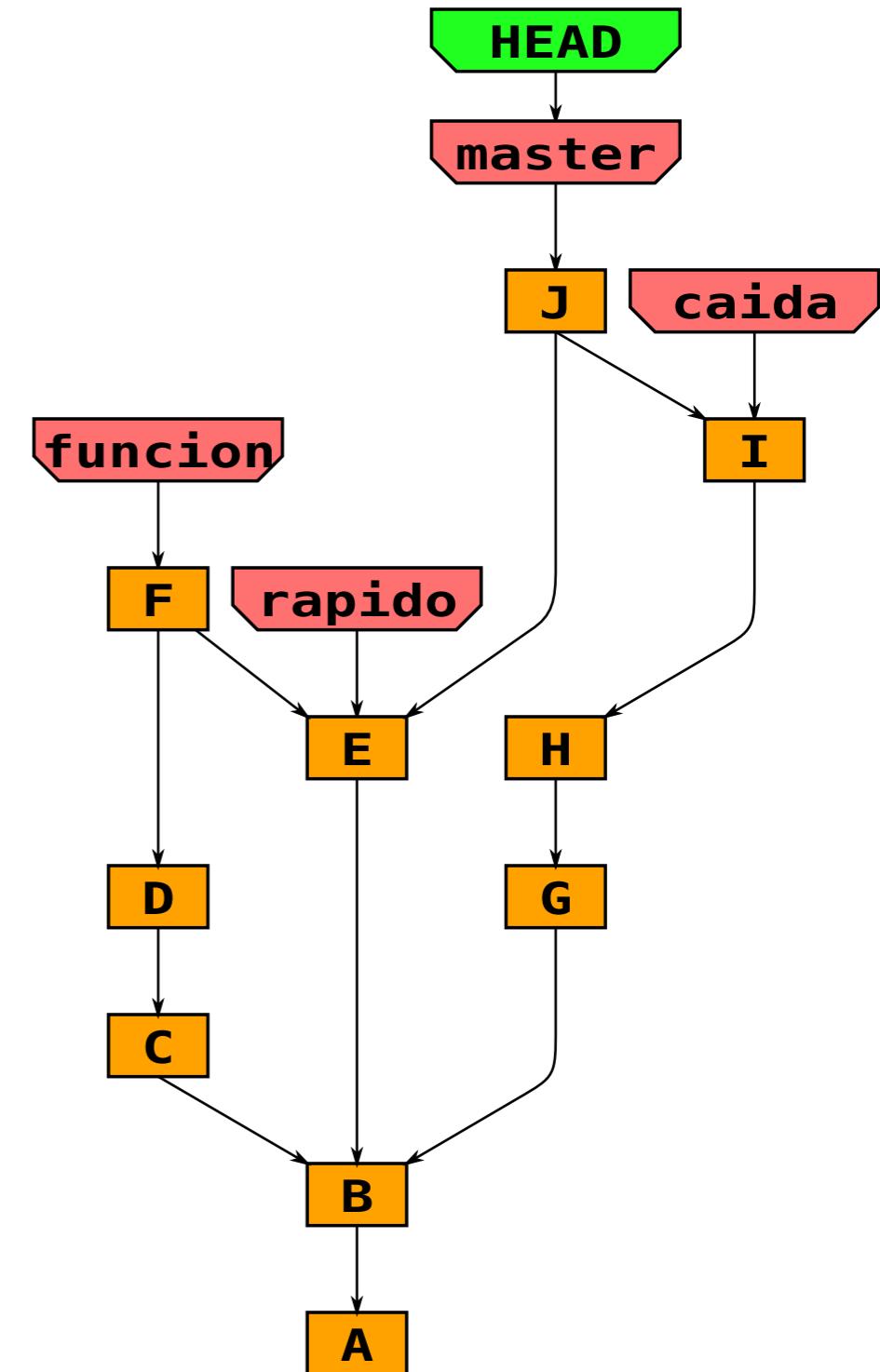
# Ramas temáticas

vamos a integrarlo todo



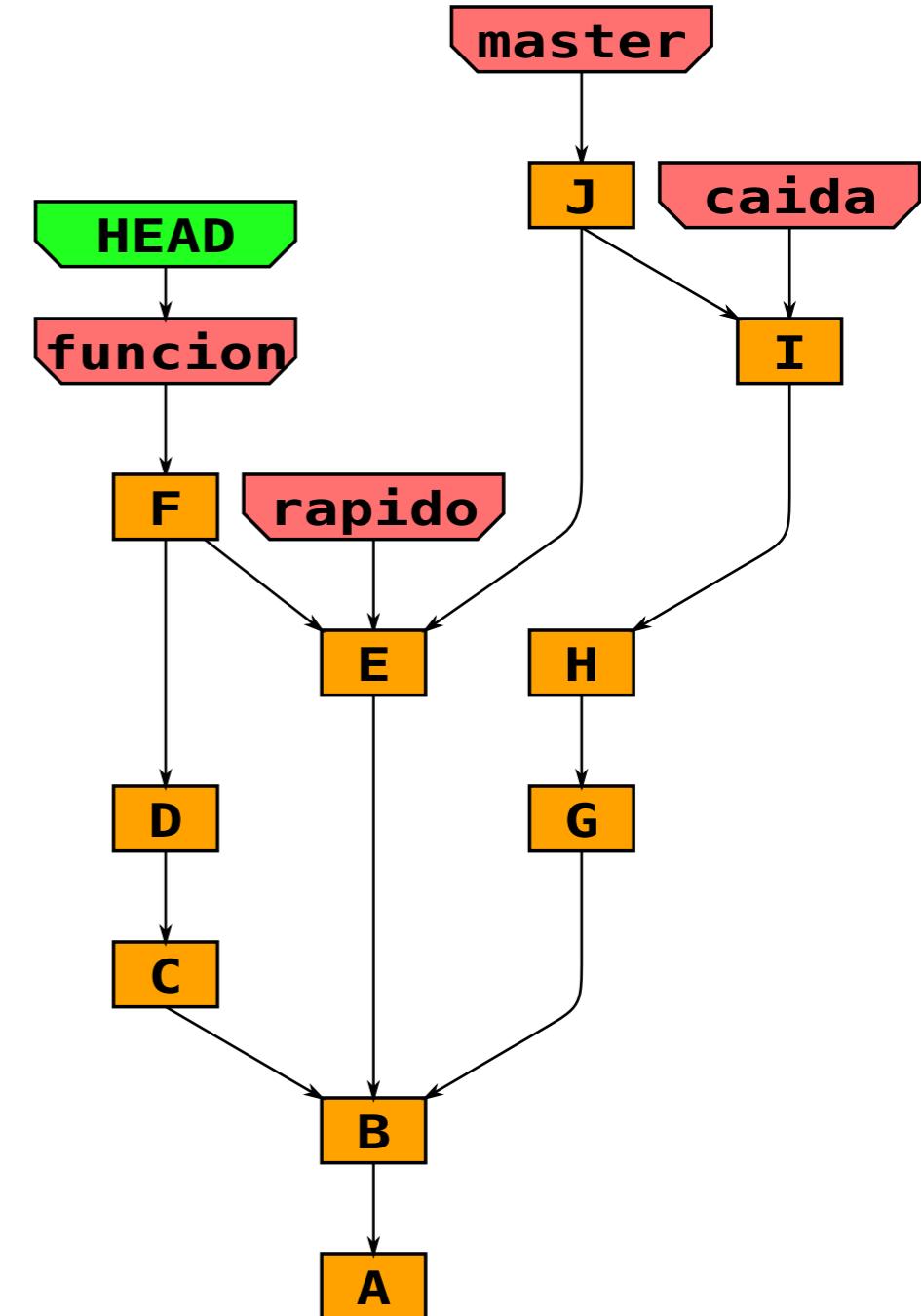
# Ramas temáticas

traemos los arreglos a master



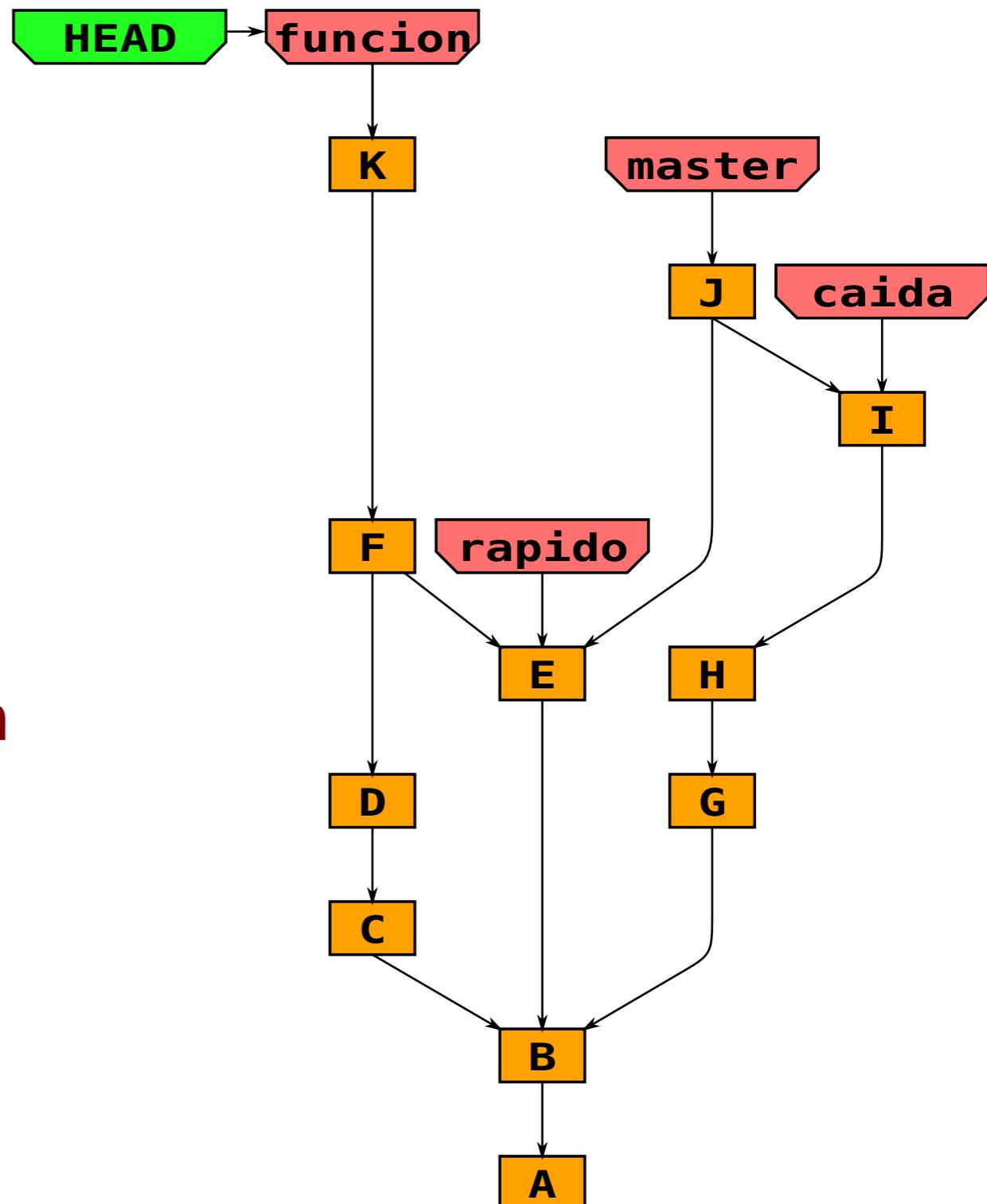
# Ramas temáticas

seguimos con la función



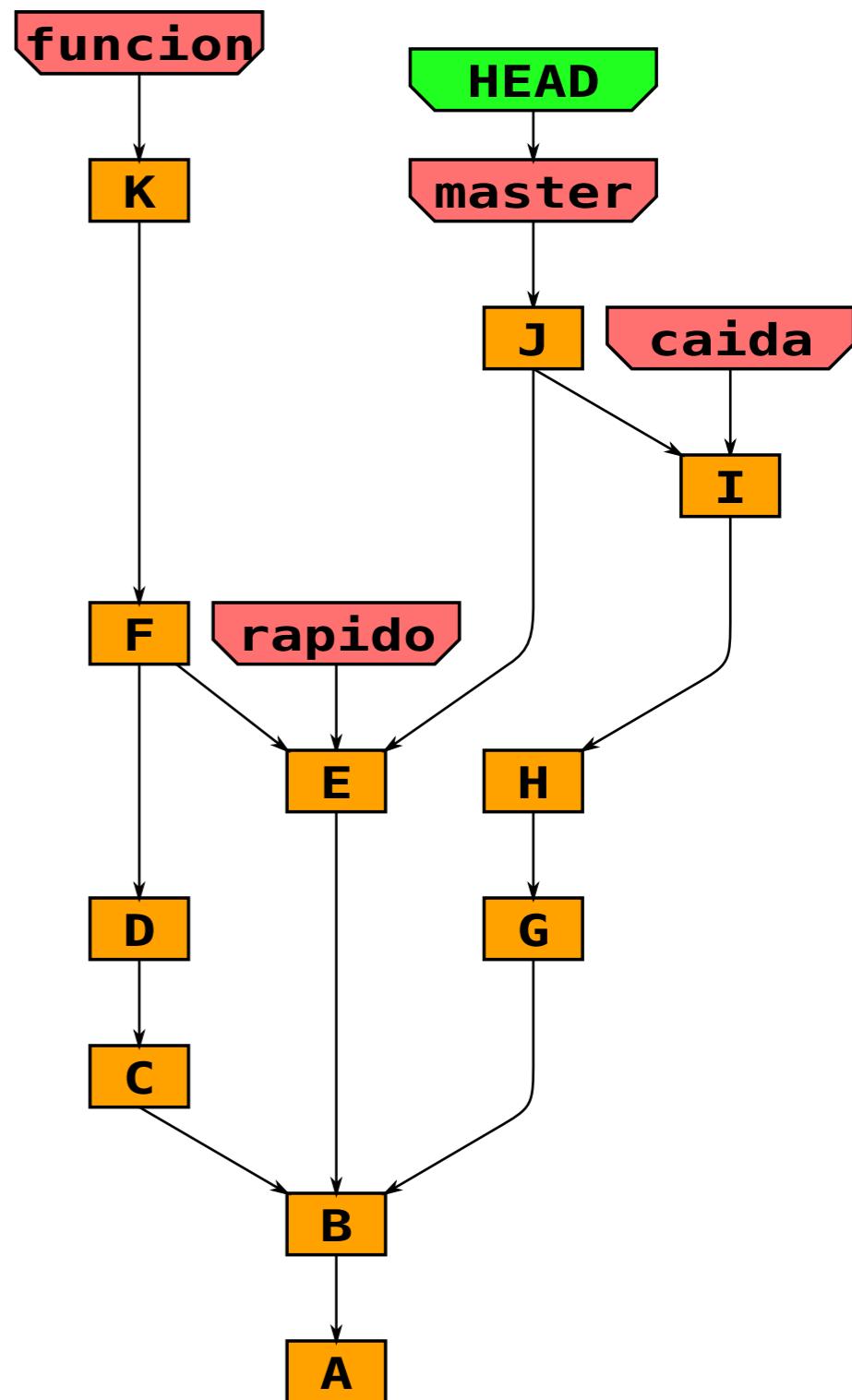
# Ramas temáticas

por fin, se termina la función

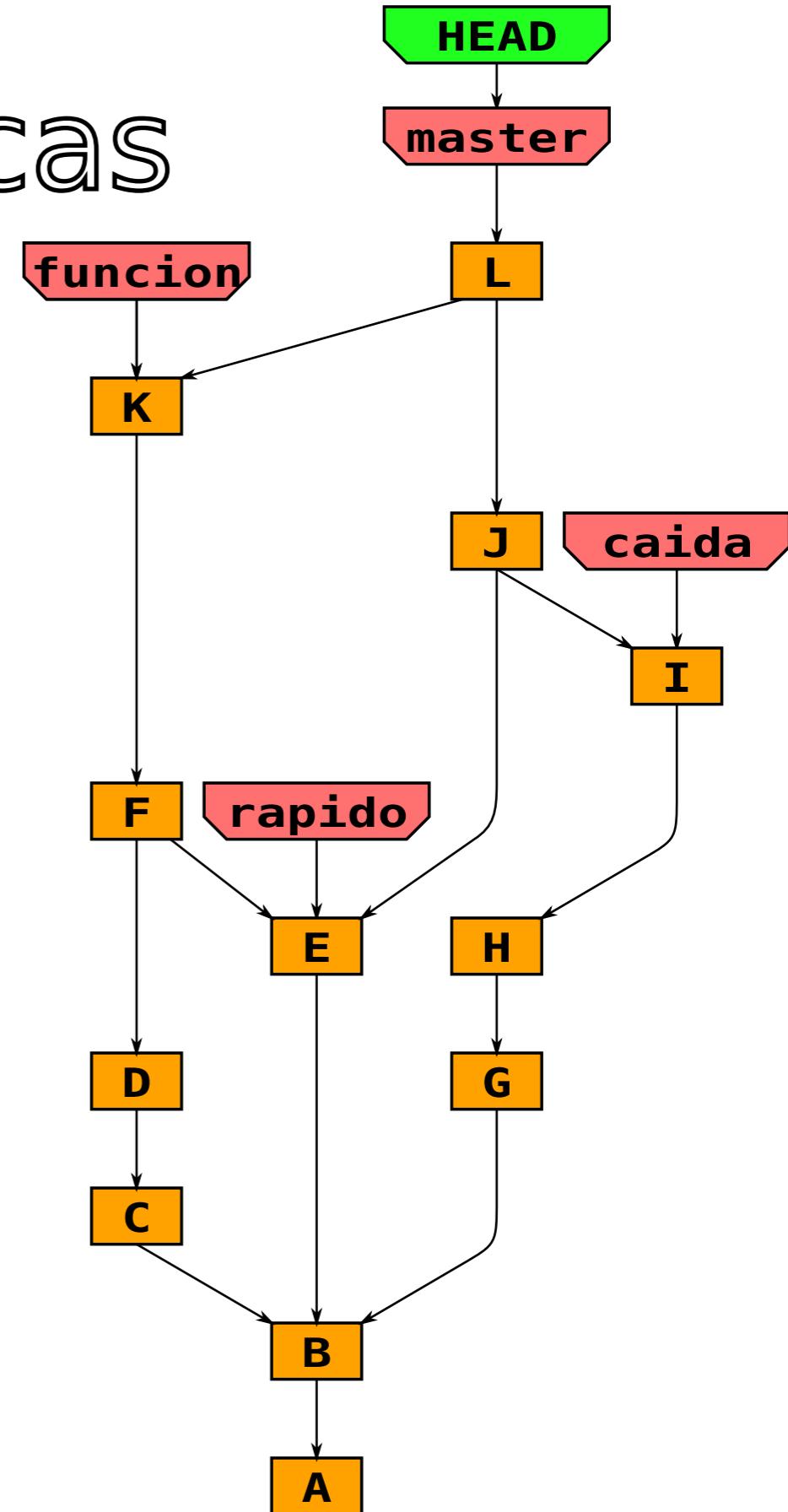


# Ramas temáticas

volvemos a master

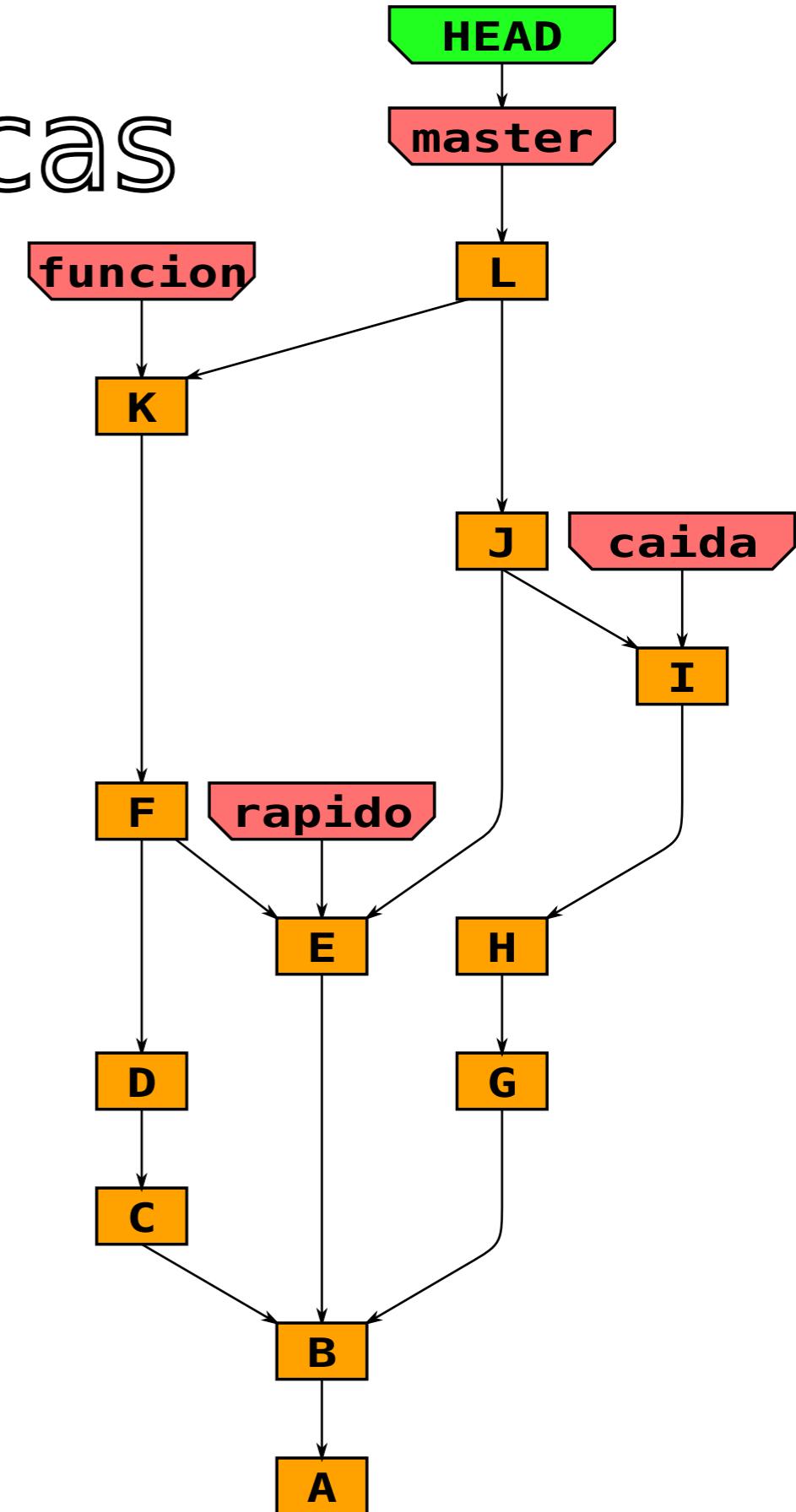


# Ramas temáticas



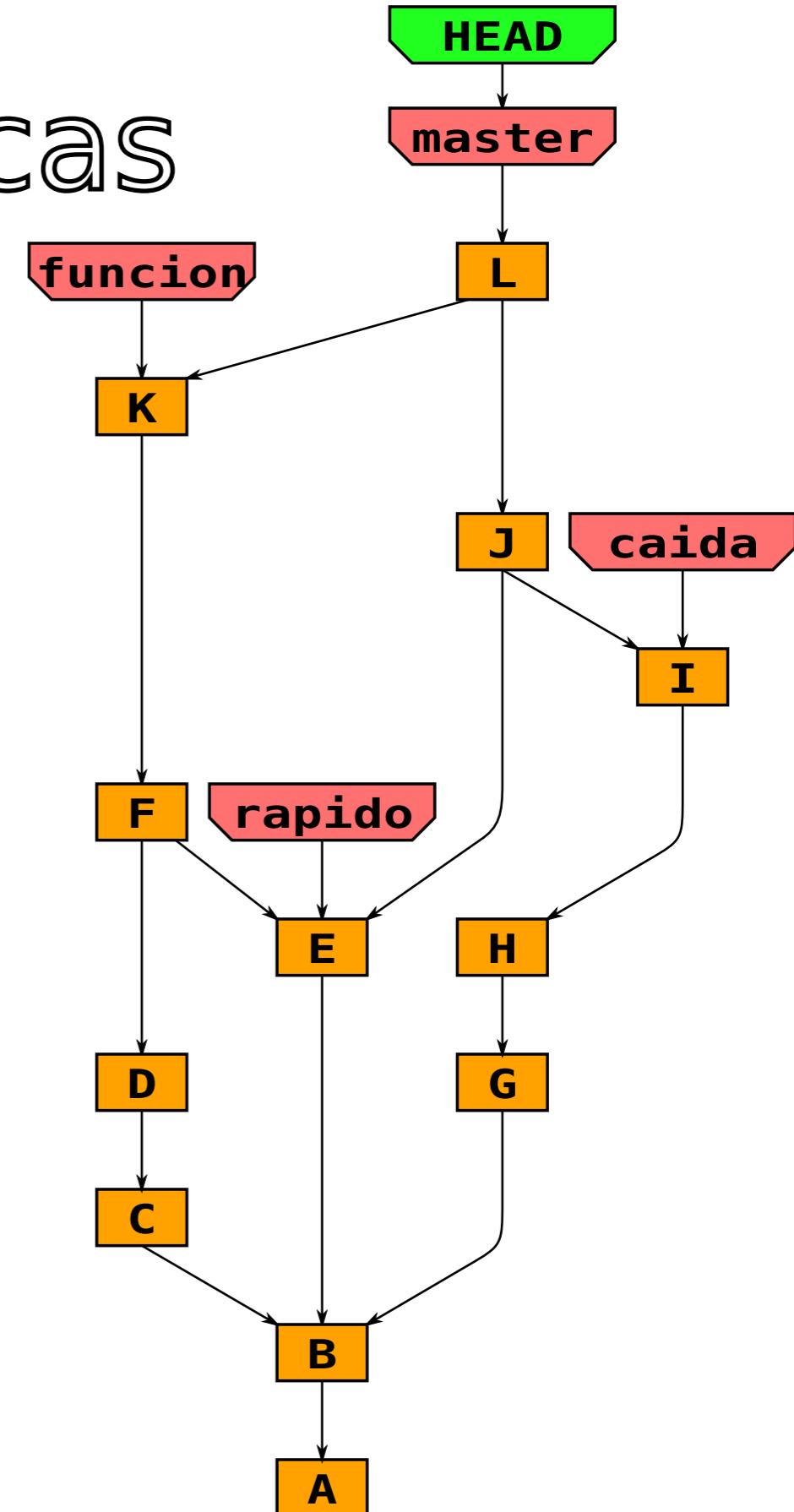
nos traemos la nueva función

# Ramas temáticas



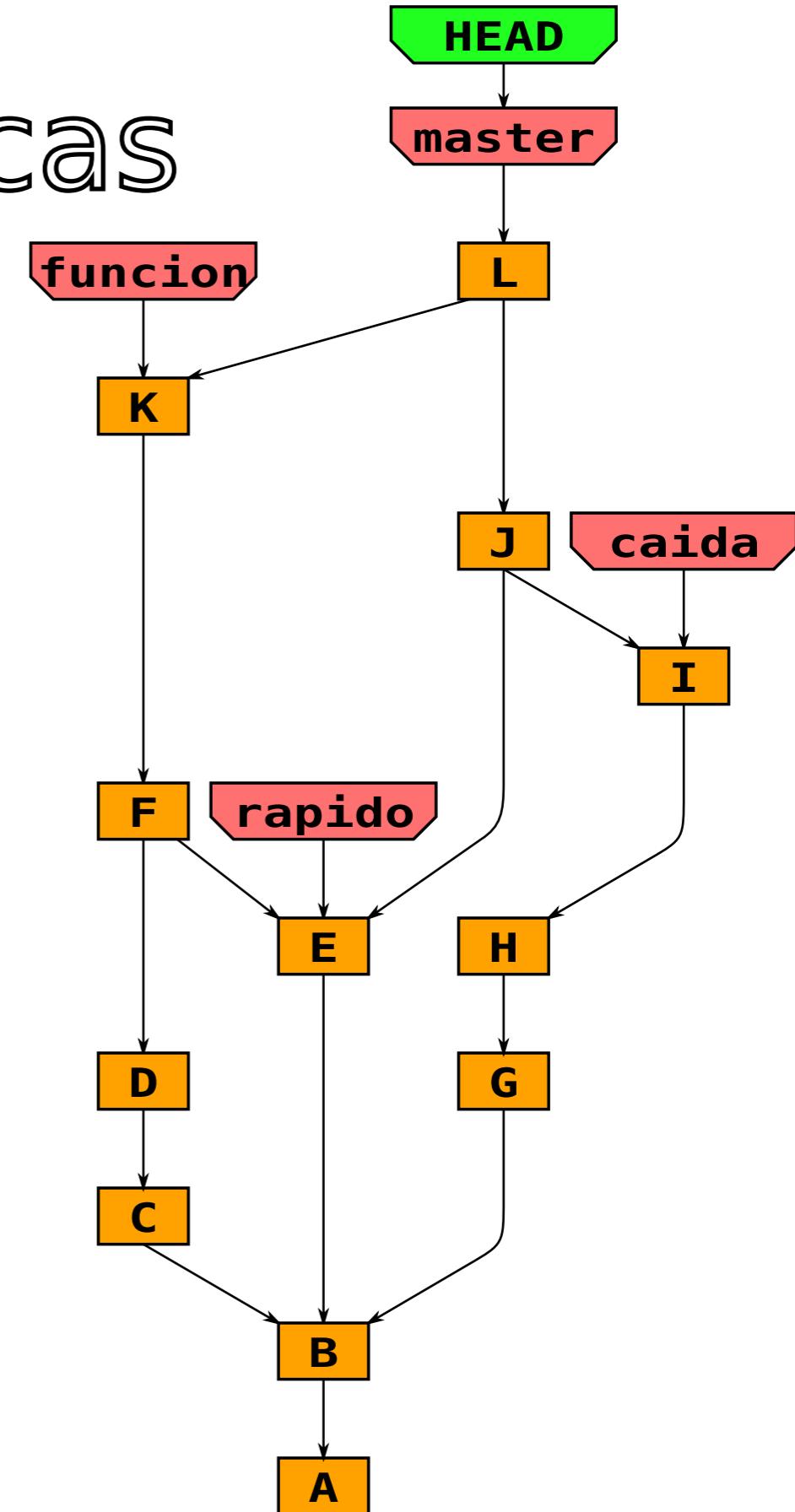
# Ramas temáticas

- mejor visibilidad



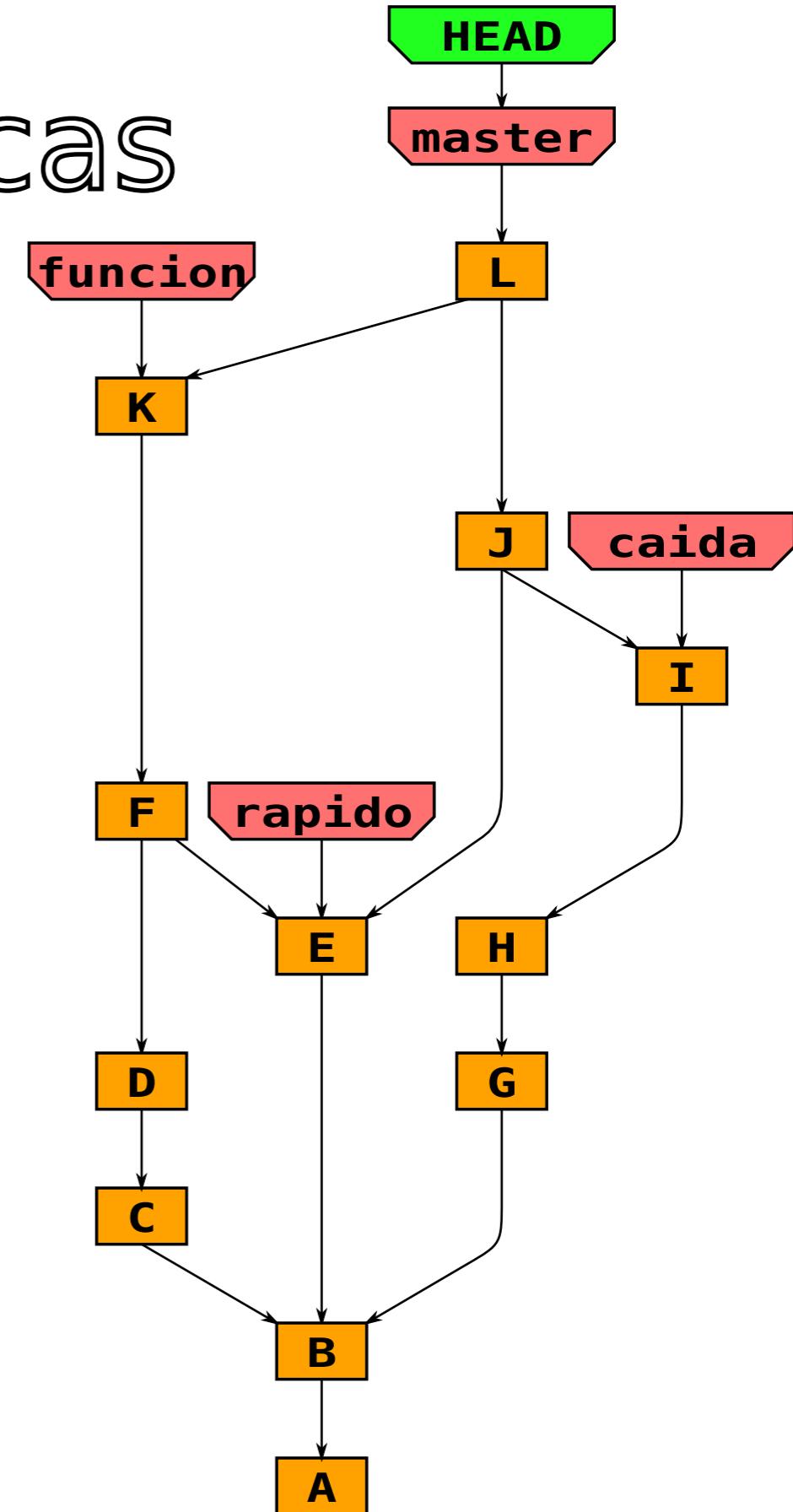
# Ramas temáticas

- mejor visibilidad
- ayuda al revisar el código



# Ramas temáticas

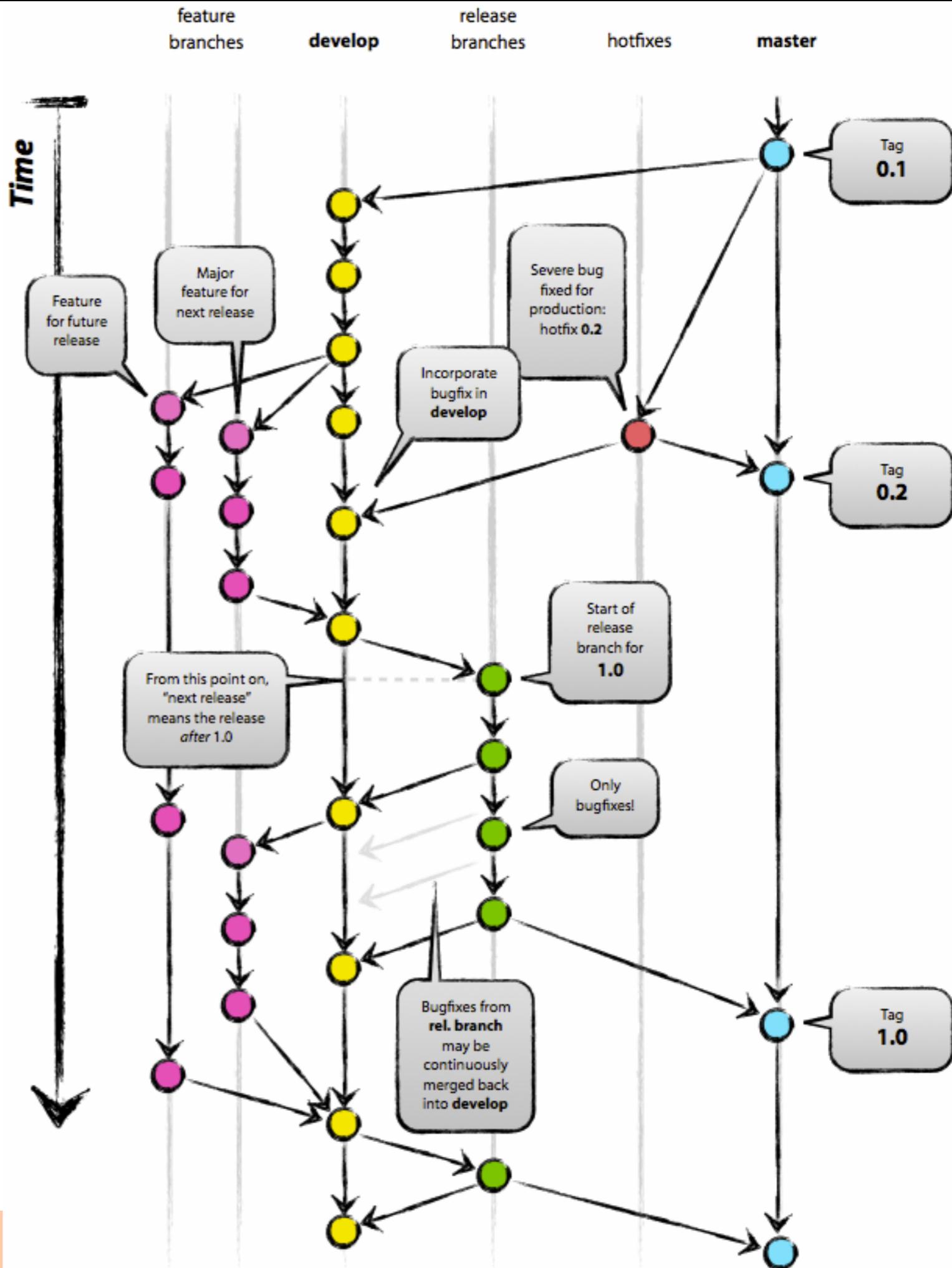
- mejor visibilidad
- ayuda al revisar el código
- se postponen las decisiones de reunión de ramas



# Híbrido

Visitad:

[http://nvie.com/  
git-model](http://nvie.com/git-model)



hasta ahora, todo ha sido local

hasta ahora, todo ha sido local

vamos al caso

Distribuido

# Alojamiento

# Alojamiento

repo.or.cz

# Alojamiento

repo.or.cz



# Alojamiento

repo.or.cz



GITORIOUS

# Alojamiento

repo.or.cz



GITORIOUS

<http://git.or.cz/gitwiki/GitHosting>

# Uso de Git Distribuido



## Create a New Repository

Create a new empty repository into which you can push your local git repo.

**NOTE:** If you intend to push a copy of a repository that is already hosted on GitHub, please [fork it instead](#).

Project Name

Description

Homepage URL

Who has access to this repository? (You can change this later)

- Anyone ([learn more about public repos](#))
- [Upgrade your plan to create more private repositories!](#)

## Global setup:

Download and install Git

```
git config --global user.name "Bart Trojanowski"
```

```
git config --global user.email bart@jukie.net
```

## Next steps:

```
mkdir test
```

```
cd test
```

```
git init
```

```
touch README
```

```
git add README
```

```
git commit -m 'first commit'
```

```
git remote add origin git@github.com:bartman/test.git
```

```
git push origin master
```

## Existing Git Repo?

```
cd existing_git_repo
```

```
git remote add origin git@github.com:bartman/test.git
```

```
git push origin master
```

## Importing a Subversion Repo?

[Click here](#)

## When you're done:

[Continue](#)

## Global setup:

Download and install Git

```
git config --global user.name "Bart Trojanowski"
```

```
git config --global user.email bart@jukie.net
```

## Next steps:

```
mkdir test
```

```
cd test
```

```
git init
```

```
touch README
```

```
git add README
```

```
git commit -m 'first commit'
```

```
git remote add origin git@github.com:bartman/test.git
```

```
git push origin master
```

## Existing Git Repo?

```
cd existing_git_repo
```

```
git remote add origin git@github.com:bartman/test.git
```

```
git push origin master
```

## Importing a Subversion Repo?

[Click here](#)

## When you're done:

[Continue](#)

## Global setup:

```
Download and install Git  
git config --global user.name "Bart Trojanowski"  
git config --global user.email bart@jukie.net
```

## Next steps:

```
mkdir test  
cd test  
git init  
touch README  
git add README  
git commit -m 'first commit'  
git remote add origin git@github.com:bartman/test.git  
git push origin master
```

## Existing Git Repo?

```
cd existing_git_repo  
git remote add origin git@github.com:bartman/test.git  
git push origin master
```

## Importing a Subversion Repo?

[Click here](#)

## When you're done:

[Continue](#)

## Global setup:

Download and install Git

```
git config --global user.name "Bart Trojanowski"
```

```
git config --global user.email bart@jukie.net
```

## Next steps:

```
mkdir test
```

```
cd test
```

```
git init
```

```
touch README
```

```
git add README
```

```
git commit -m 'first commit'
```

```
git remote add origin git@github.com:bartman/test.git
```

```
git push origin master
```

## Existing Git Repo?

```
cd existing git repo
```

```
git remote add origin git@github.com:bartman/test.git
```

```
git push origin master
```

## Importing a Subversion Repo?

[Click here](#)

## When you're done:

[Continue](#)

# Remotes

repositorios externos que vigilamos

# Remotes

git remote

# Remotes

```
git remote  
origin
```

# Remotes

git remote

github  
gitorious  
origin  
publish  
repo

# Remotes

`git remote -v`

github	<code>git@github.com:bartman/test.git</code>
gitorious	<code>git://gitorious.org/bartman/test.git</code>
origin	<code>http://www.kernel.org/pub/scm/bartman/test.git</code>
publish	<code>/var/www/bartman/test.git</code>
repo	<code>ssh://repo.or.cz/srv/git/bartman-test.git</code>

# Remotes

git remote -v

github	git@github.com:bartman/test.git
gitorious	git://gitorious.org/bartman/test.git
origin	http://www.kernel.org/pub/scm/bartman/test.git
publish	/var/www/bartman/test.git
repo	ssh://repo.or.cz/srv/git/bartman-test.git

ssh

# Remotes

`git remote -v`

github	<code>git@github.com:bartman/test.git</code>
gitorious	<code>git://gitorious.org/bartman/test.git</code>
origin	<code>http://www.kernel.org/pub/scm/bartman/test.git</code>
publish	<code>/var/www/bartman/test.git</code>
repo	<code>ssh://repo.or.cz/srv/git/bartman-test.git</code>

`git nativo`

# Remotes

git remote -v

github	git@github.com:bartman/test.git
gitorious	git://gitorious.org/bartman/test.git
origin	http://www.kernel.org/pub/scm/bartman/test.git
publish	/var/www/bartman/test.git
repo	ssh://repo.or.cz/srv/git/bartman-test.git

http / https

# Remotes

`git remote -v`

github	<code>git@github.com:bartman/test.git</code>
gitorious	<code>git://gitorious.org/bartman/test.git</code>
origin	<code>http://www.kernel.org/pub/scm/bartman/test.git</code>
publish	<code>/var/www/bartman/test.git</code>
repo	<code>ssh://repo.or.cz/srv/git/bartman-test.git</code>

sistema de ficheros local

# Remotes

`git remote -v`

github	<code>git@github.com:bartman/test.git</code>
gitorious	<code>git://gitorious.org/bartman/test.git</code>
origin	<code>http://www.kernel.org/pub/scm/bartman/test.git</code>
publish	<code>/var/www/bartman/test.git</code>
repo	<code>ssh://repo.or.cz/srv/git/bartman-test.git</code>

... también rsync

# Remotes

```
git remote add <nombre> <url>
```

## Global setup:

```
Download and install Git  
git config --global user.name "Bart Trojanowski"  
git config --global user.email bart@jukie.net
```

## Next steps:

```
mkdir test  
cd test  
git init  
touch README  
git add README  
git commit -m 'first commit'  
git remote add origin git@github.com:bartman/test.git  
git push origin master
```

## Existing Git Repo?

```
cd existing git repo  
git remote add origin git@github.com:bartman/test.git  
git push origin master
```

## Importing a Subversion Repo?

[Click here](#)

## When you're done:

[Continue](#)

```
git remote add origin \
git@github.com:bartman/test.git
```

```
git push git@github.com:bartman/test.git master
```

```
git remote add origin \
git@github.com:bartman/test.git
```

```
git push origin master
```

# Envío de objetos

`git push`

# Envío de objetos

`git push <remote>`

- envía nuevos objetos a `<remote>`

# Envío de objetos

`git push <remote> <rama>`

- envía nuevos objetos a `<remote>`
- limitado a una `<rama>`

# Envío de objetos

`git push origin master`

# Envío de objetos

`git push`

# Recepción de objetos

`git fetch`

# Recepción de objetos

`git fetch <remote>`

- actualiza refs locales de `<remote>`

# Recepción de objetos

`git fetch <remote>`

- actualiza refs locales de `<remote>`
- no toca **HEAD**  
(no cambia el directorio de trabajo)

# Recepción de objetos

`git fetch <remote>`

- actualiza refs locales de `<remote>`
- no toca **HEAD**  
(no cambia el directorio de trabajo)

Para actualizar el árbol de trabajo...

# Recepción de objetos

`git fetch <remote>`

- actualiza refs locales de `<remote>`
- no toca **HEAD**  
(no cambia el directorio de trabajo)

Para actualizar el árbol de trabajo...

`git fetch origin`

# Recepción de objetos

`git fetch <remote>`

- actualiza refs locales de `<remote>`
- no toca **HEAD**  
(no cambia el directorio de trabajo)

Para actualizar el árbol de trabajo...

`git fetch origin`

`git merge origin/master`

# Recepción de objetos

`git fetch <remote>`

- actualiza refs locales de `<remote>`
- no toca **HEAD**  
(no cambia el directorio de trabajo)

Para actualizar el árbol de trabajo...

```
git fetch origin  
git merge origin/master
```

# Recepción y actualización

`git pull`

# Recepción y actualización

`git pull <remote> <rama>`

# Recepción y actualización

`git pull <remote> <rama>`

- `fetch` de `<remote>`

# Recepción y actualización

`git pull <remote> <rama>`

- `fetch` de `<remote>`
- `merge` `<rama>` remota con `HEAD`

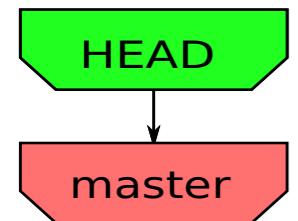
# Recepción y actualización

`git pull`

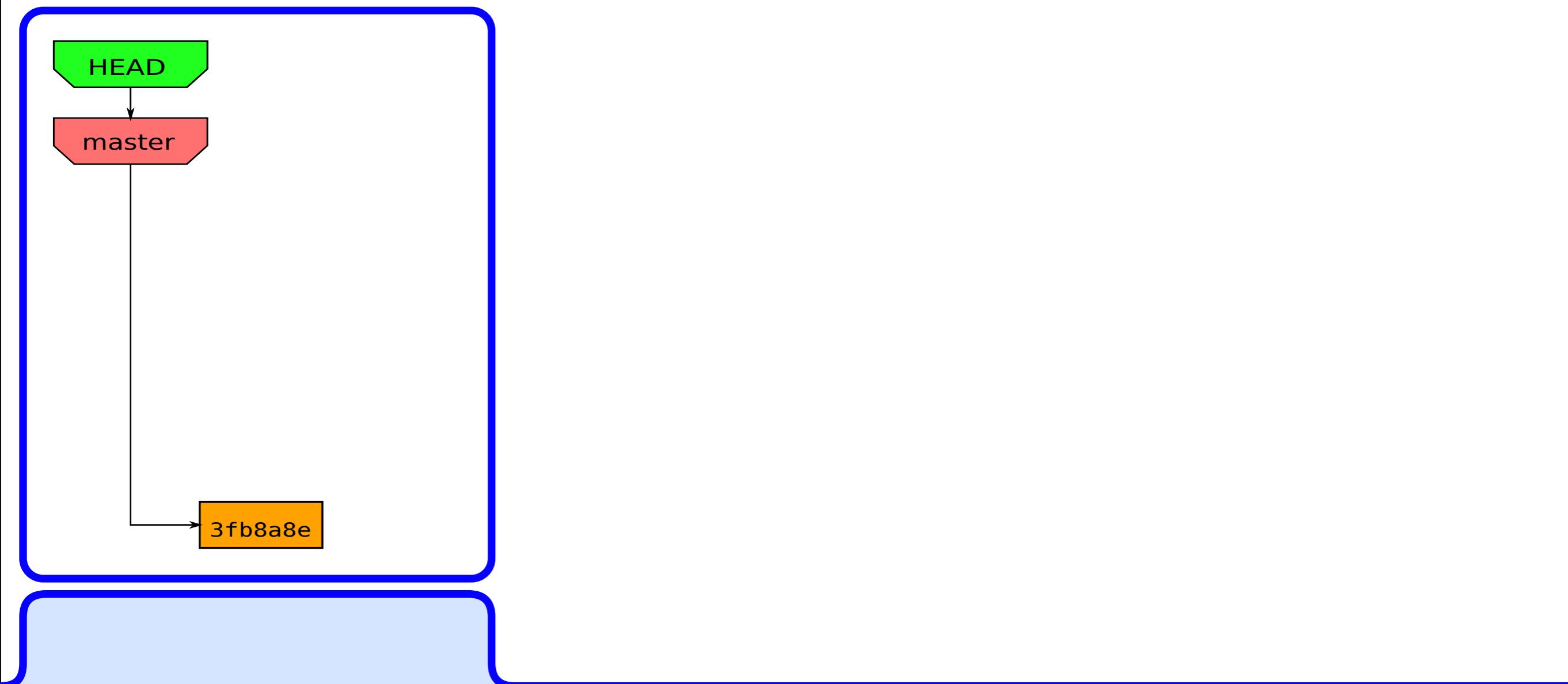
- `fetch` de **origin**
- `merge` una rama remota *coincidente*

# Ejemplo distribuido

# local

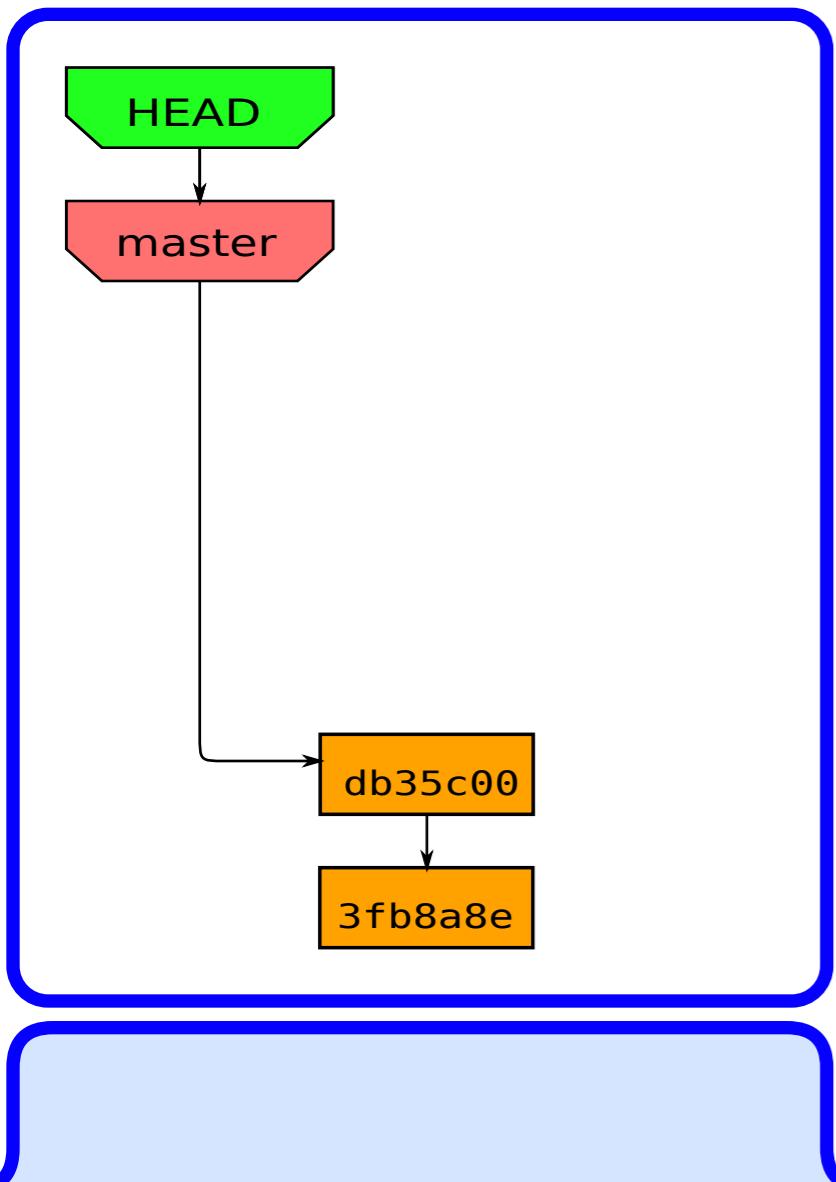


# local



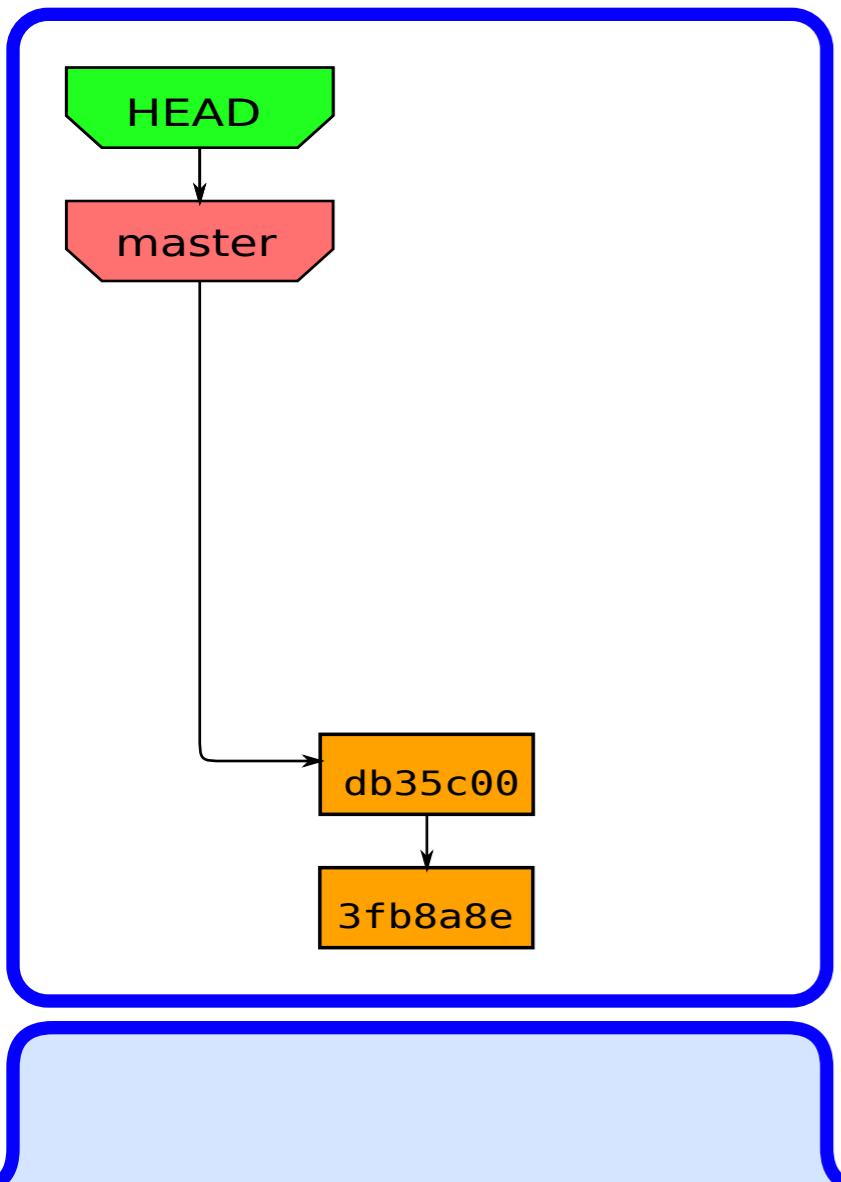
git commit

# local



git commit

# local



```
git remote add origin git@github:yo/proy
```

# local

# “origin” github

HEAD

master

HEAD

master

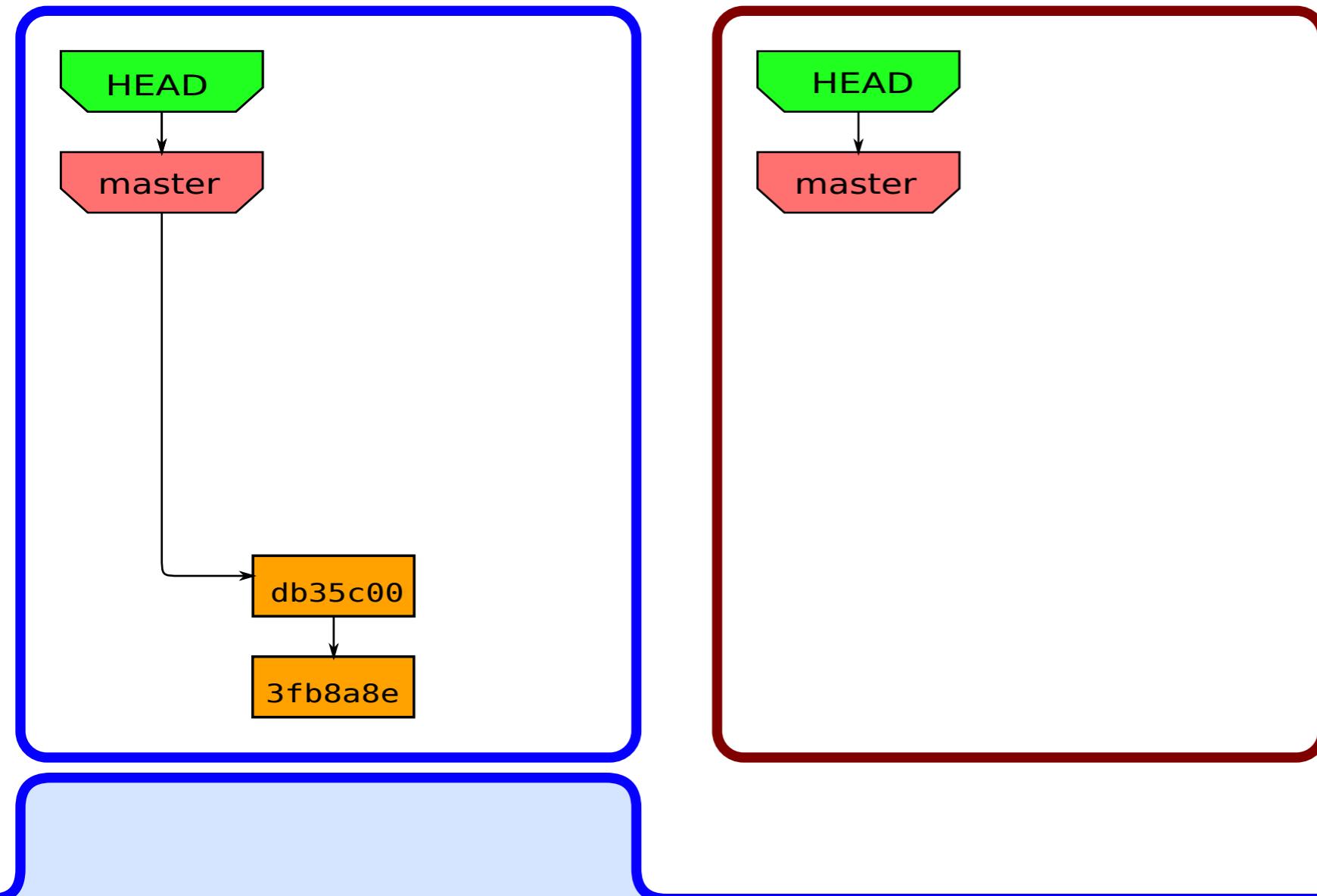
db35c00

3fb8a8e

```
git remote add origin git@github:yo/proy
```

# local

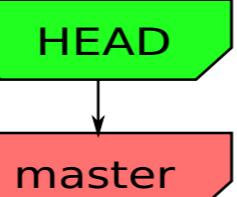
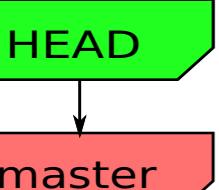
# “origin” github



git push origin master

# local

# “origin” github



PUSH

git push origin master

# local

# “origin” github

HEAD

master

db35c00

3fb8a8e

HEAD

master

db35c00

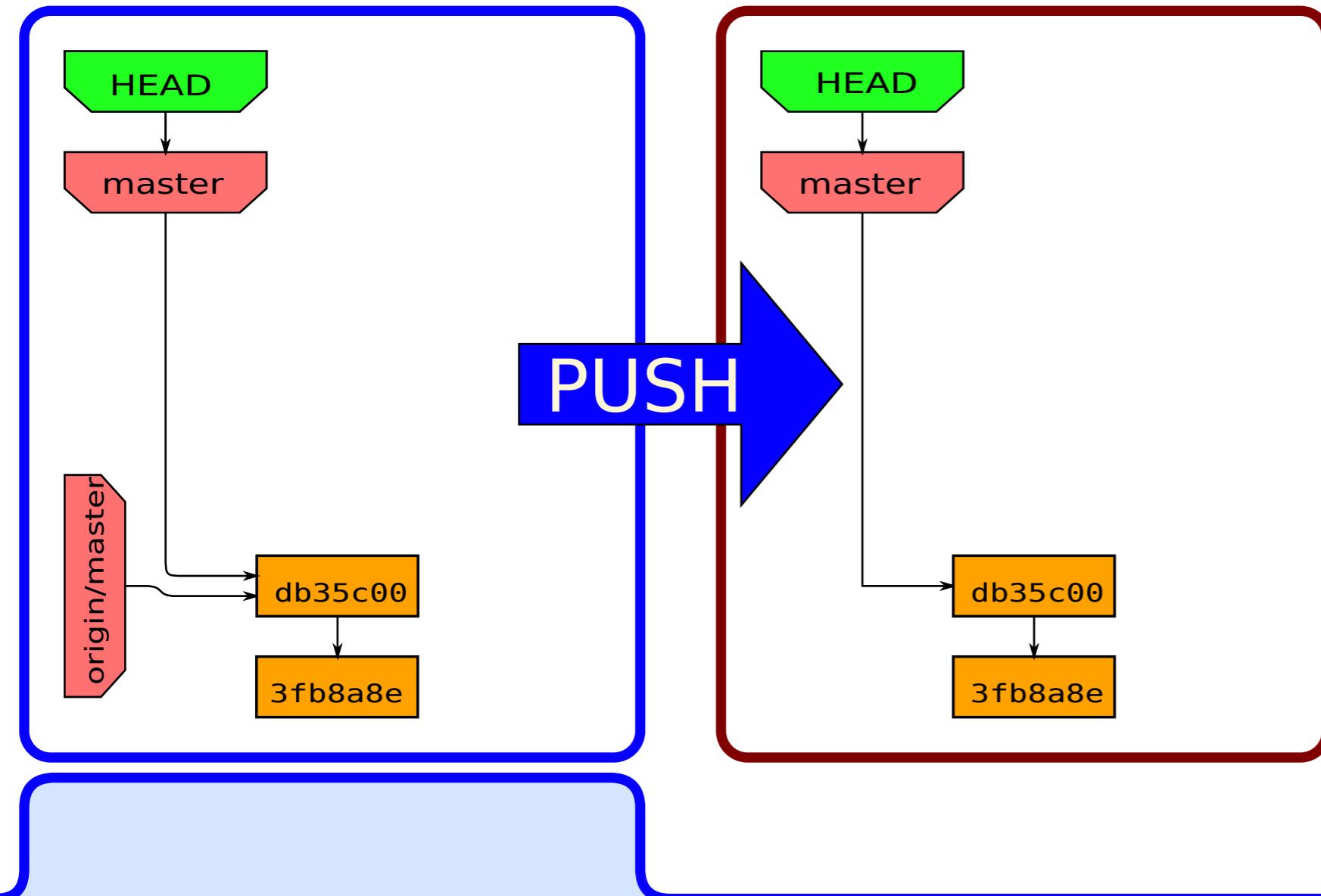
3fb8a8e

PUSH

git push origin master

# local

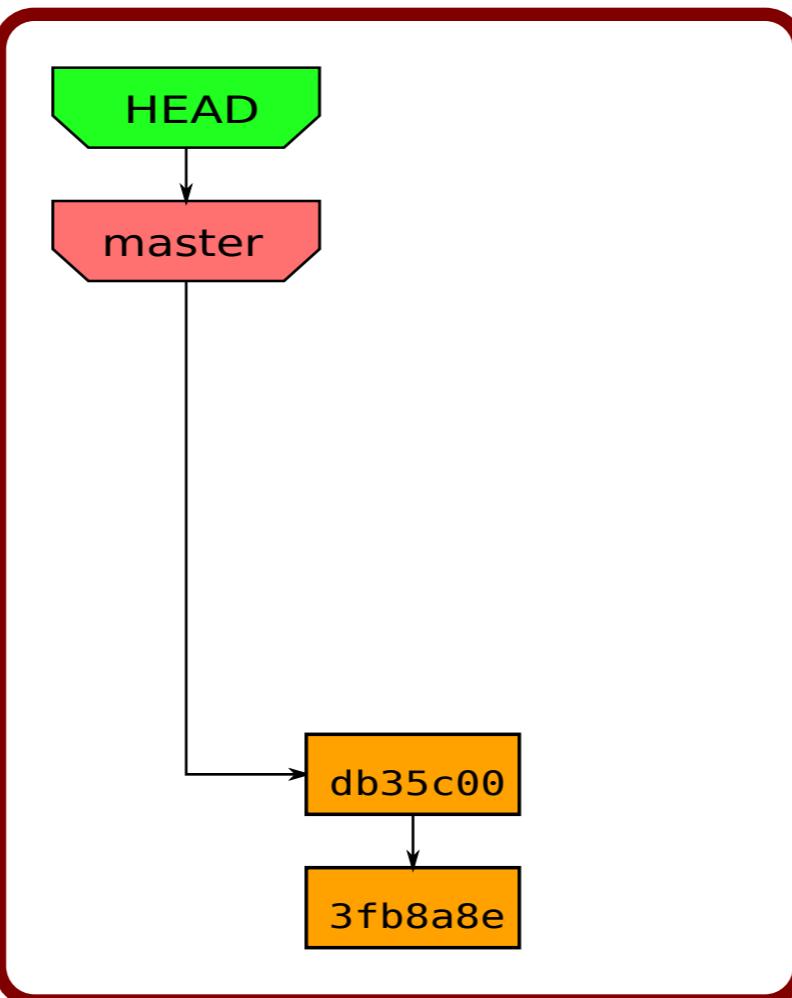
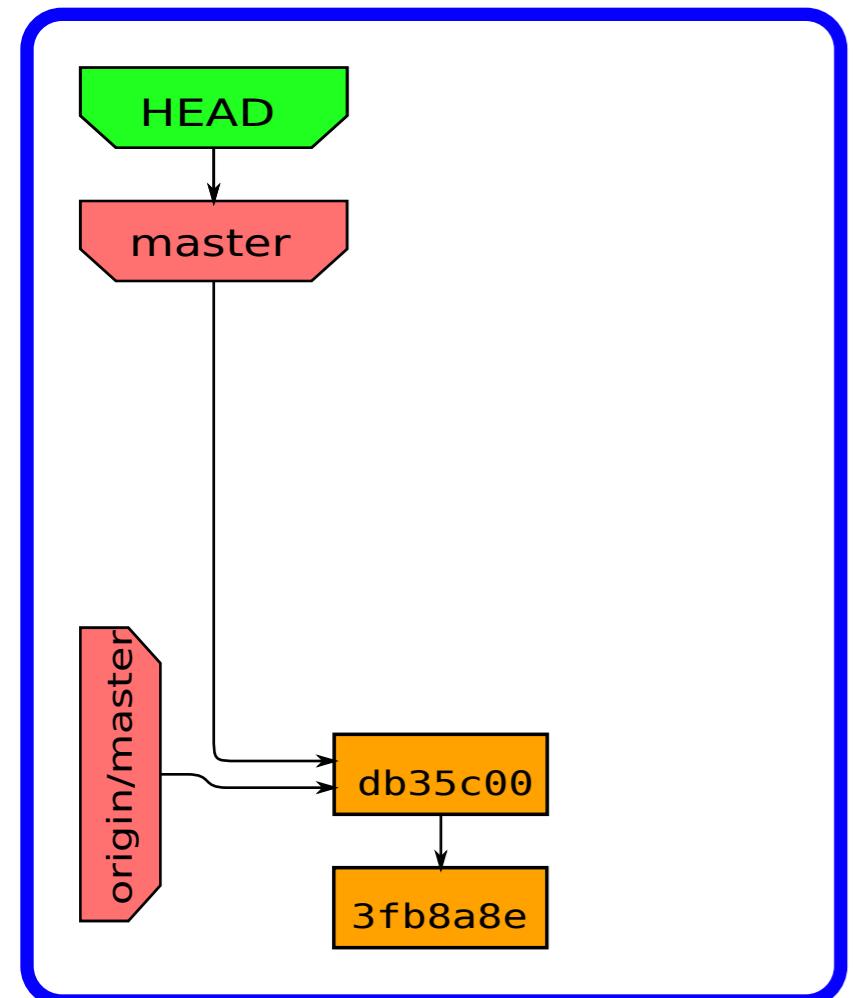
# “origin” github



git push origin master

# local

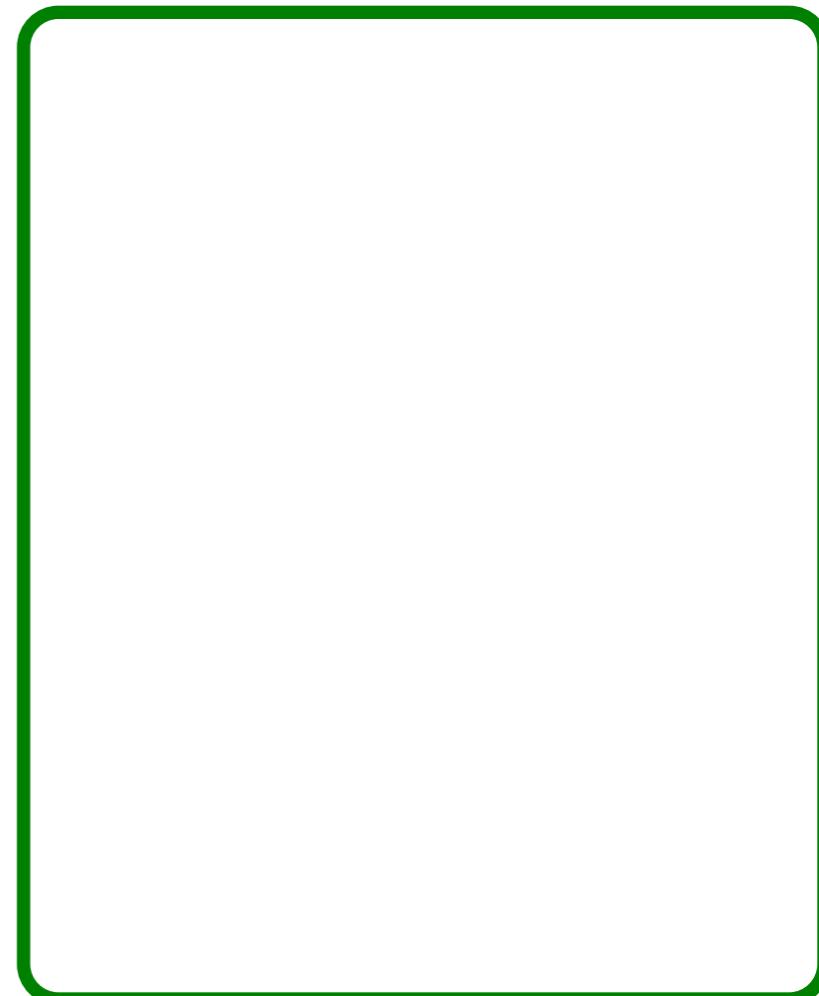
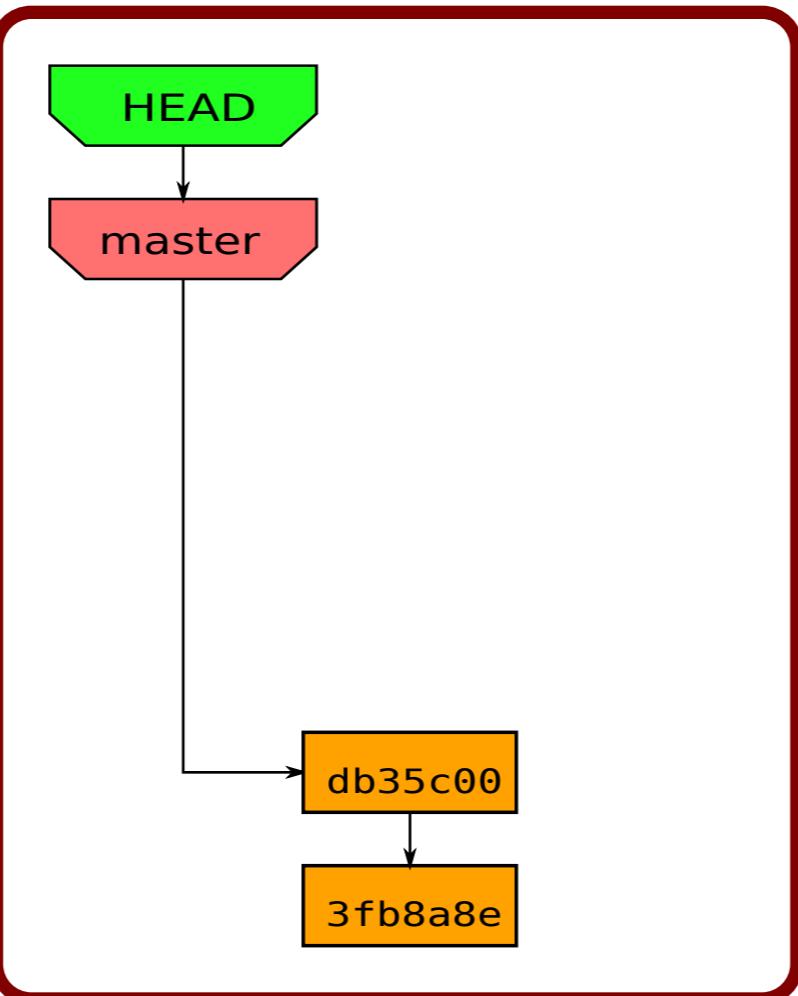
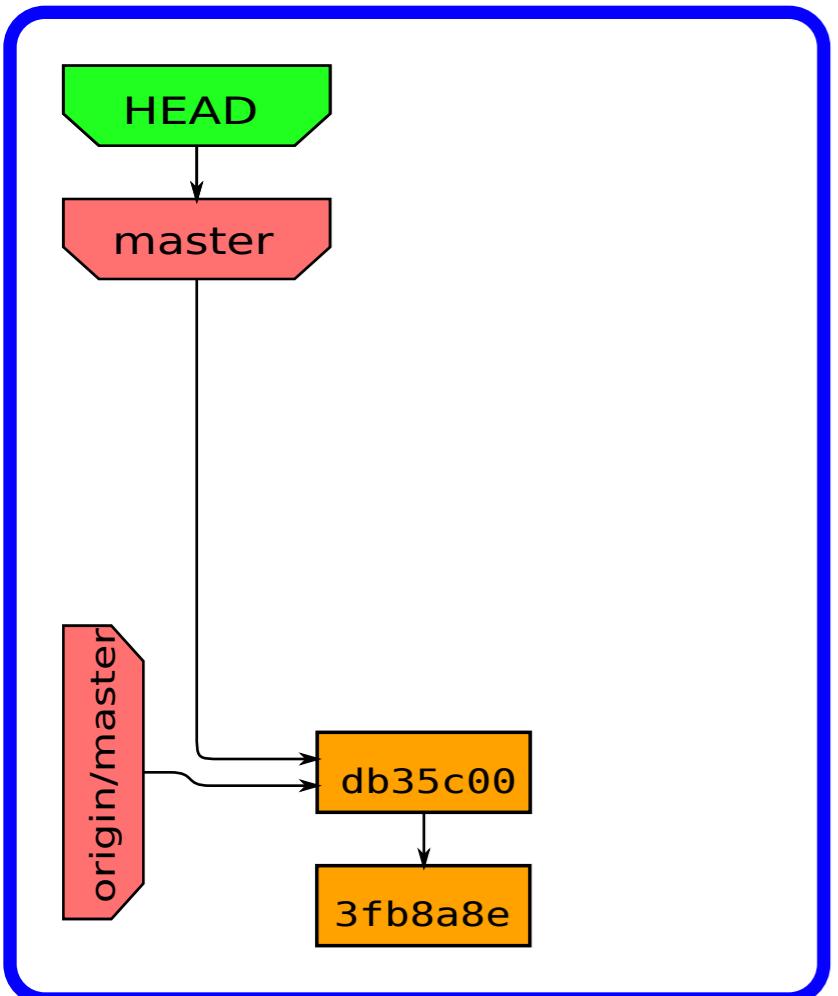
# “origin” github



# local

# “origin” github

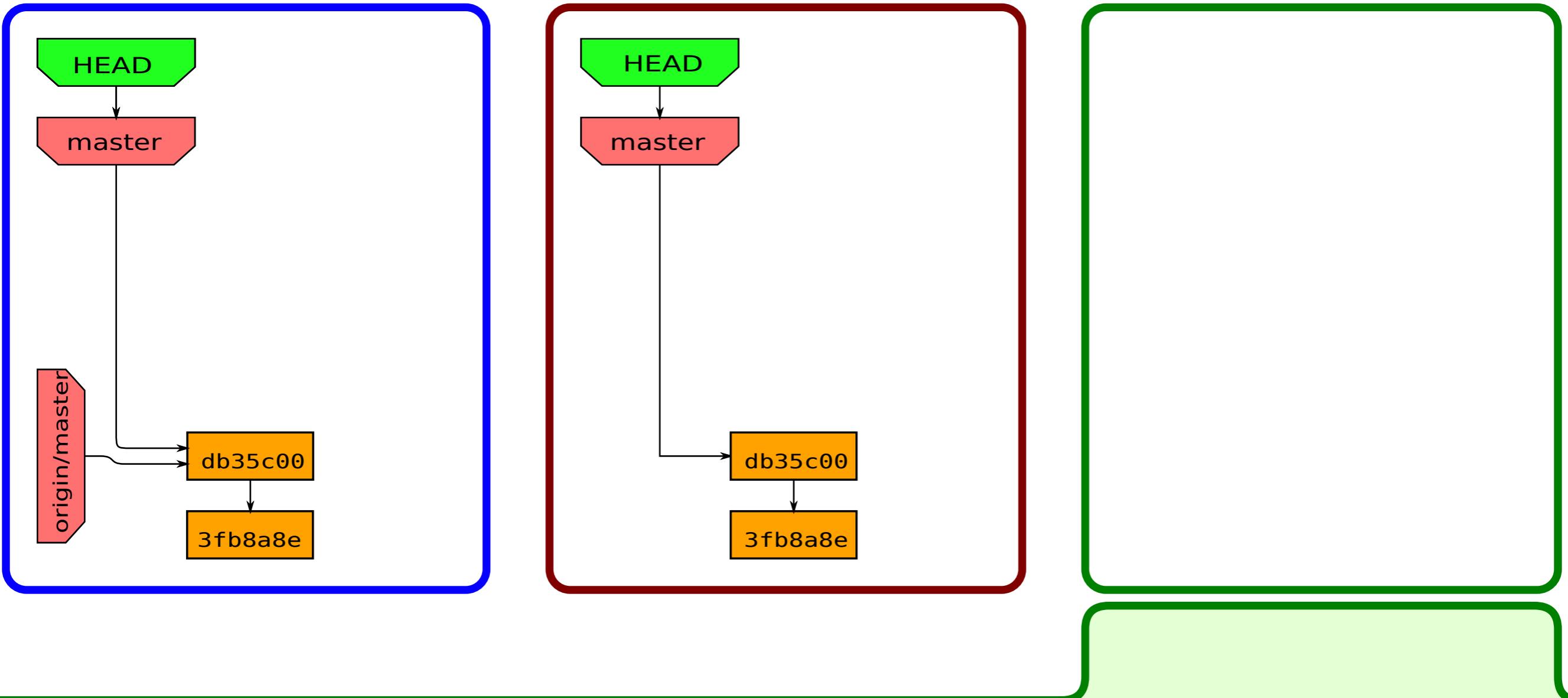
# Fred



# local

# “origin” github

# Fred

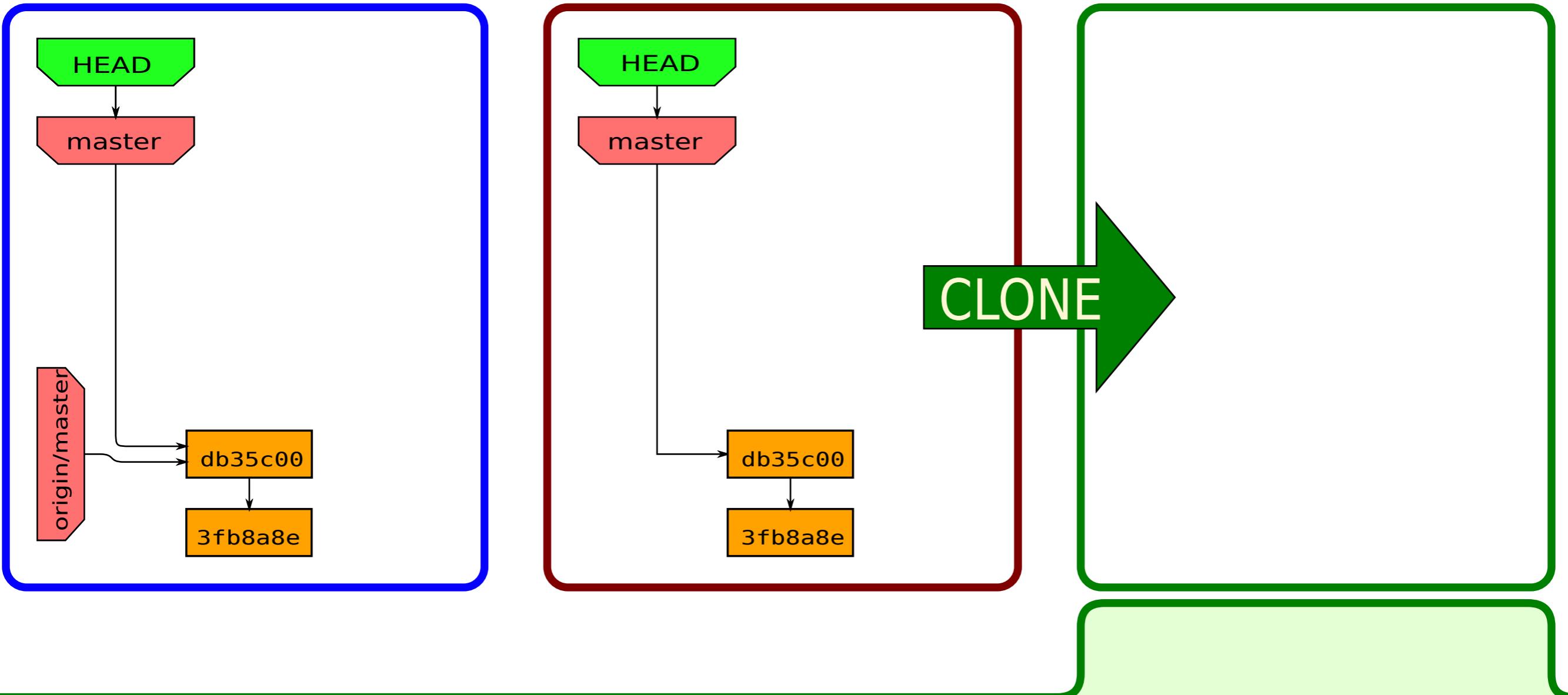


```
git clone git://github.com/bartman/proyecto.git
```

# local

# “origin” github

# Fred

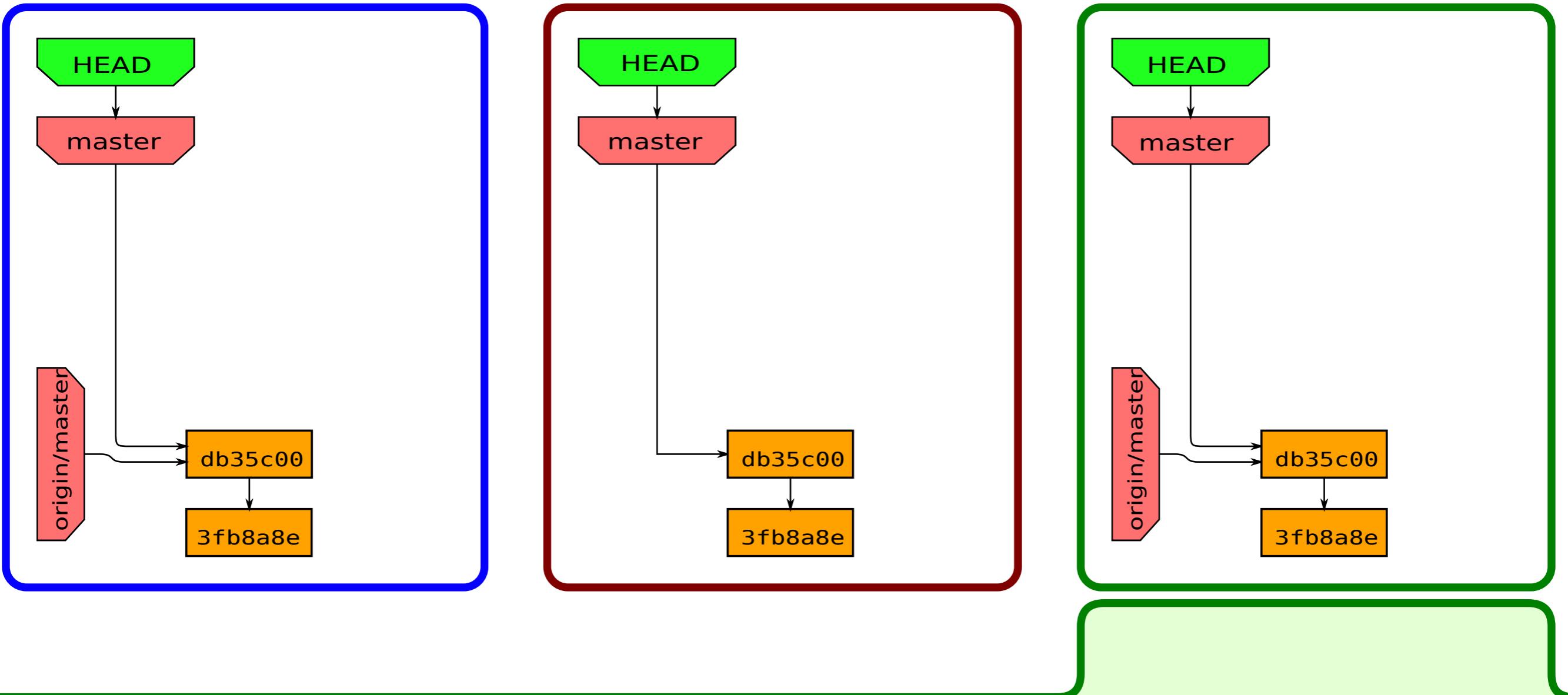


```
git clone git://github.com/bartman/proyecto.git
```

# local

# “origin” github

# Fred

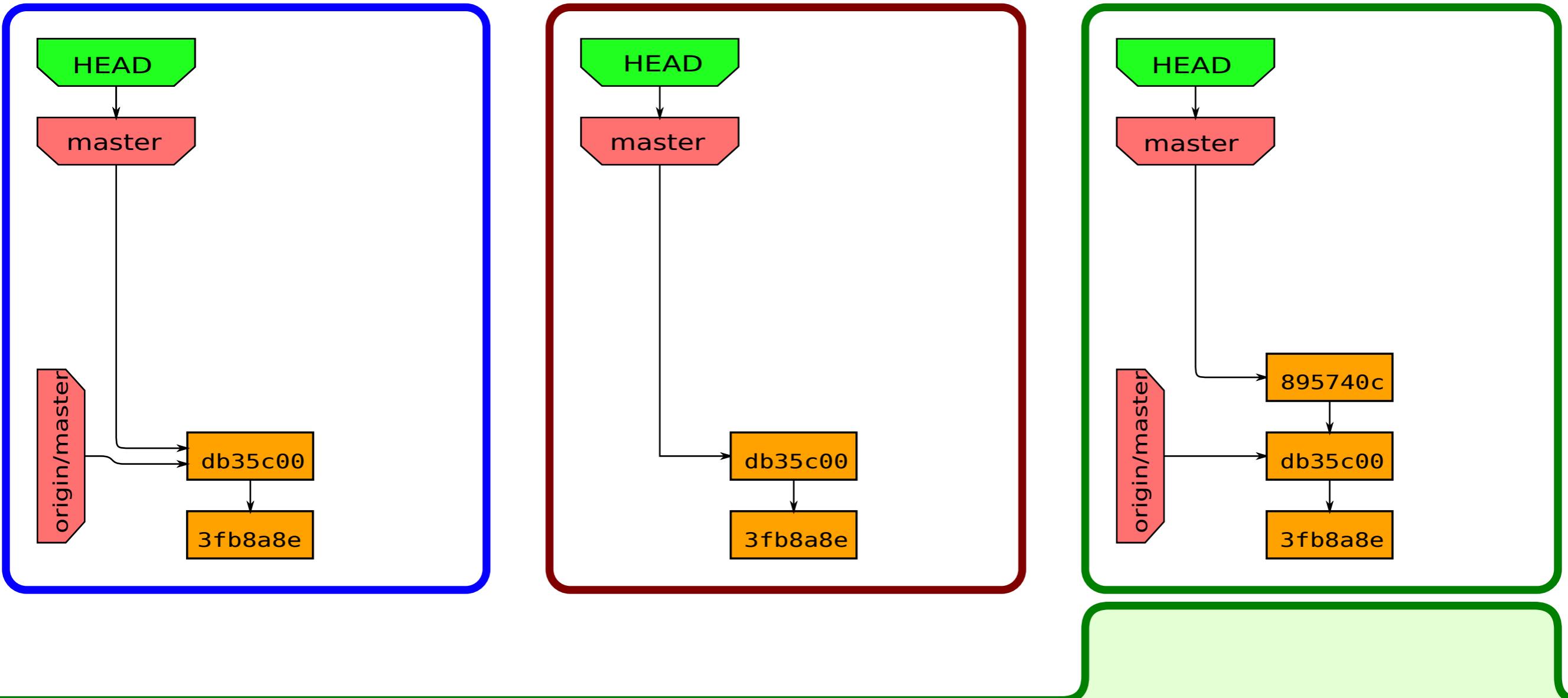


```
git clone git://github.com/bartman/proyecto.git
```

# local

# “origin” github

# Fred

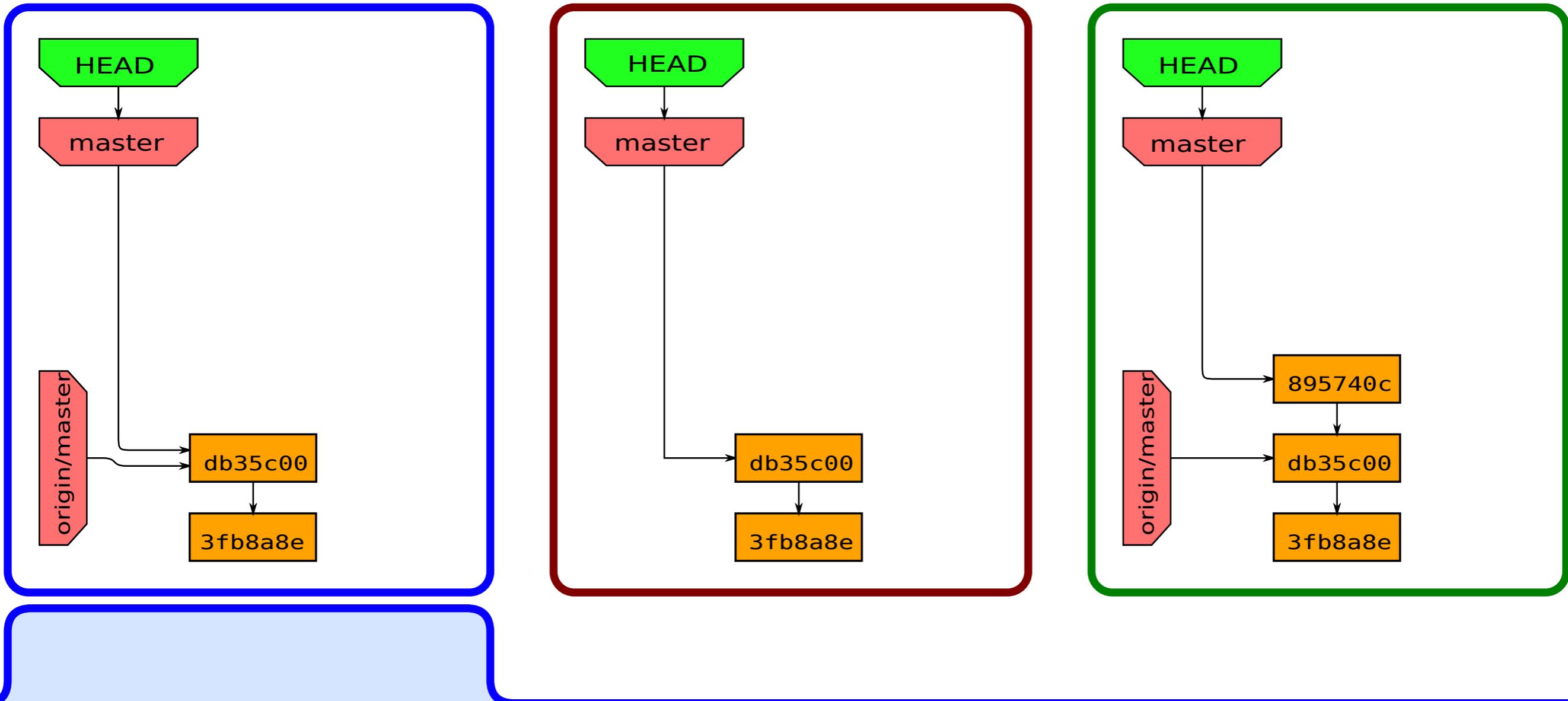


```
git commit -m "arreglado un defecto"
```

# local

# “origin” github

# Fred

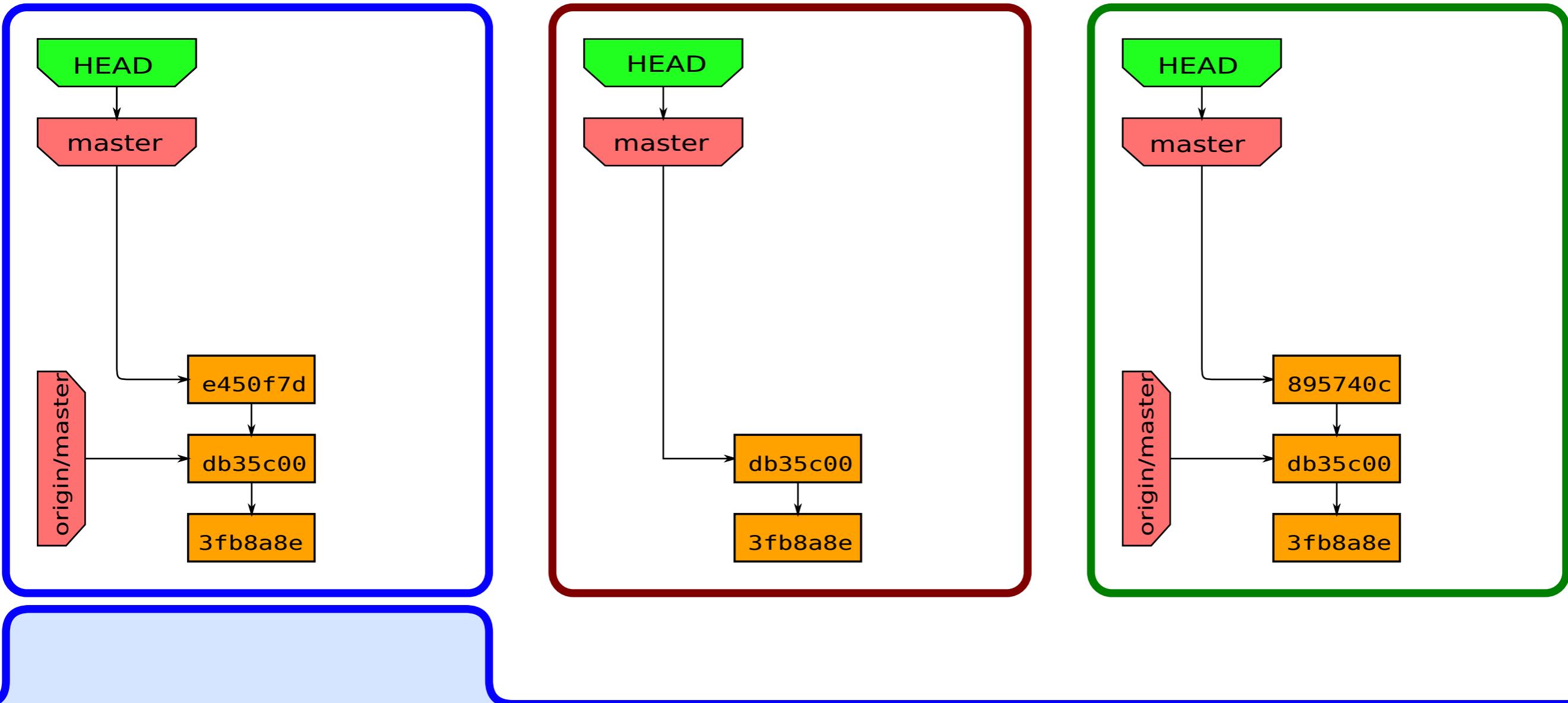


```
git commit -m "añadido funcionalidad"
```

# local

# “origin” github

# Fred

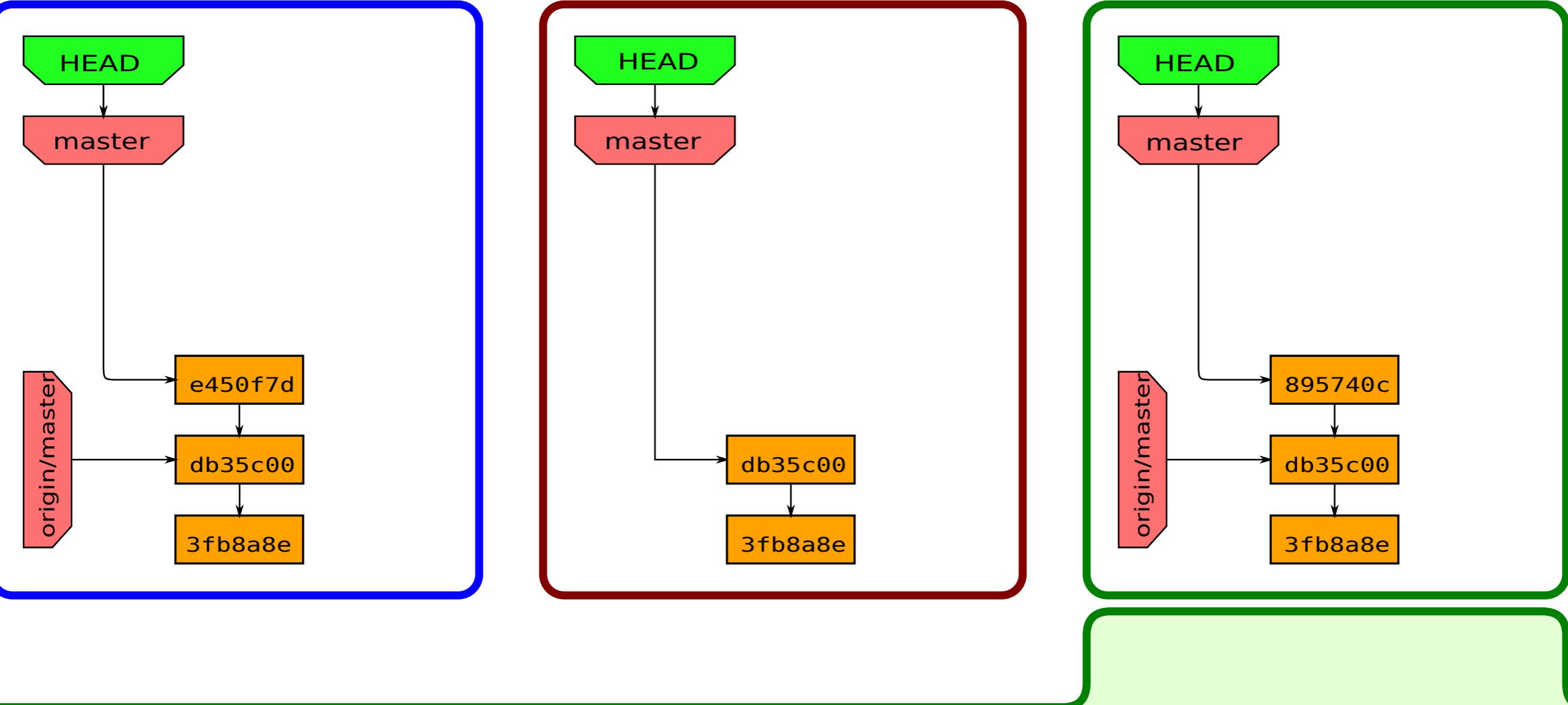


```
git commit -m "añadido funcionalidad"
```

# local

# “origin” github

# Fred



De: fred@foo.com  
Asunto: He arreglado un defecto

Por favor, tira de

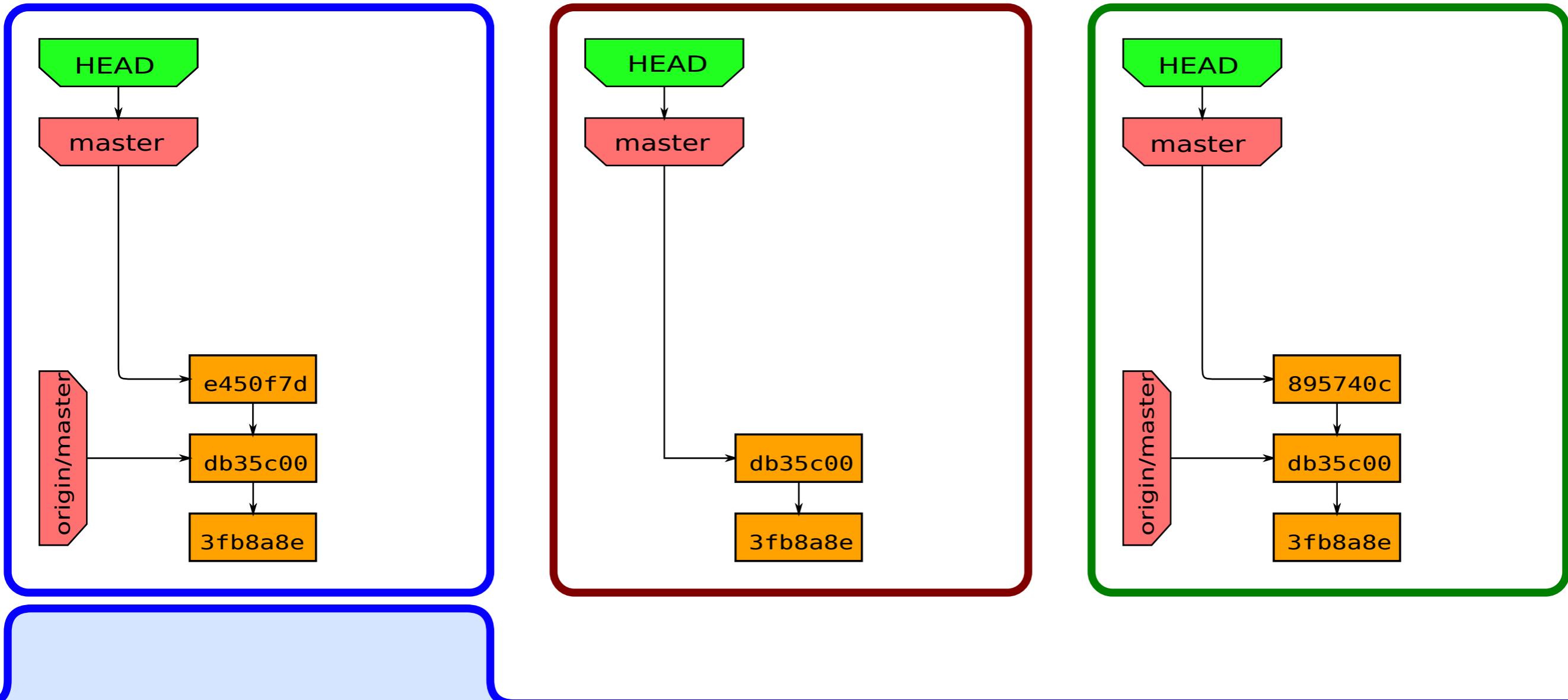
**<http://foo.com/proyecto.git>**

-Fred

# local

# “origin” github

# Fred

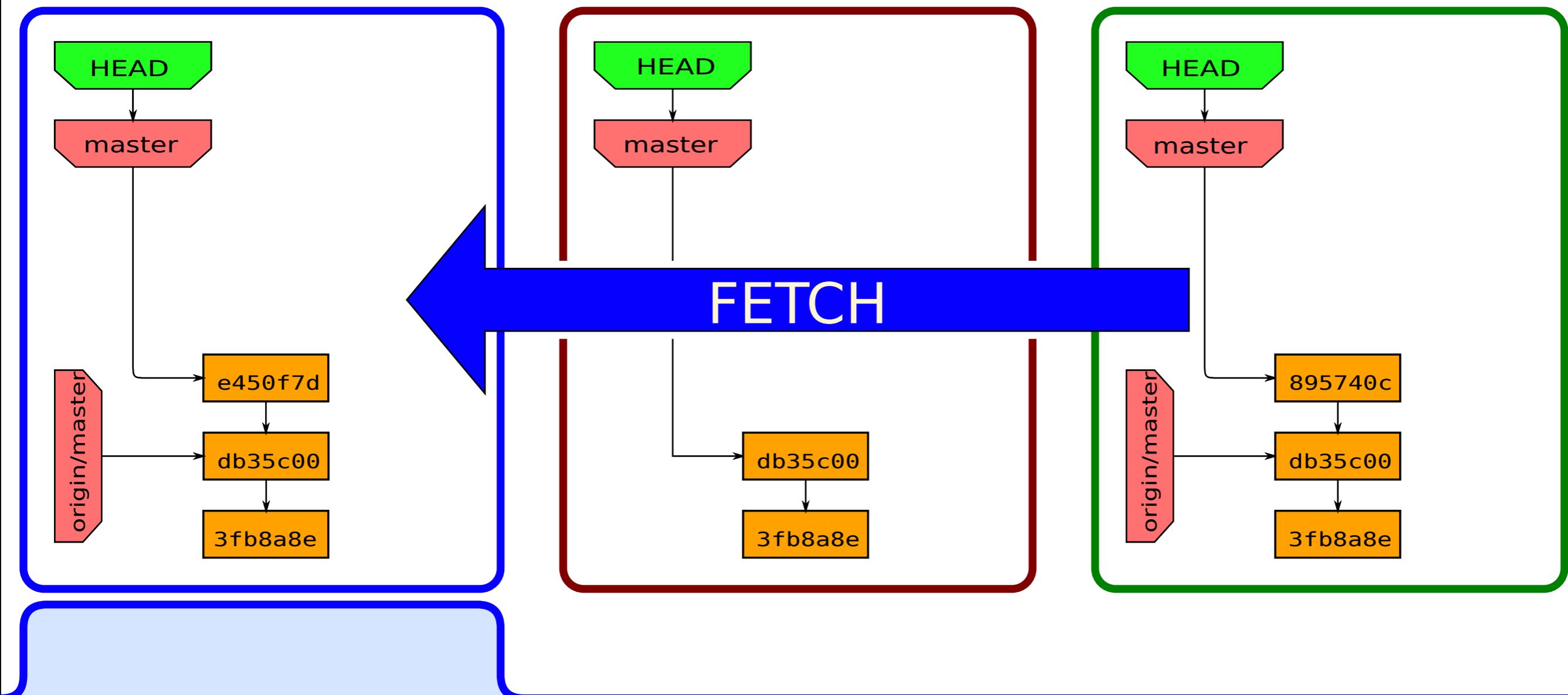


```
git remote add fred \
http://foo.com/proyecto.git
```

# local

# “origin” github

# Fred

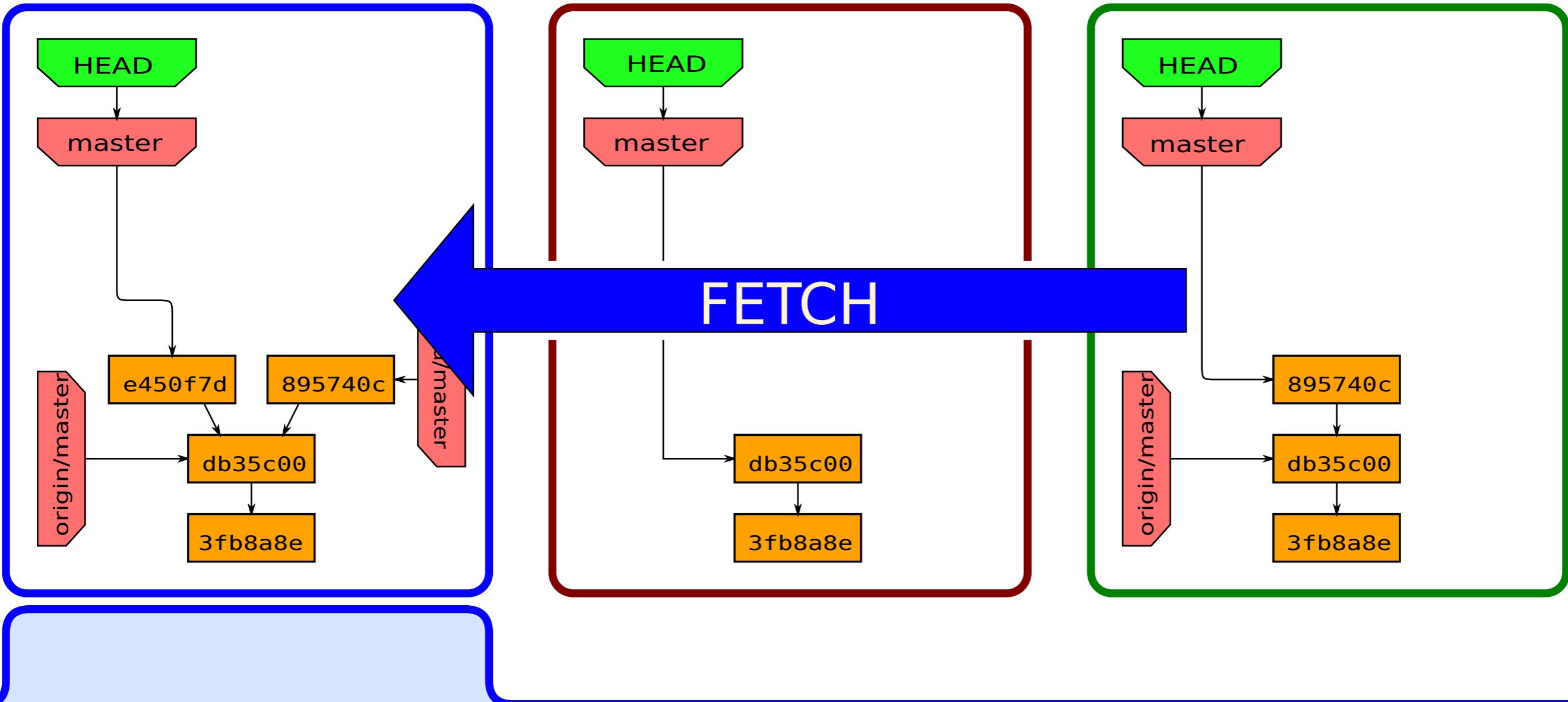


git fetch fred

# local

# “origin” github

# Fred

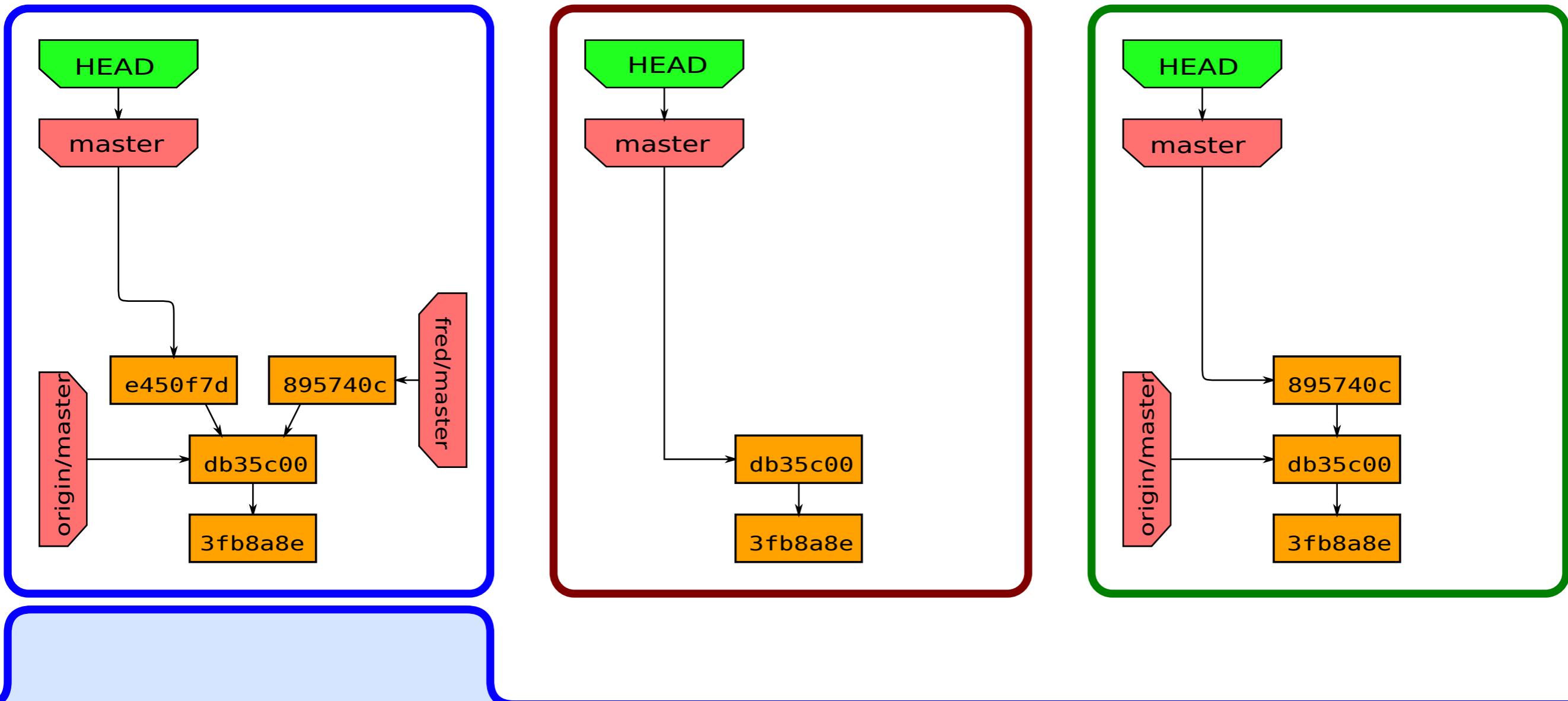


git fetch fred

# local

# “origin” github

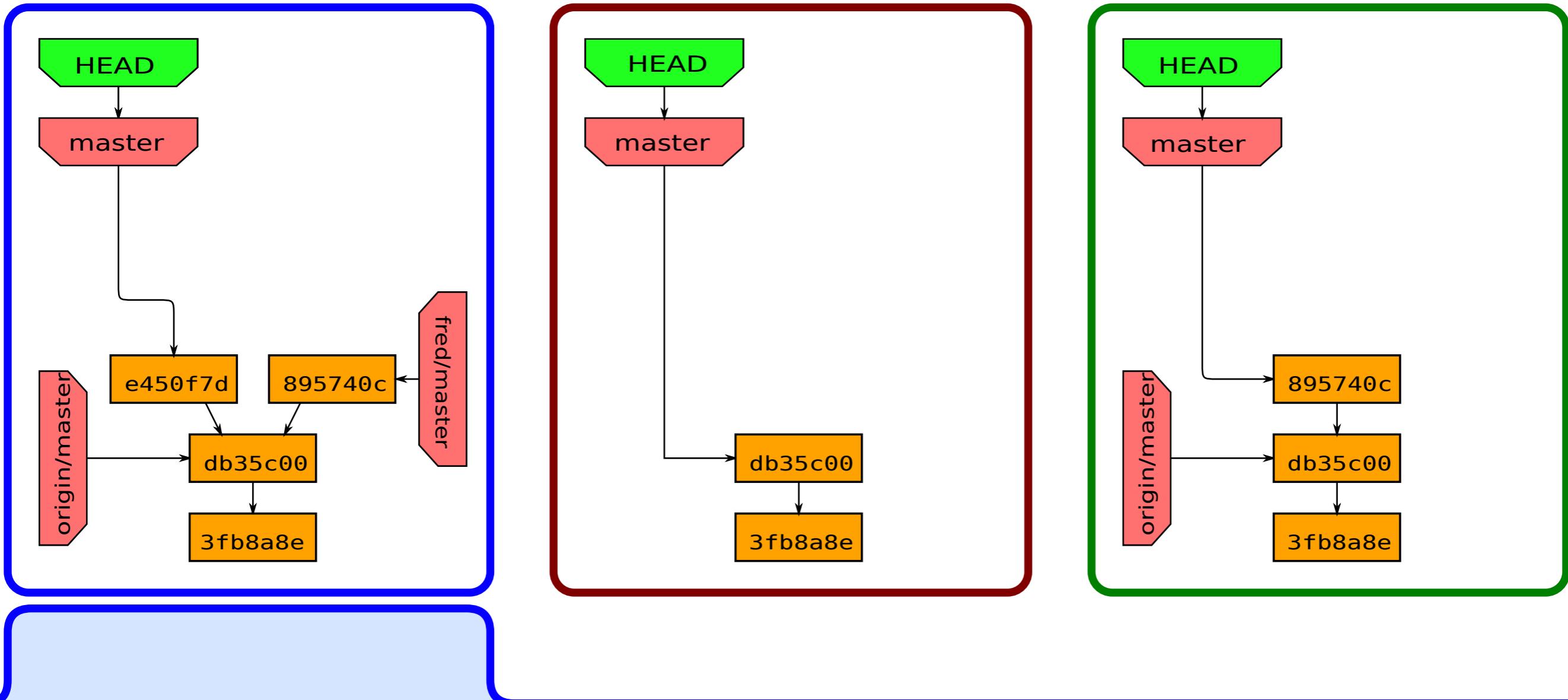
# Fred



# local

# “origin” github

# Fred

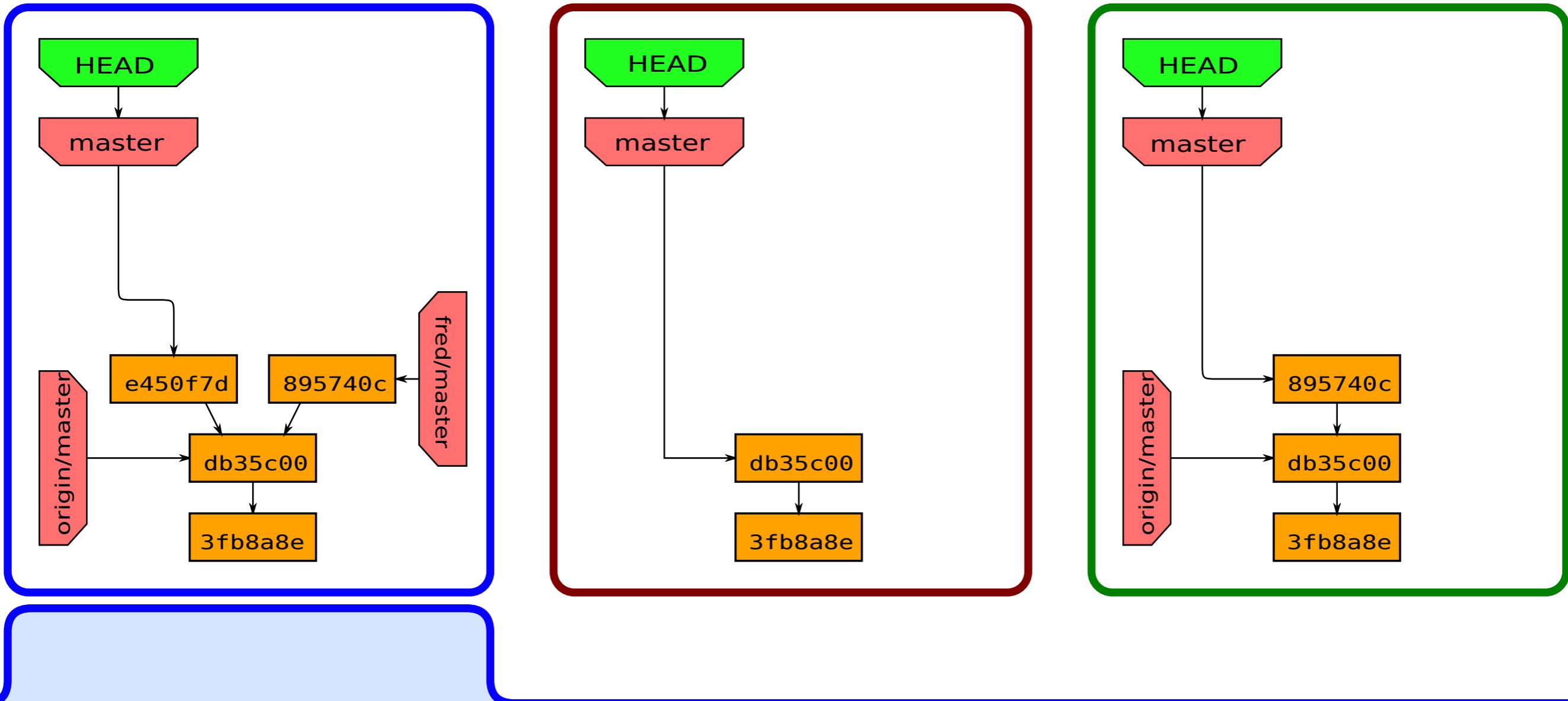


```
git log -p fred/master
```

# local

# “origin” github

# Fred



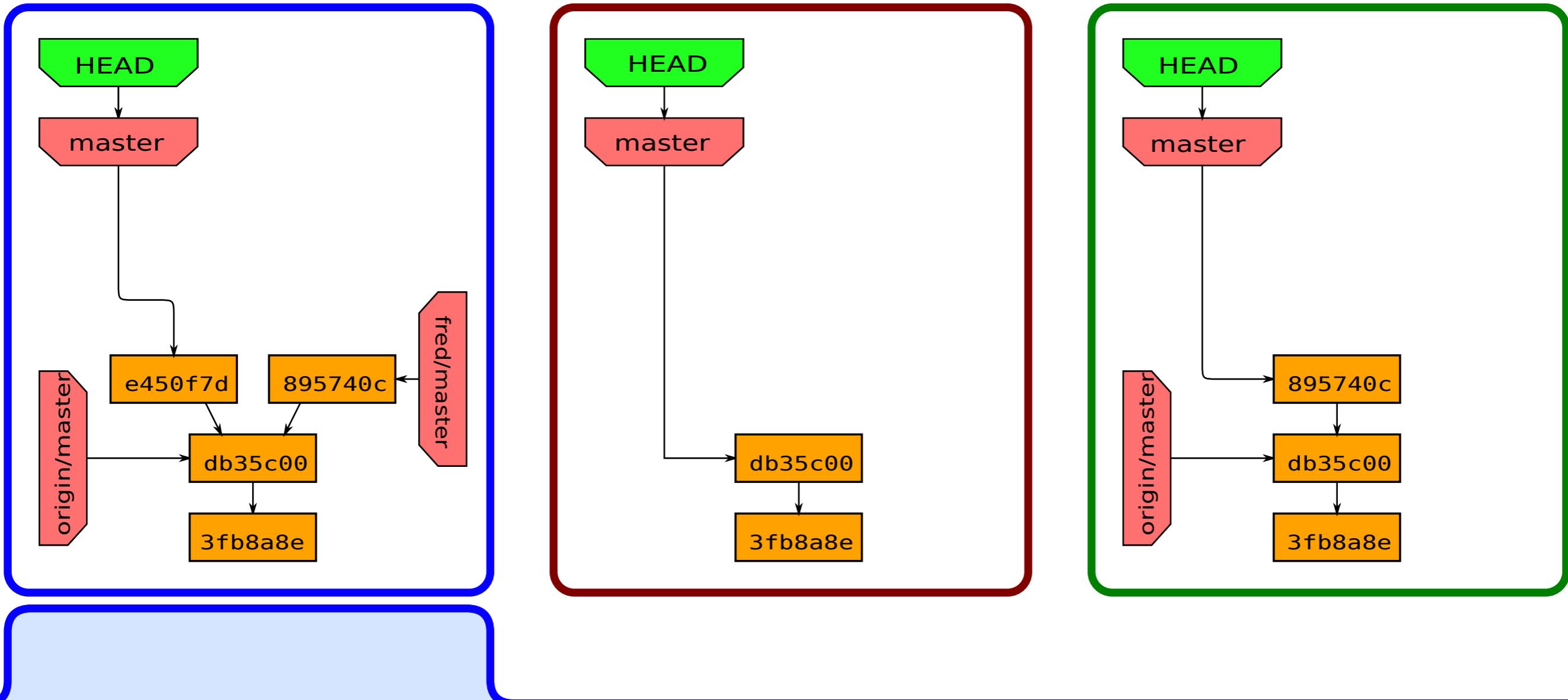
```
git log -p fred/master
```

```
git checkout -b fred fred/master
```

# local

# “origin” github

# Fred



```
git log -p fred/master
```

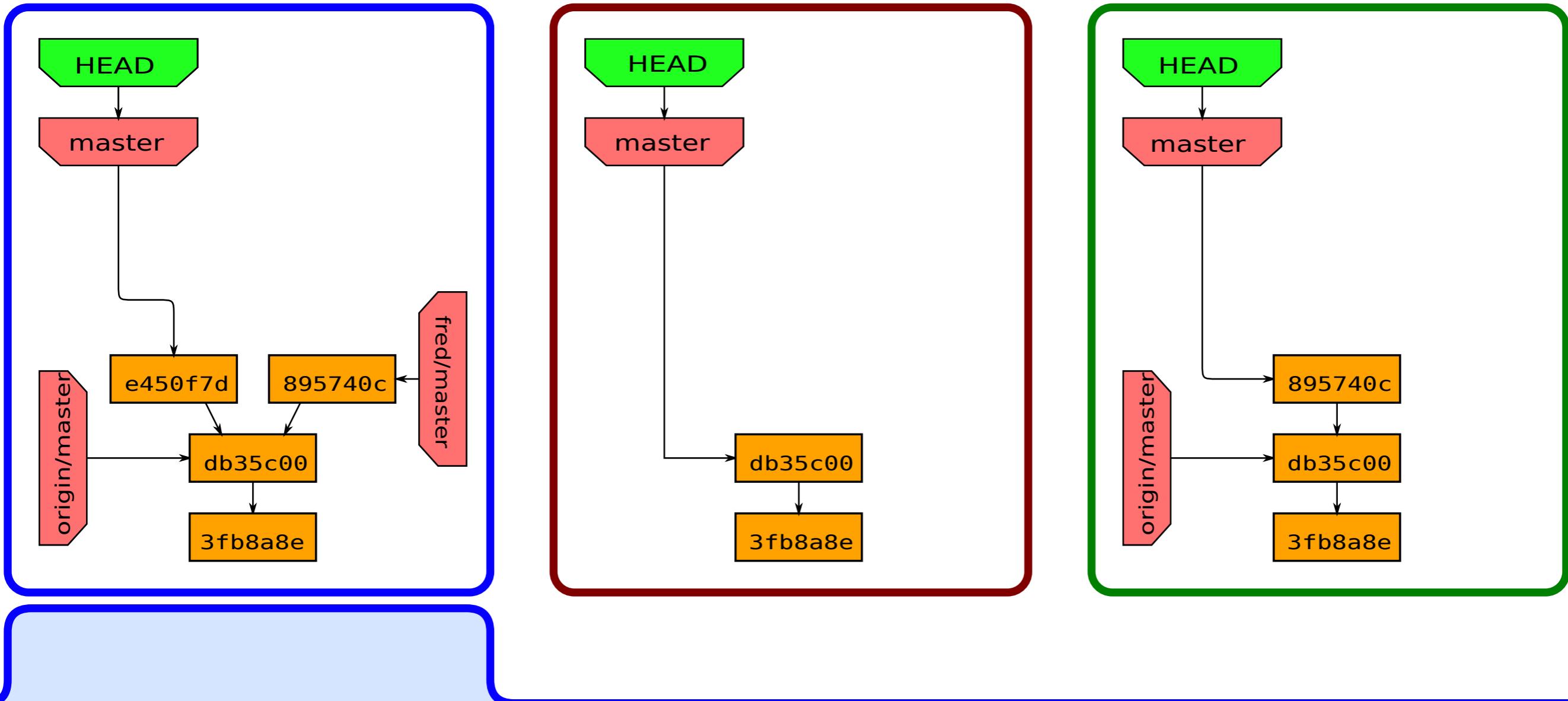
```
git checkout -b fred fred/master
```

```
gitk --all
```

# local

# “origin” github

# Fred

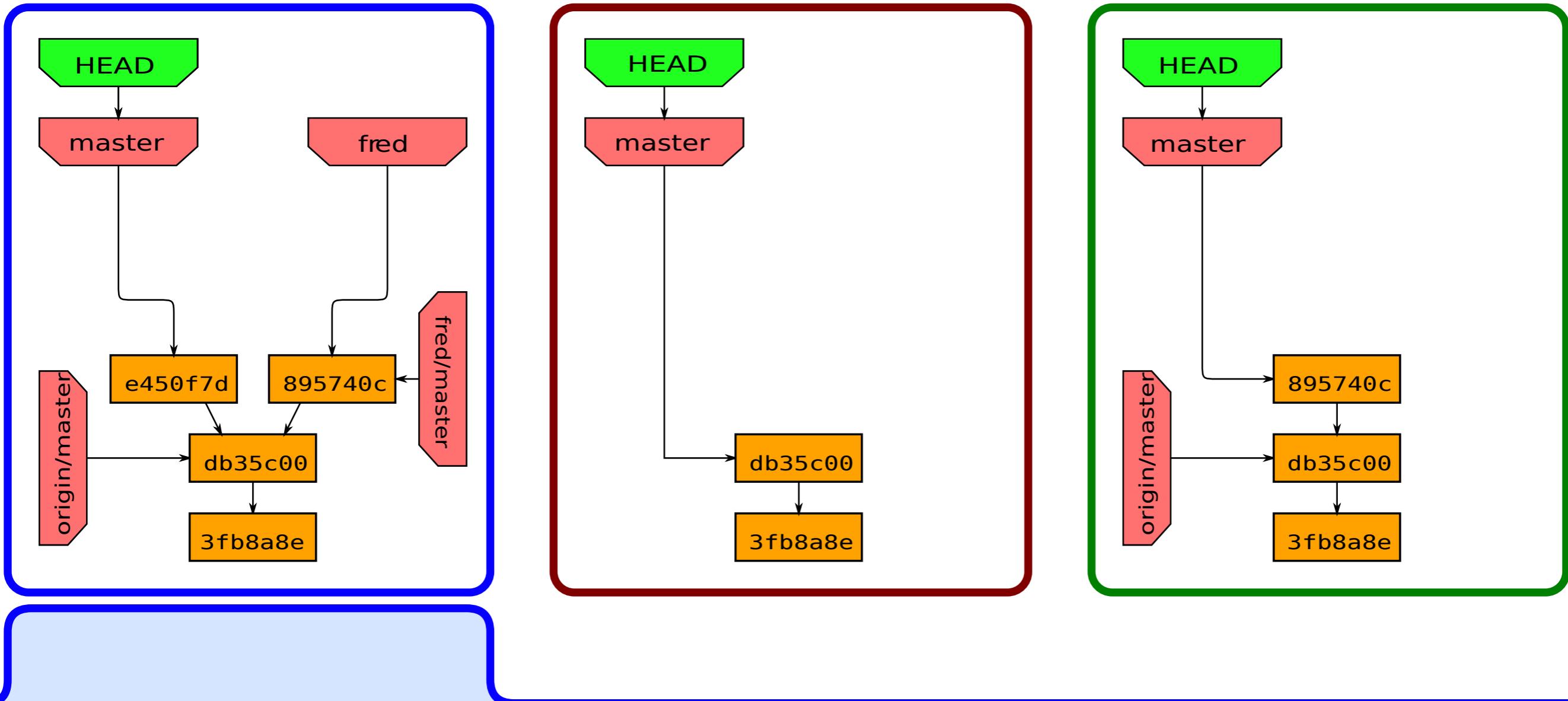


```
git checkout -b fred fred/master
```

# local

# “origin” github

# Fred

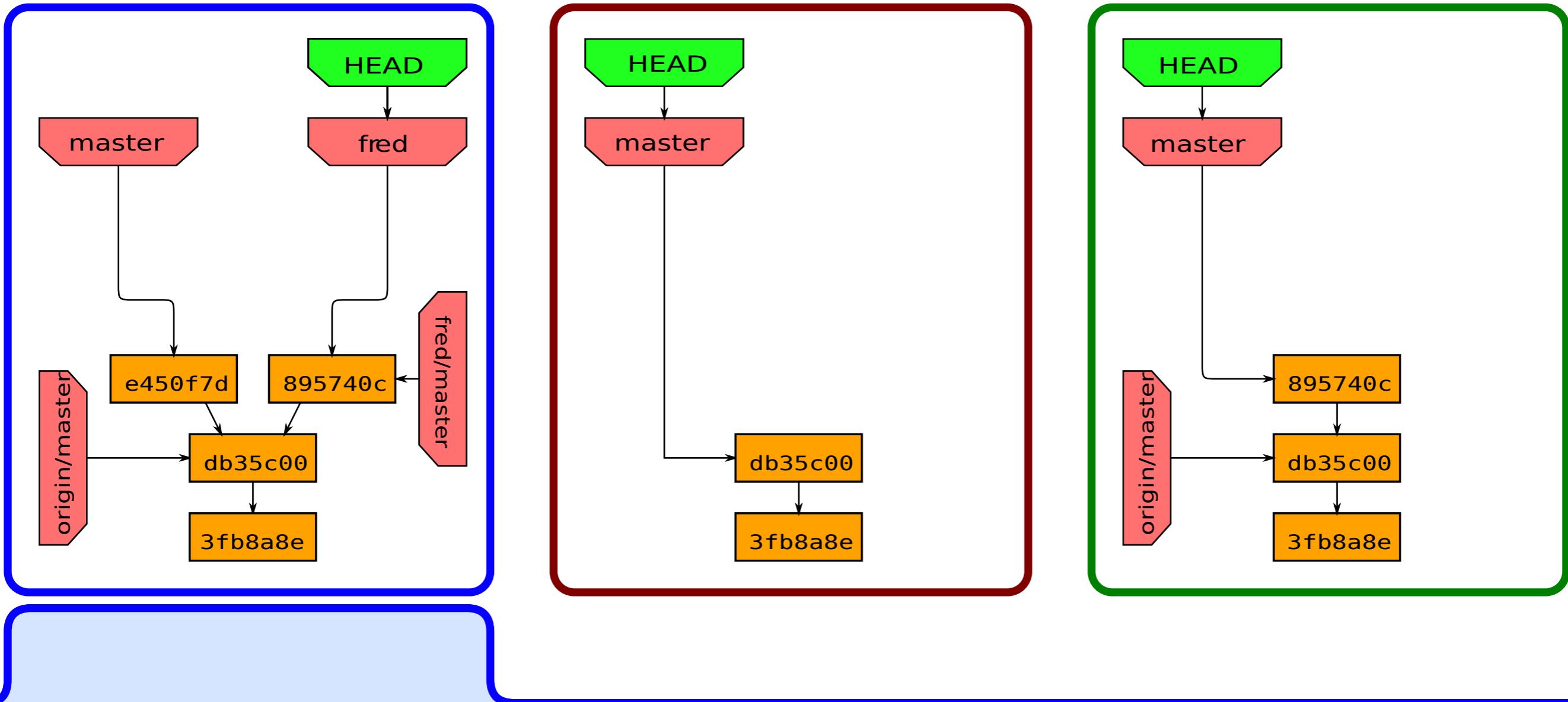


git checkout -b fred fred/master

# local

# “origin” github

# Fred

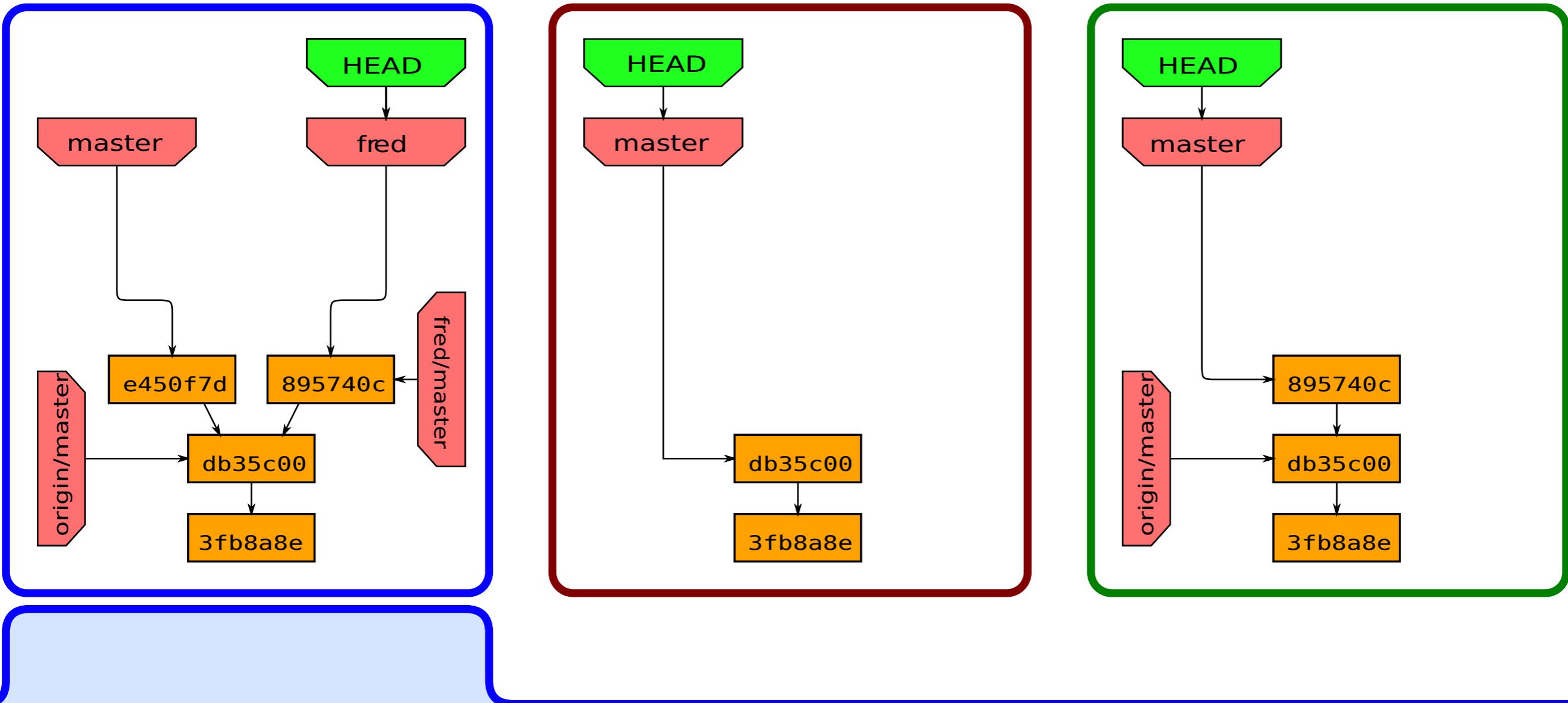


git checkout -b fred fred/master

# local

# “origin” github

# Fred



git log

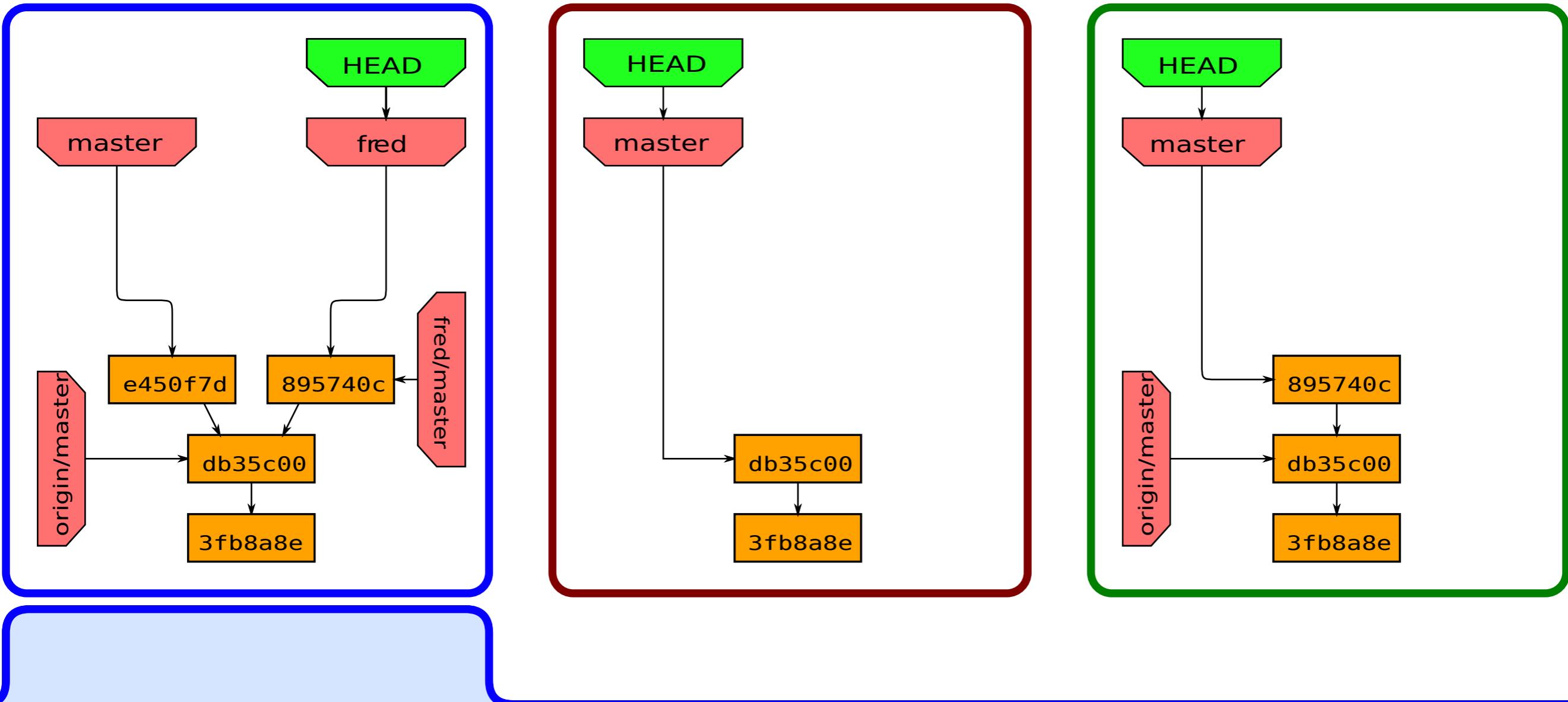
git show

git diff master

# local

# “origin” github

# Fred

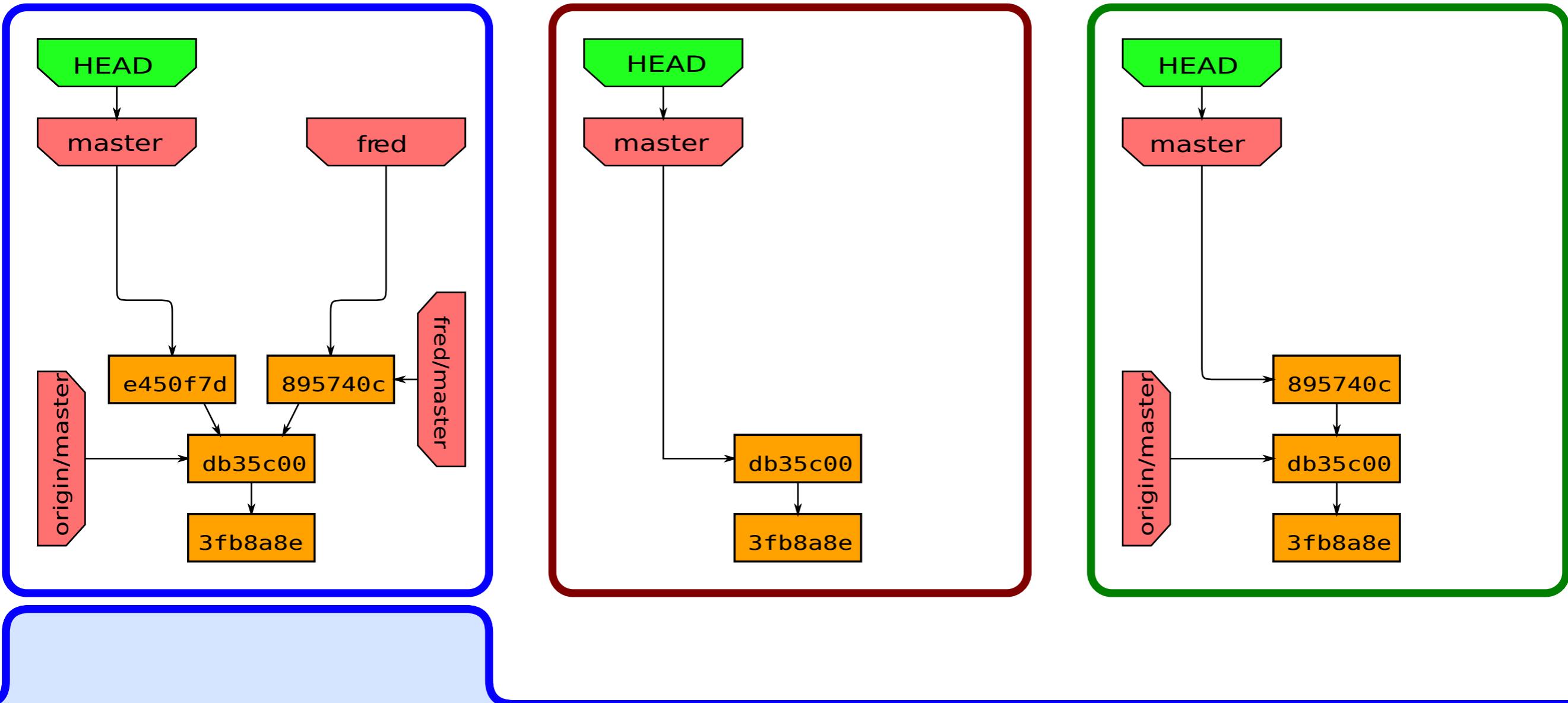


git checkout master

# local

# “origin” github

# Fred

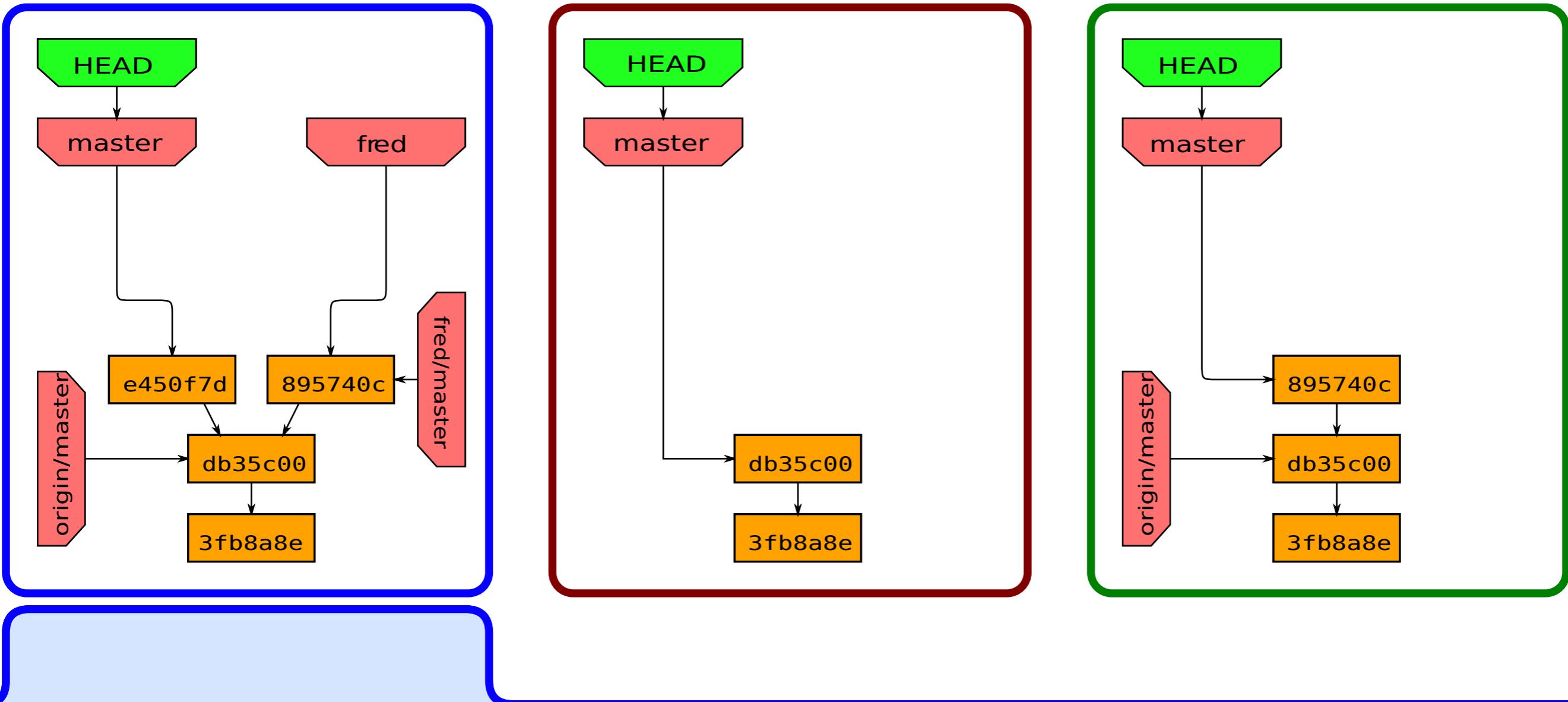


git checkout master

# local

# “origin” github

# Fred

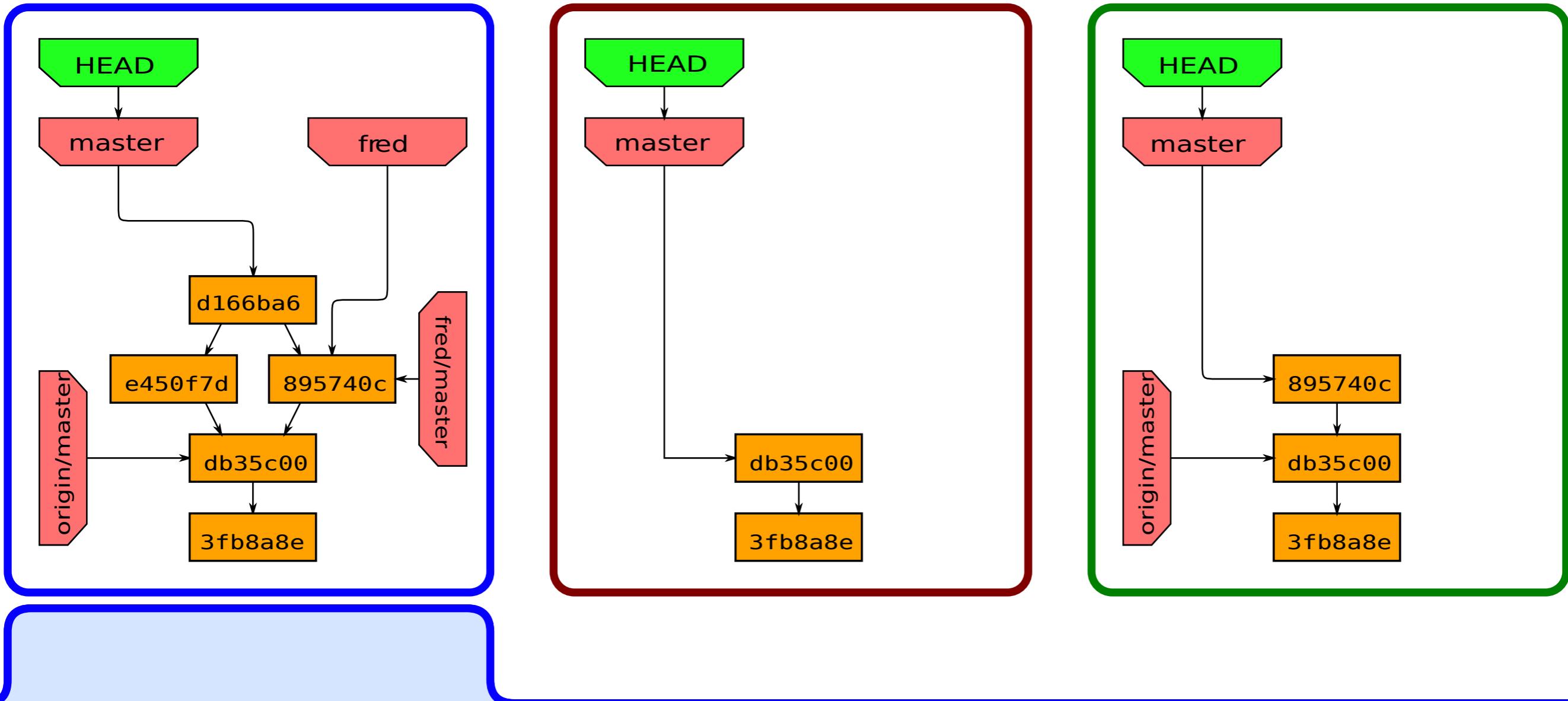


git merge fred

# local

# “origin” github

# Fred

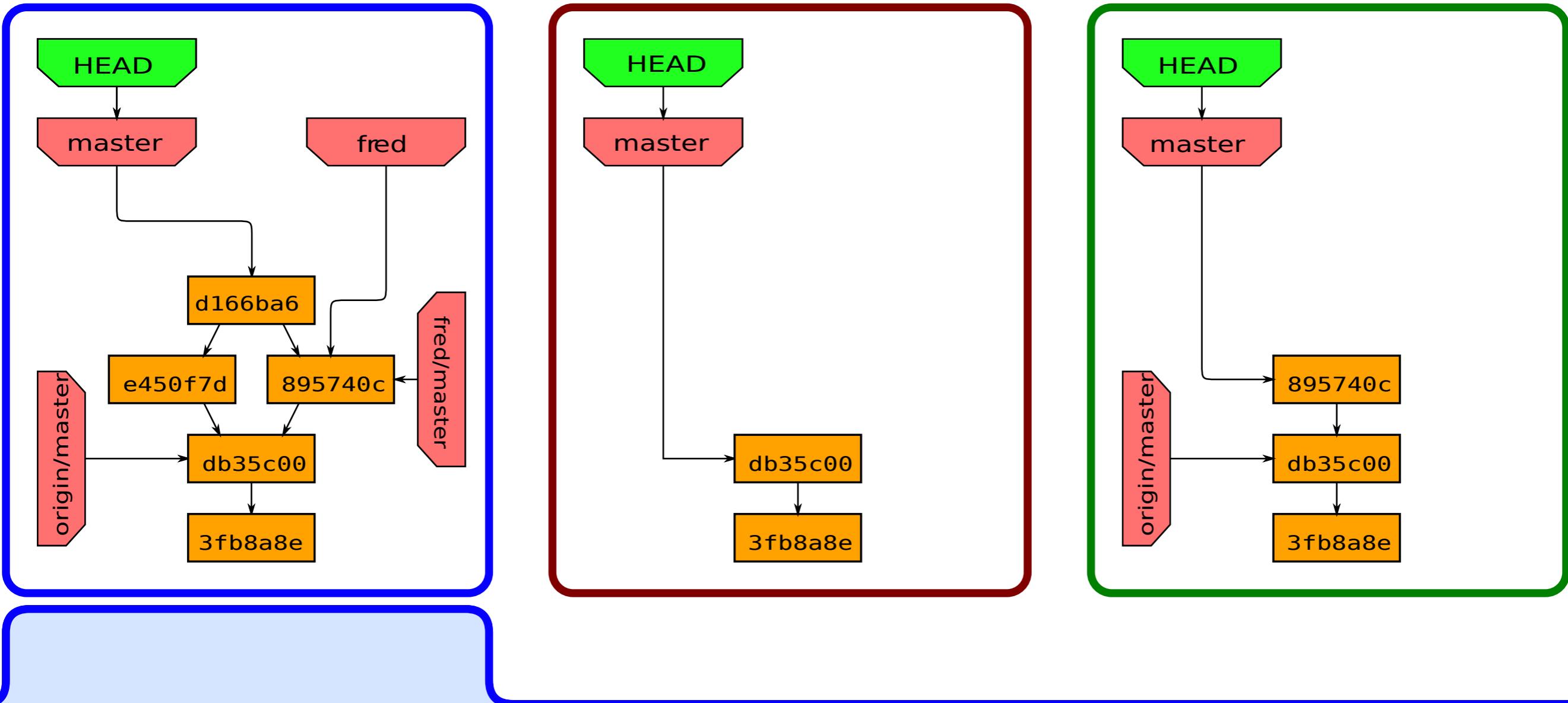


git merge fred

# local

# “origin” github

# Fred

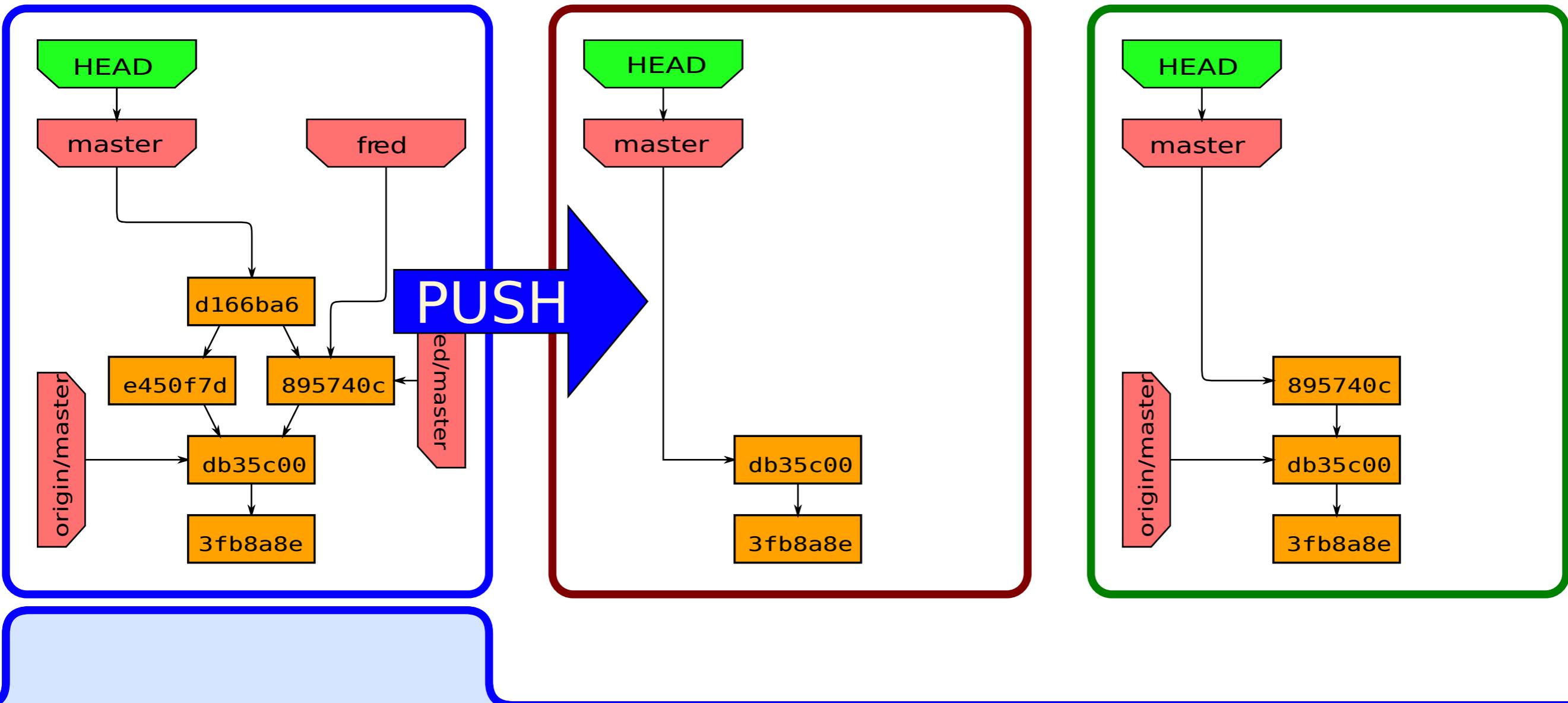


# git push

# local

# “origin” github

# Fred

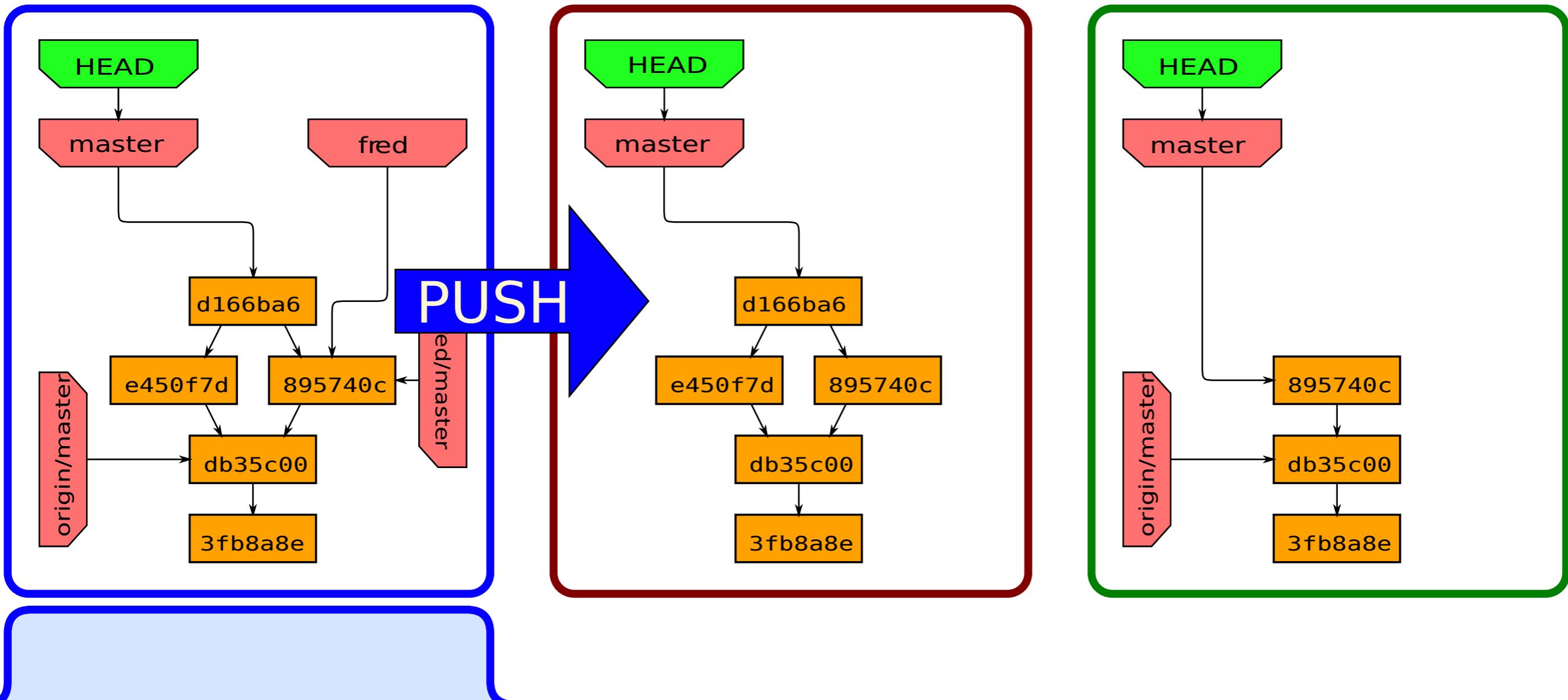


git push

# local

# “origin” github

# Fred

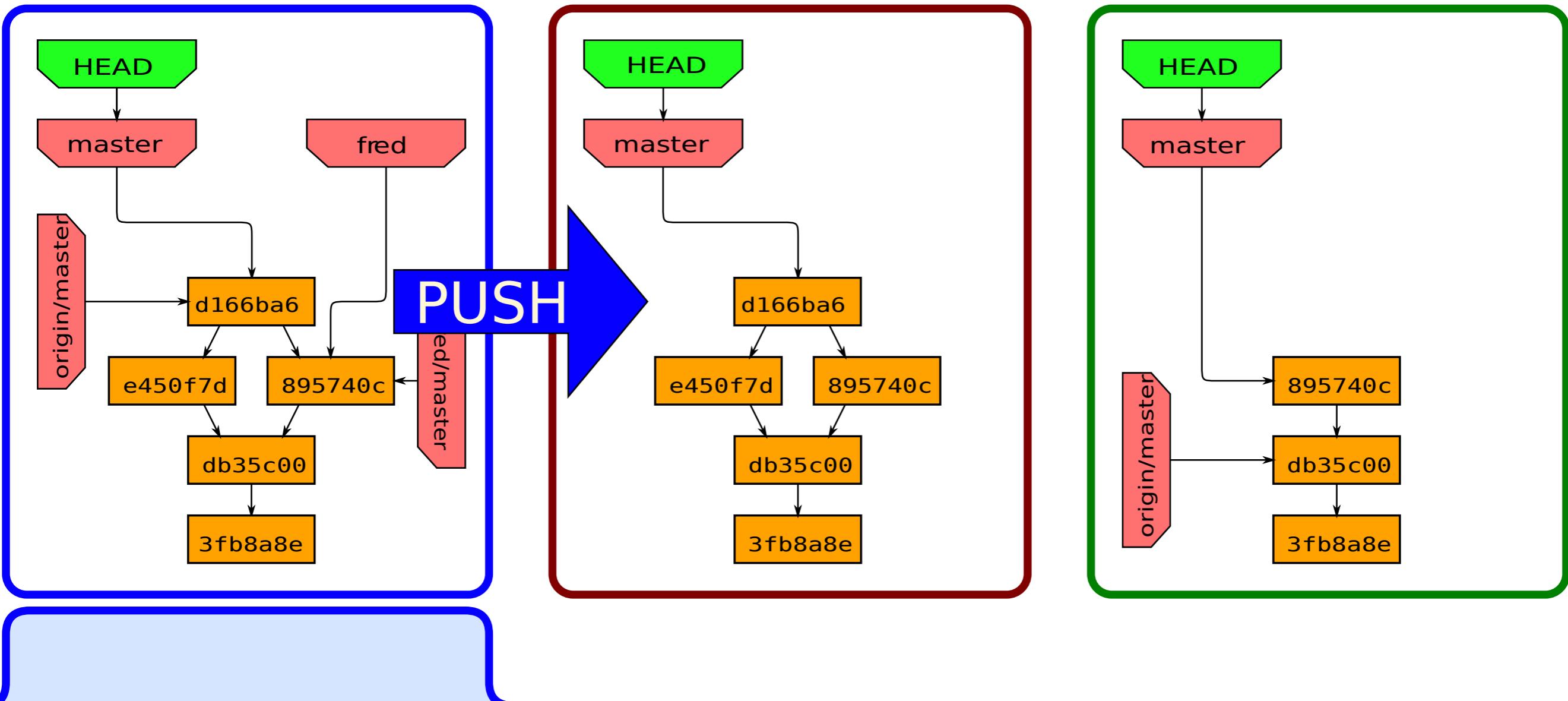


git push

# local

# “origin” github

# Fred

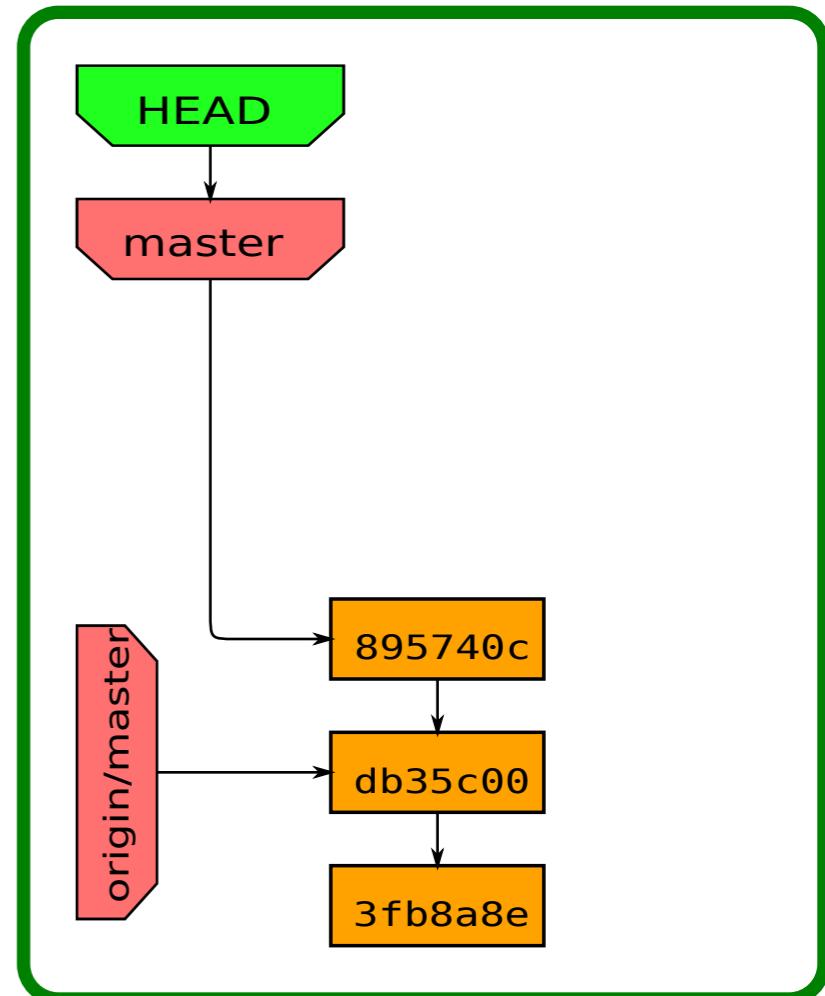
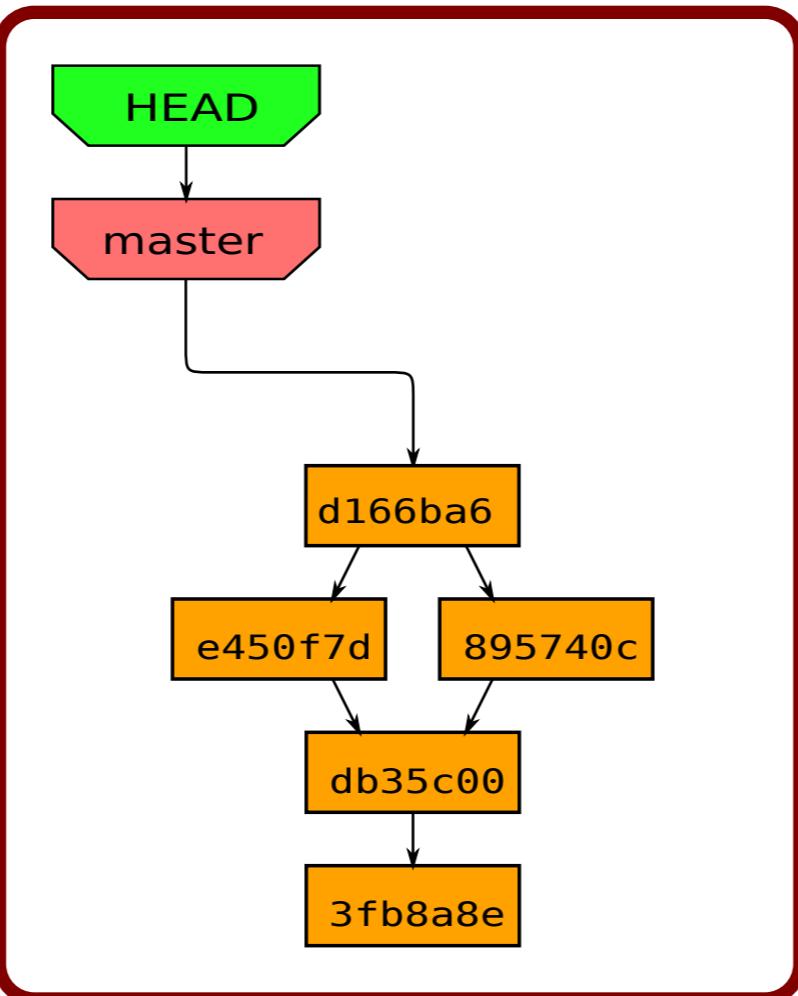
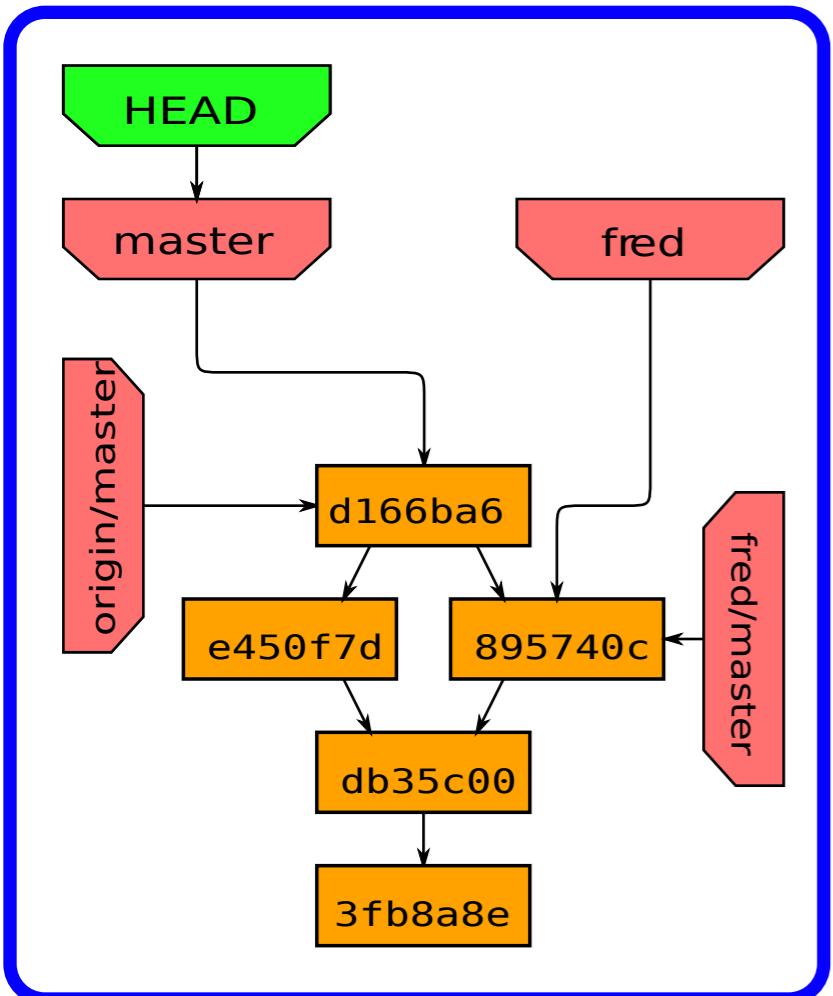


git push

# local

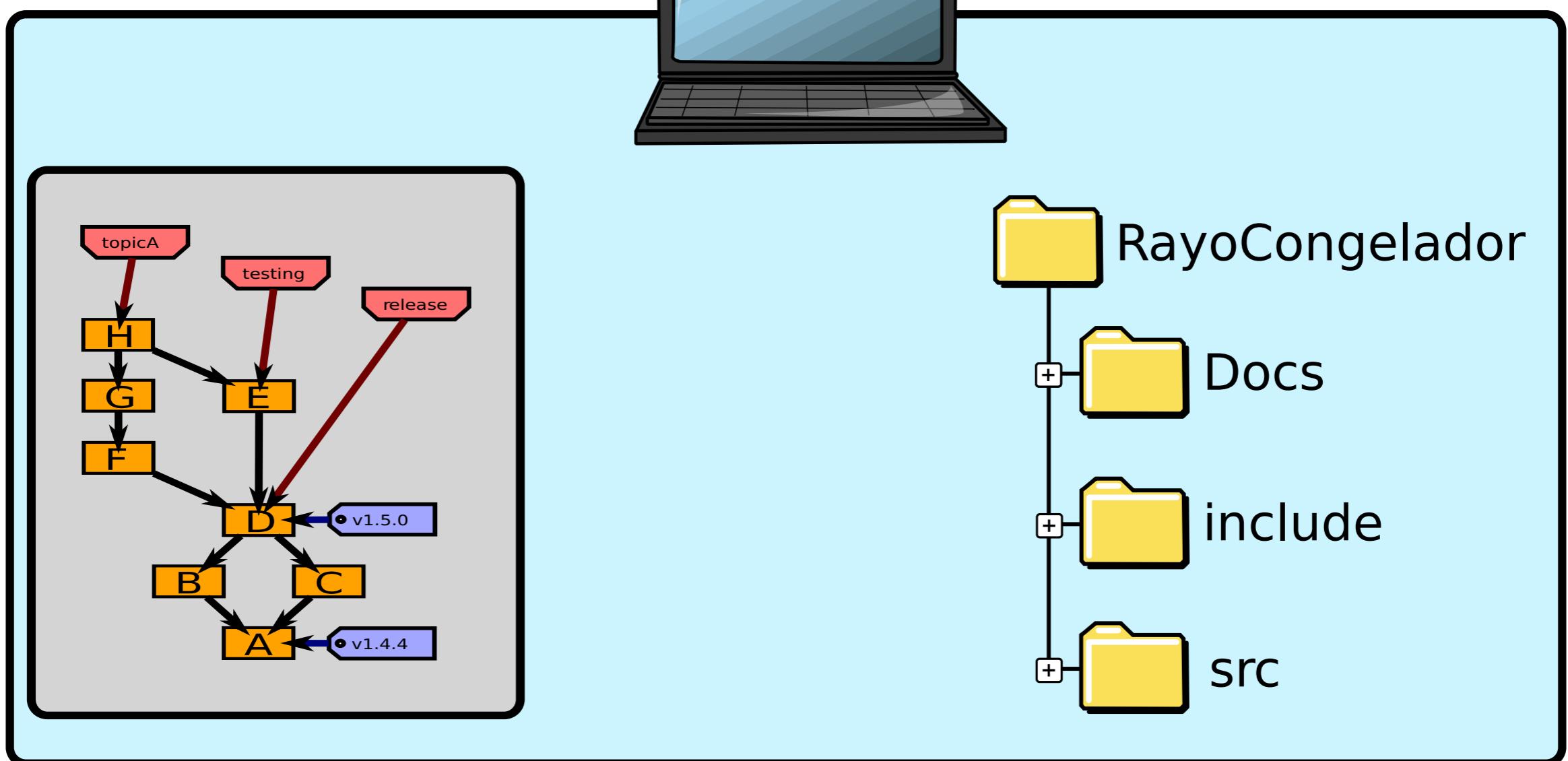
# “origin” github

# Fred

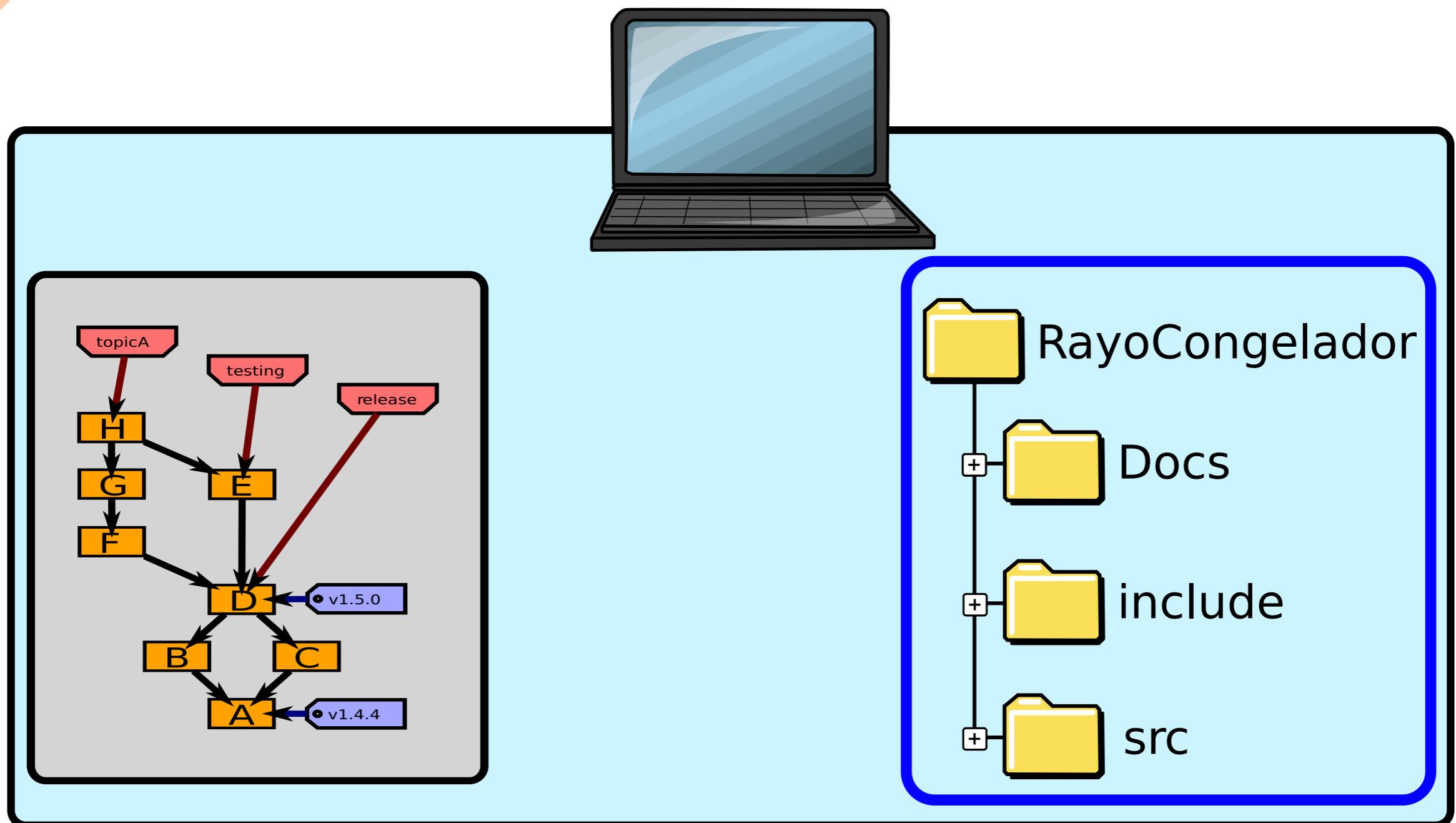


# Topologías

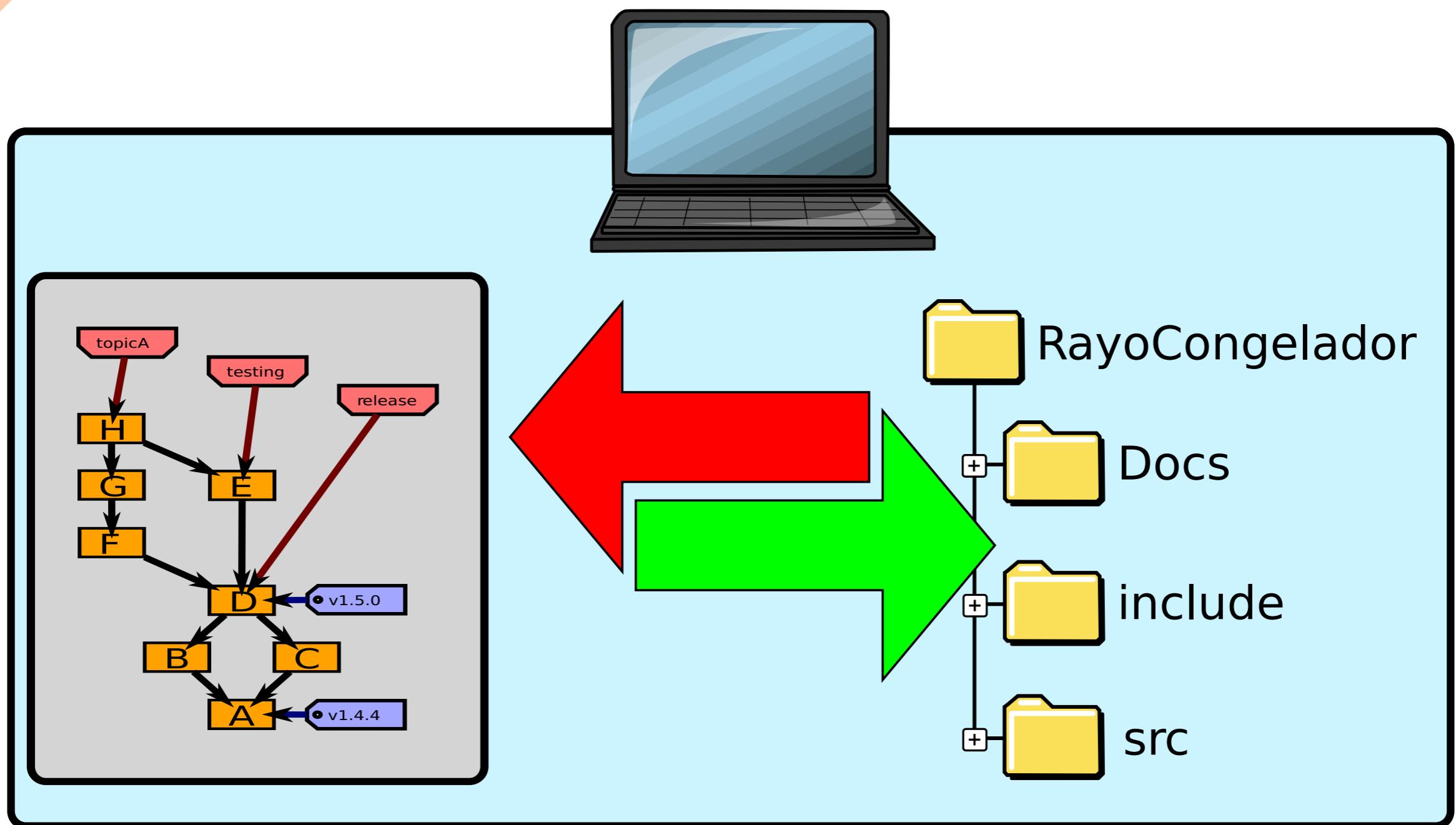
# Un solo desarrollador



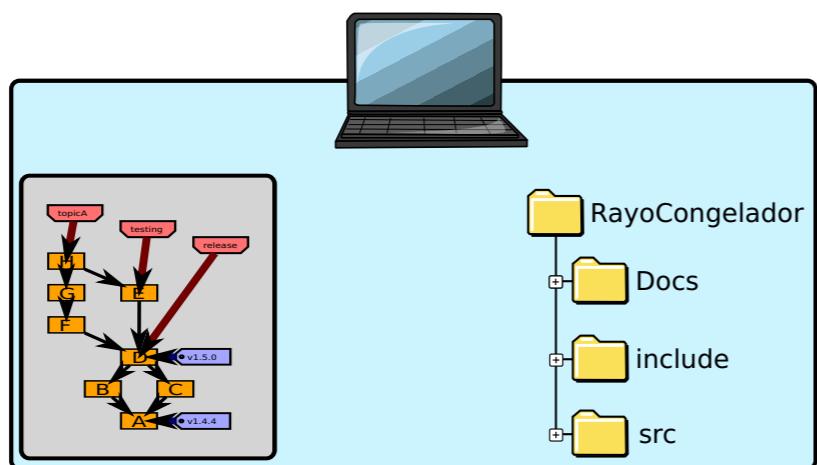
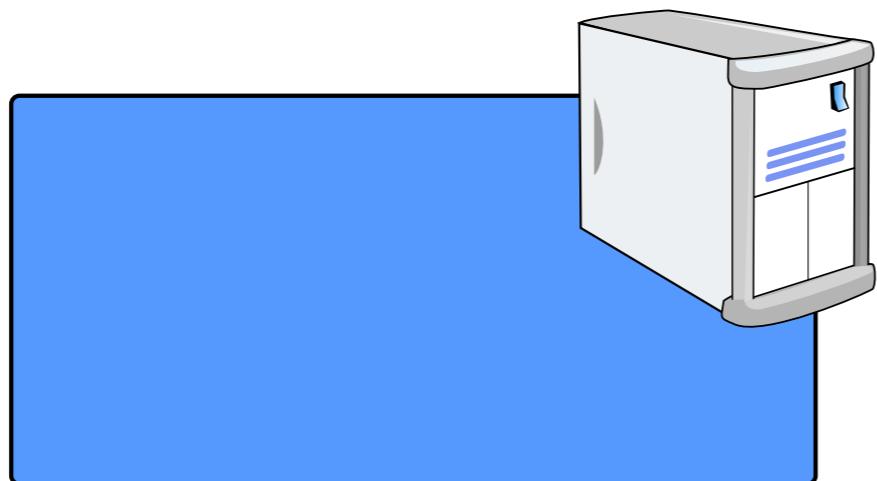
# Un solo desarrollador



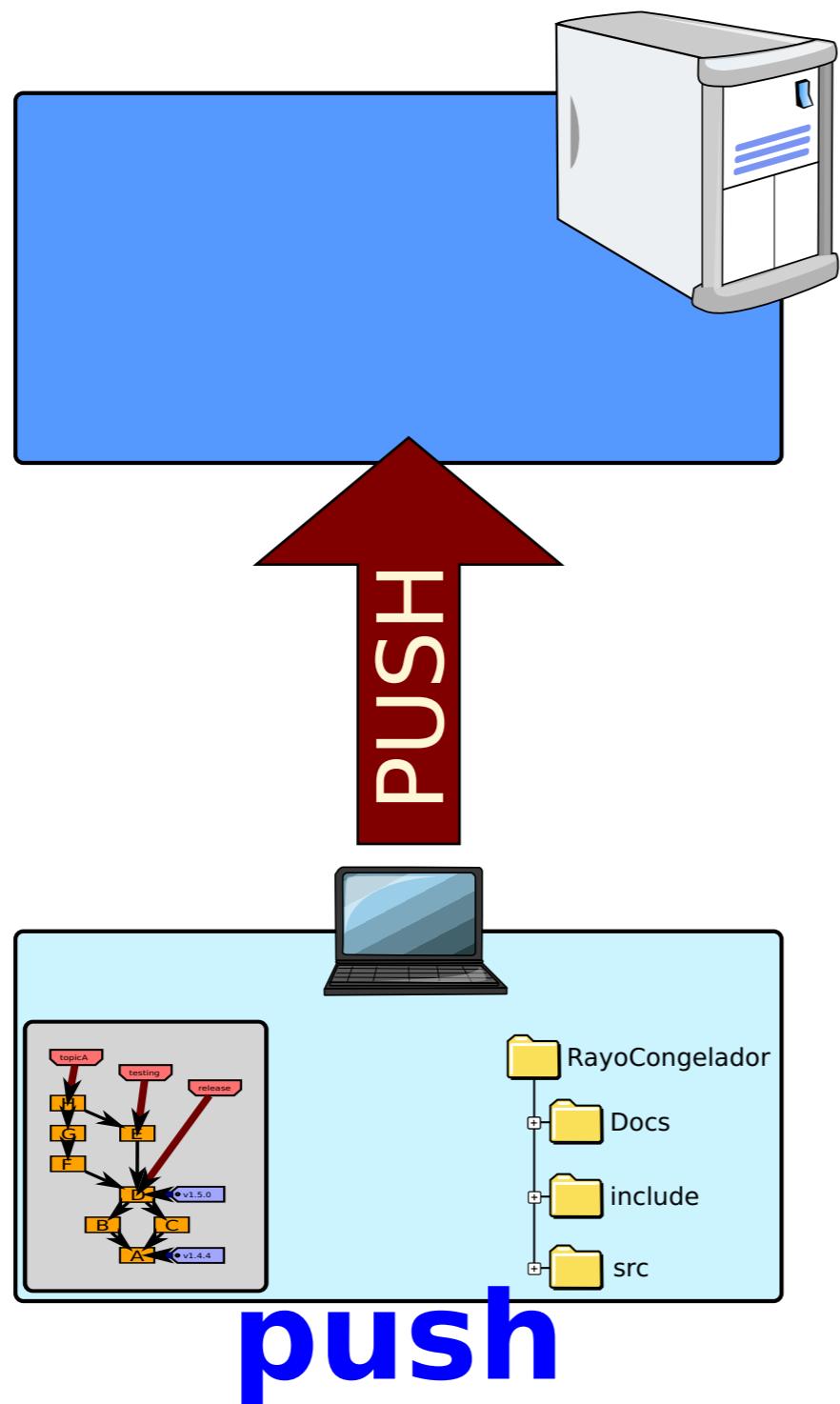
# Un solo desarrollador



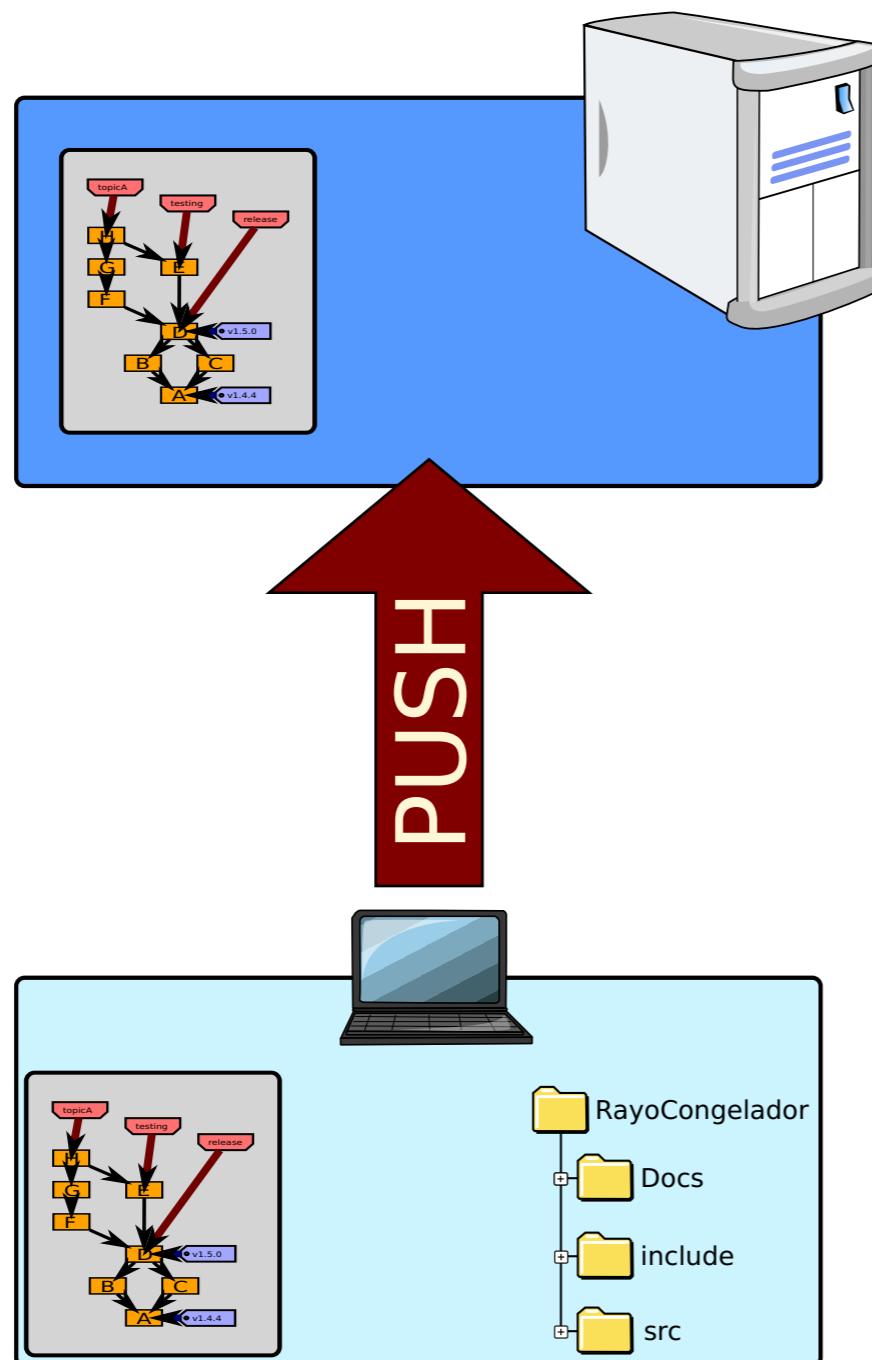
# Publicando a otro repositorio



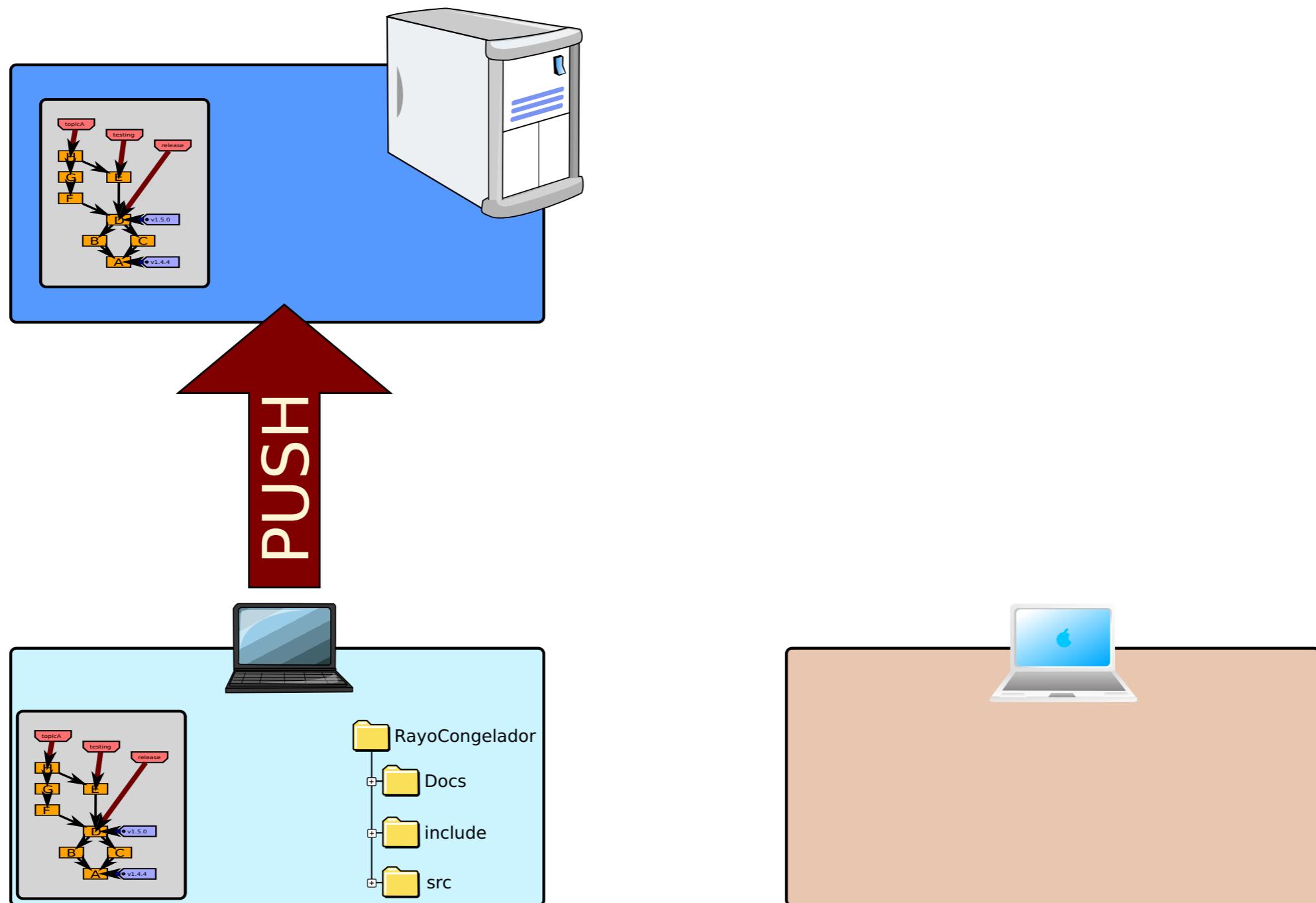
# Publicando a otro repositorio



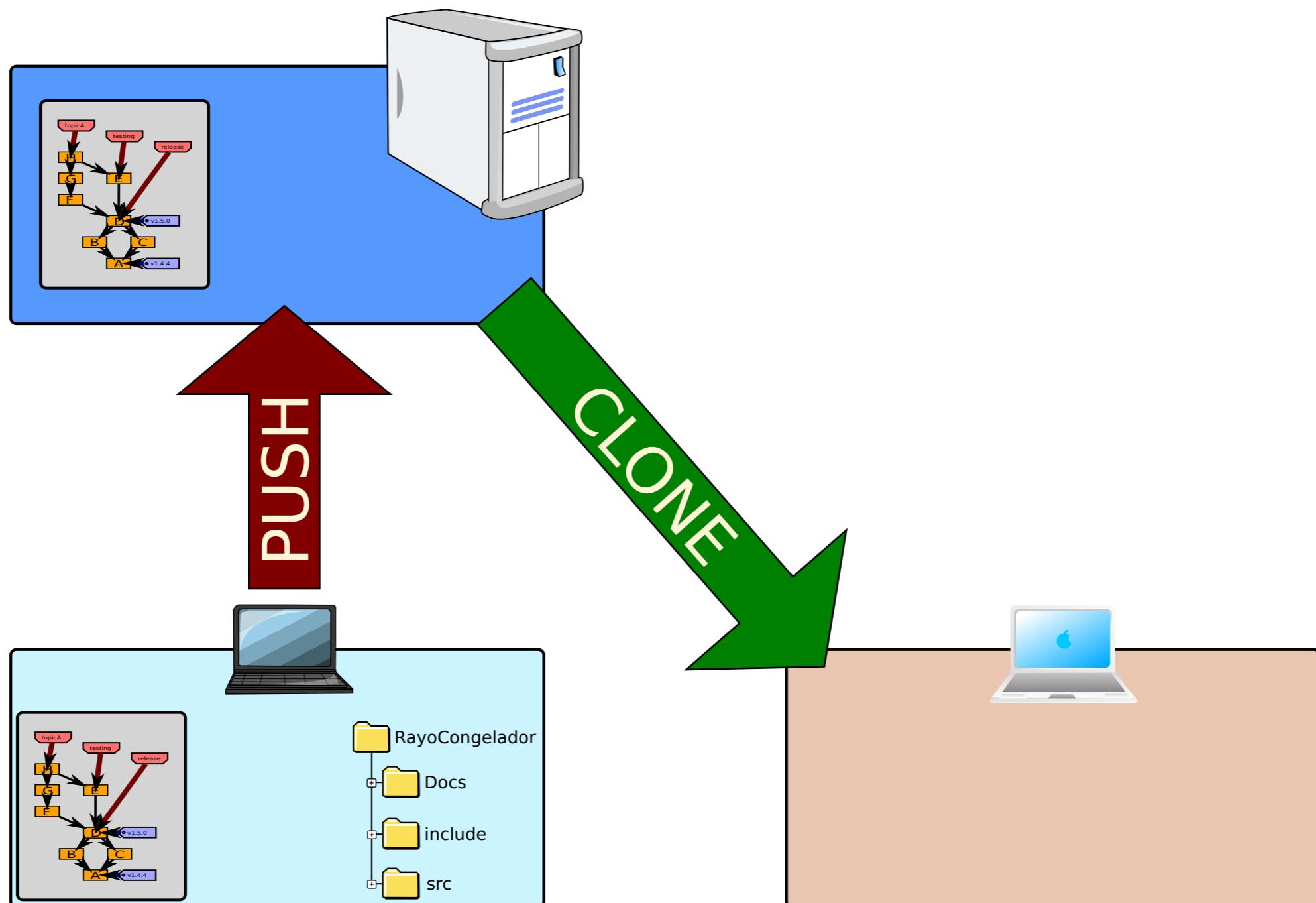
# Publicando a otro repositorio



# Publicando a otro repositorio

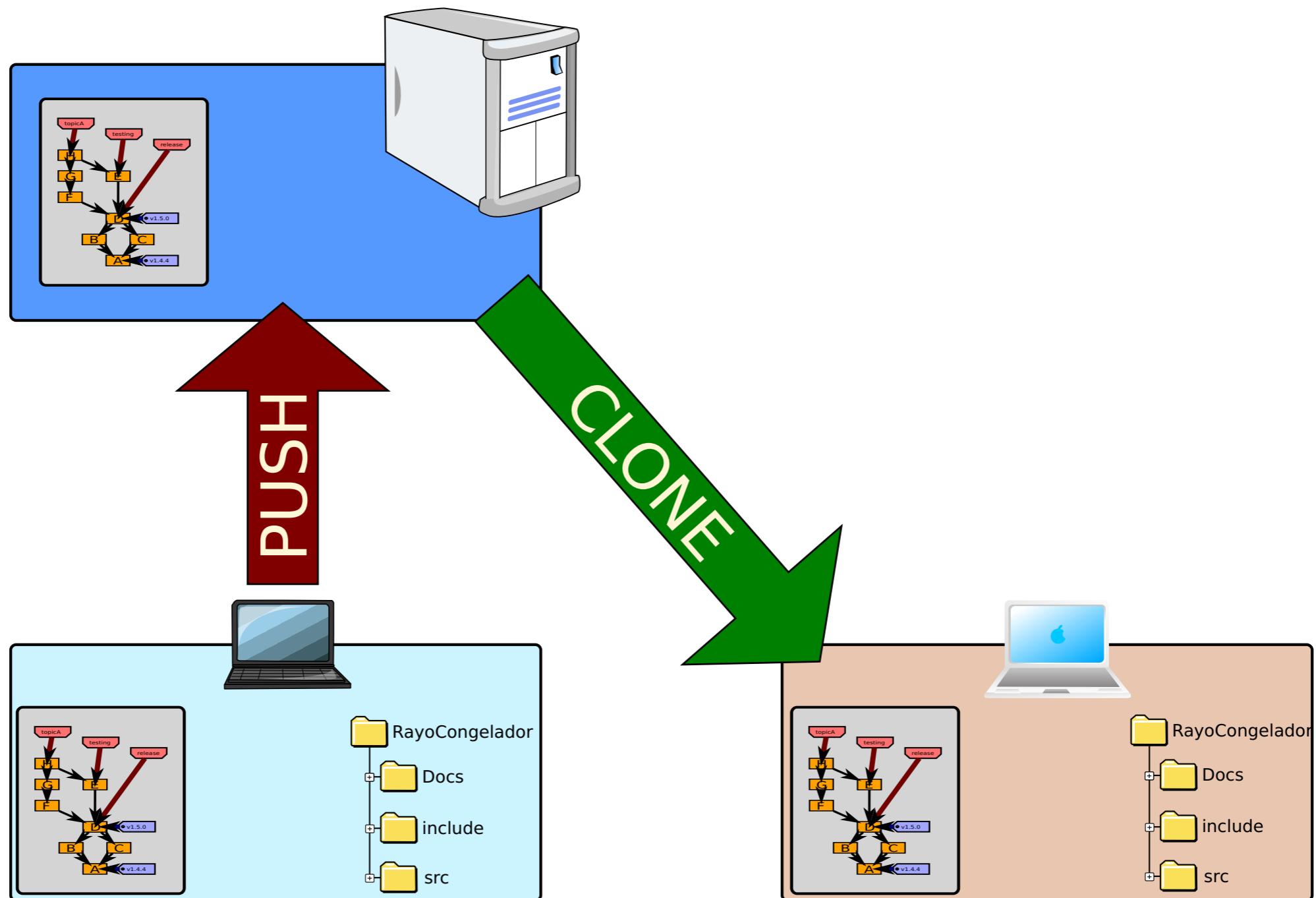


# Publicando a otro repositorio

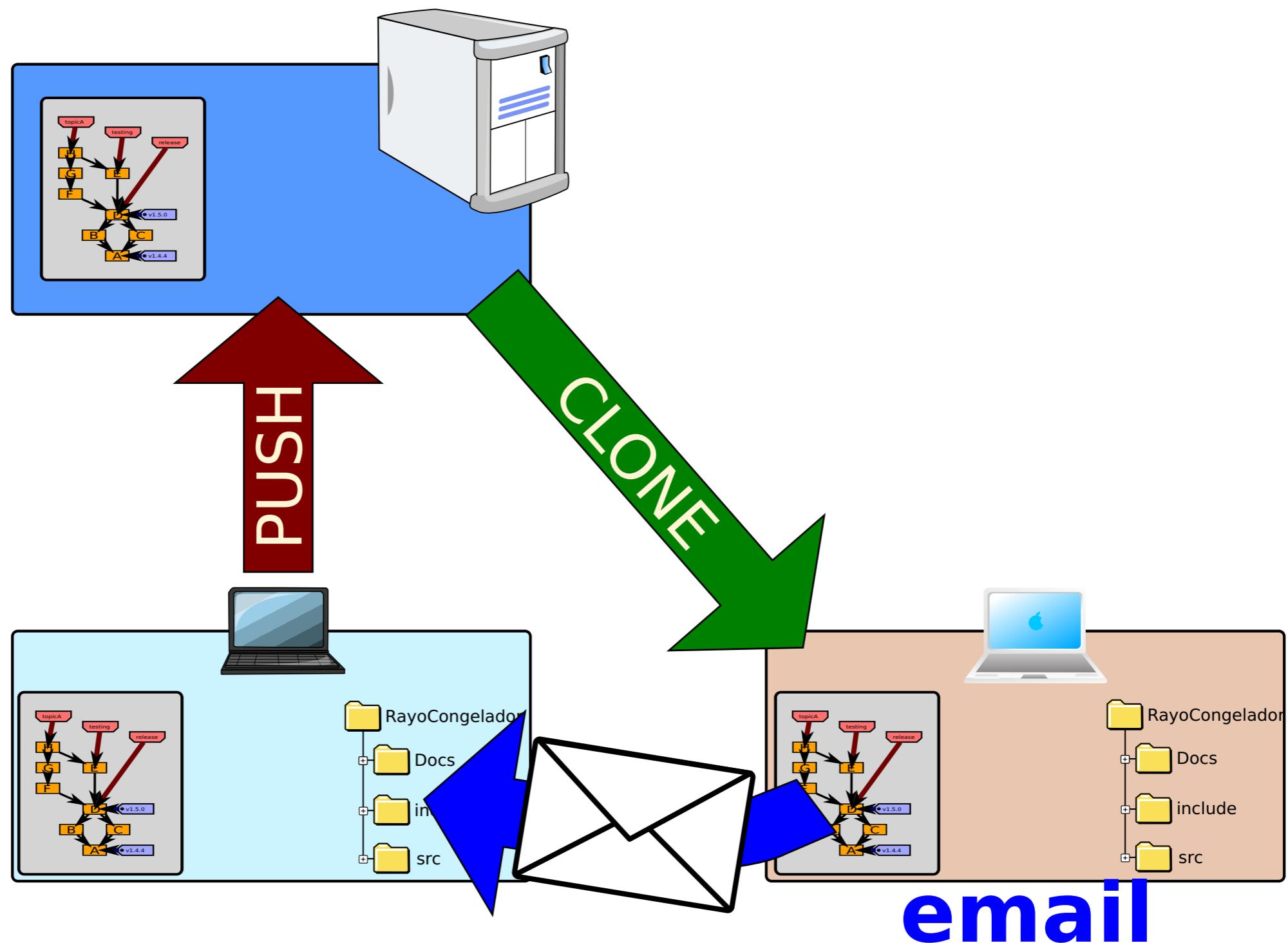


**clone**

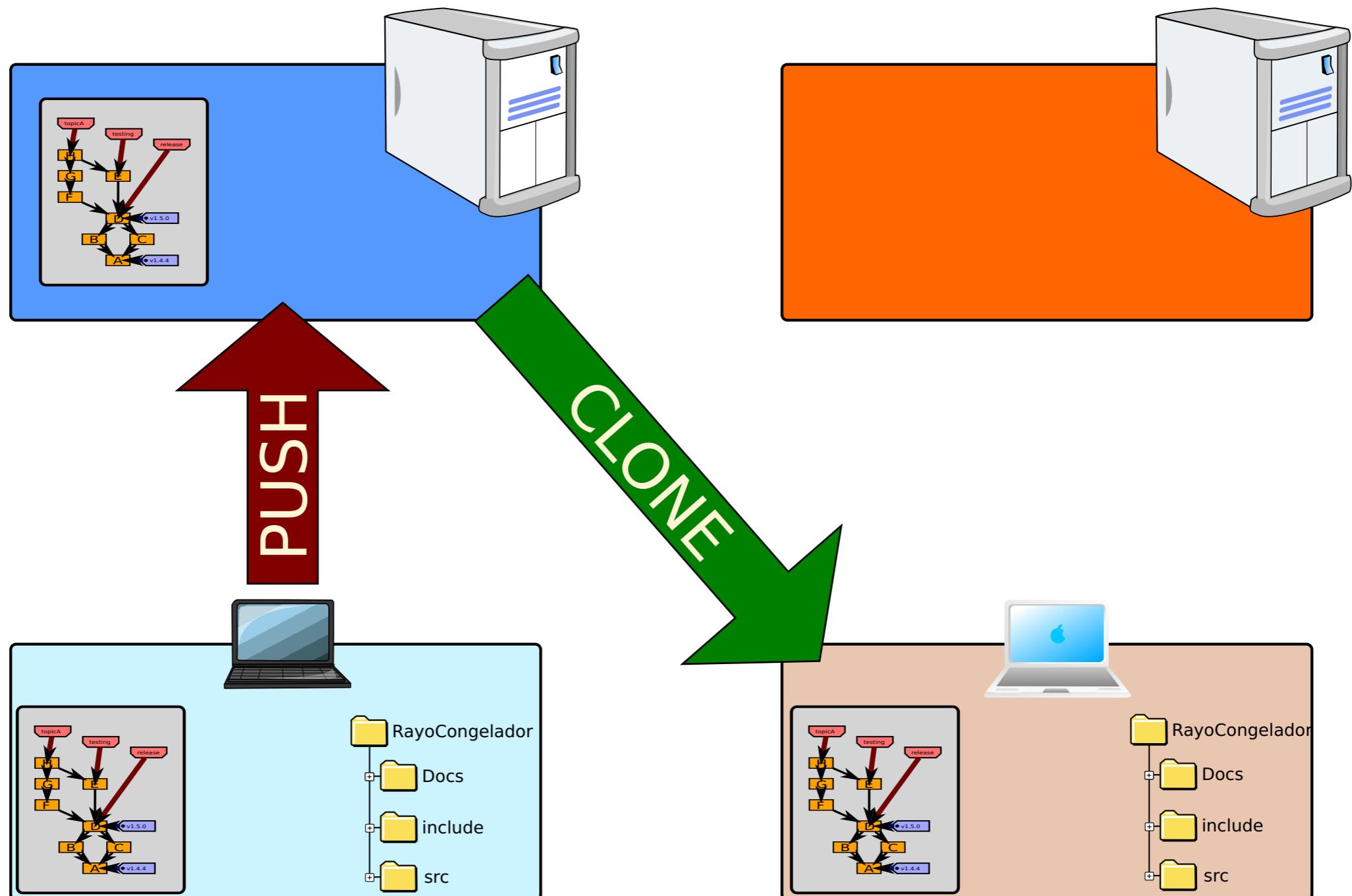
# Publicando a otro repositorio



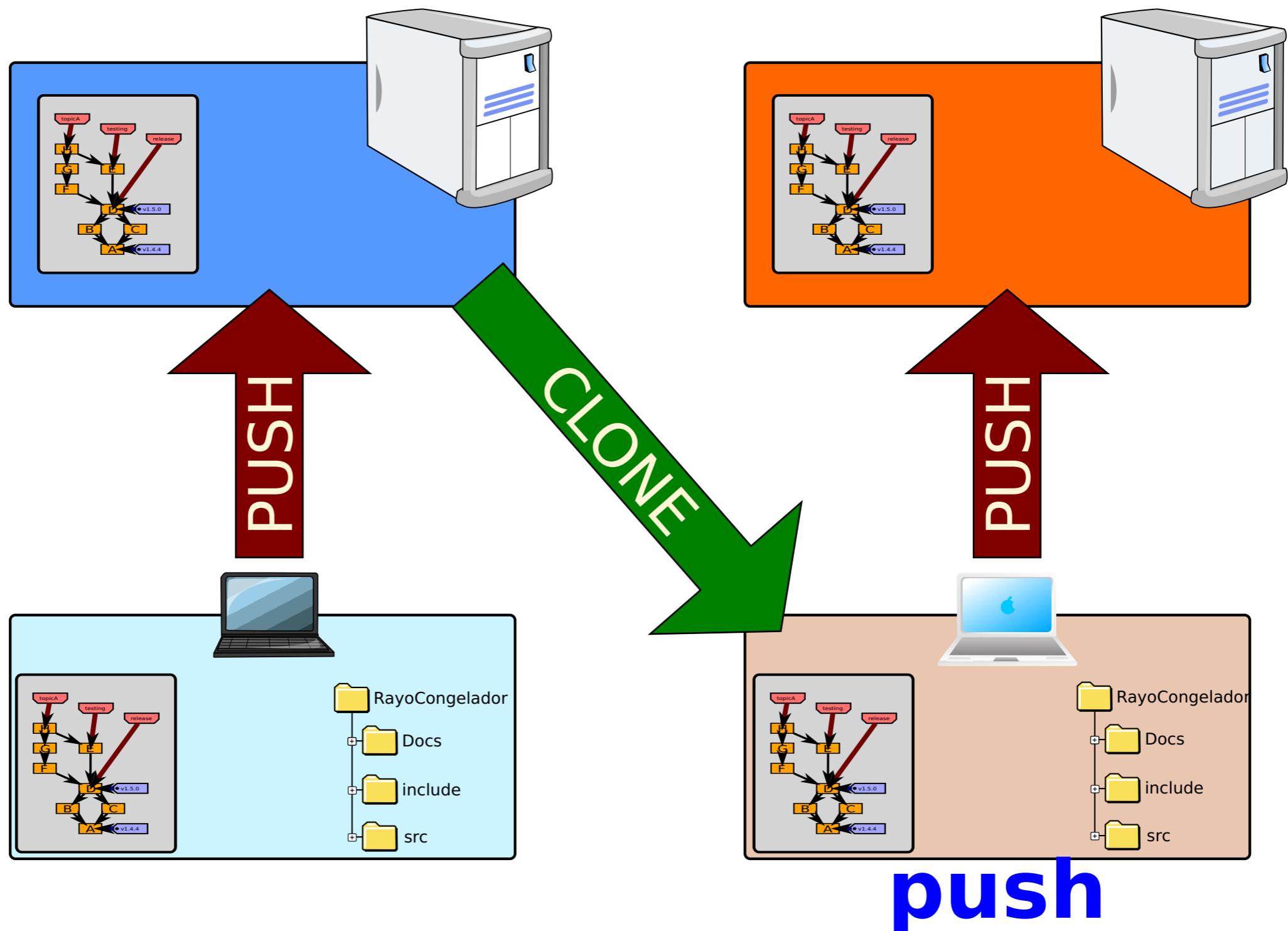
# Publicando a otro repositorio



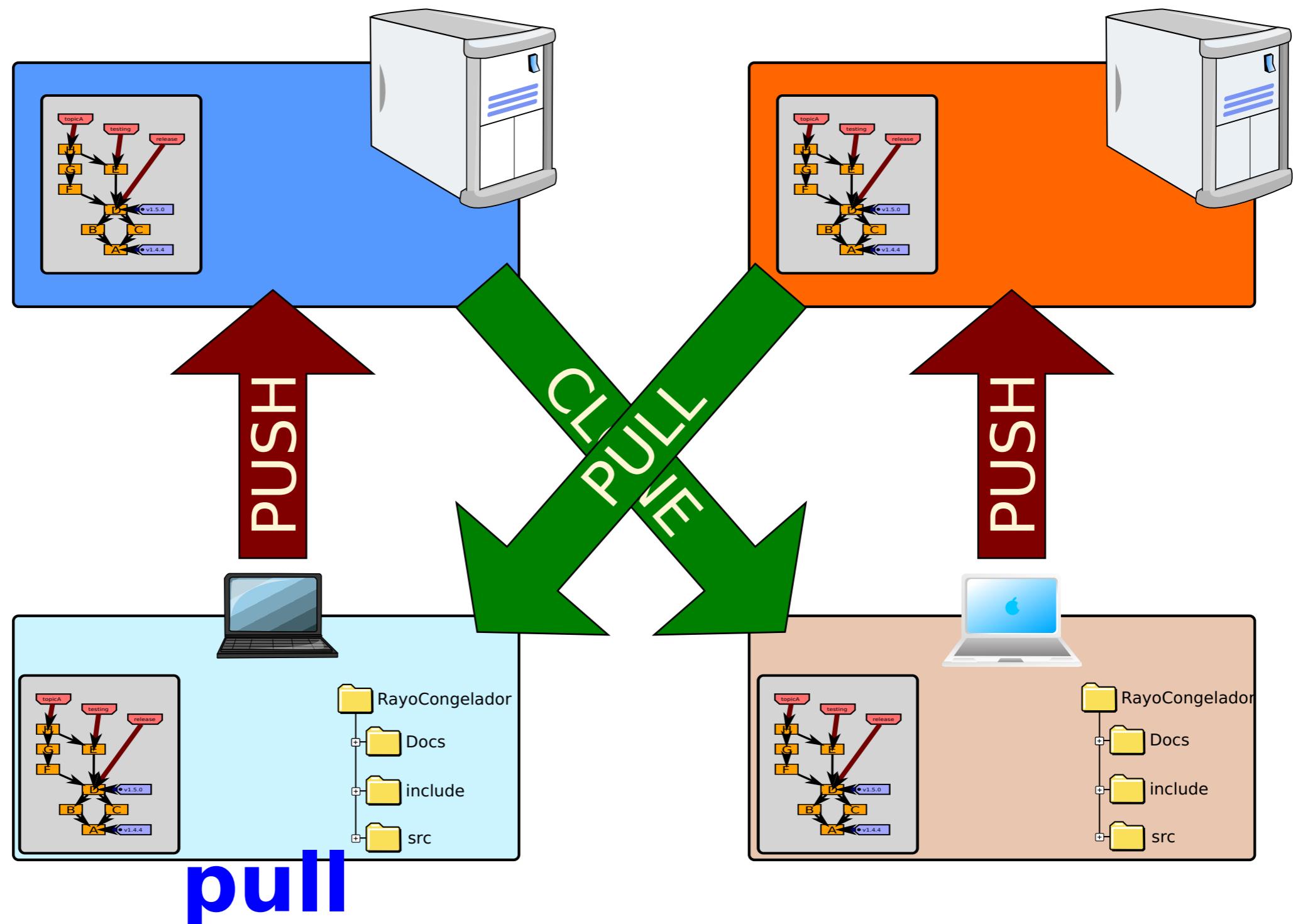
# Publicando a otro repositorio



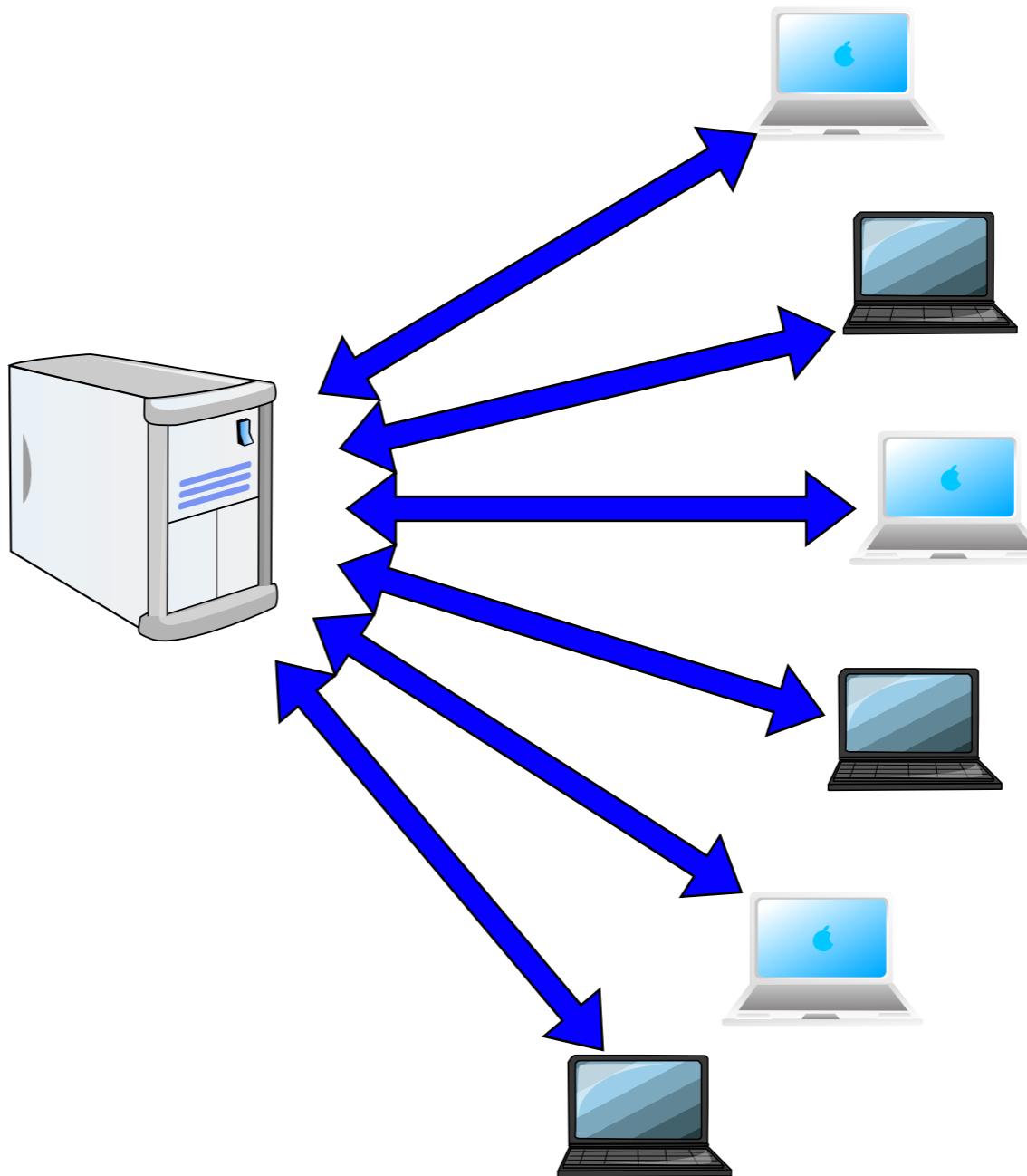
# Publicando a otro repositorio



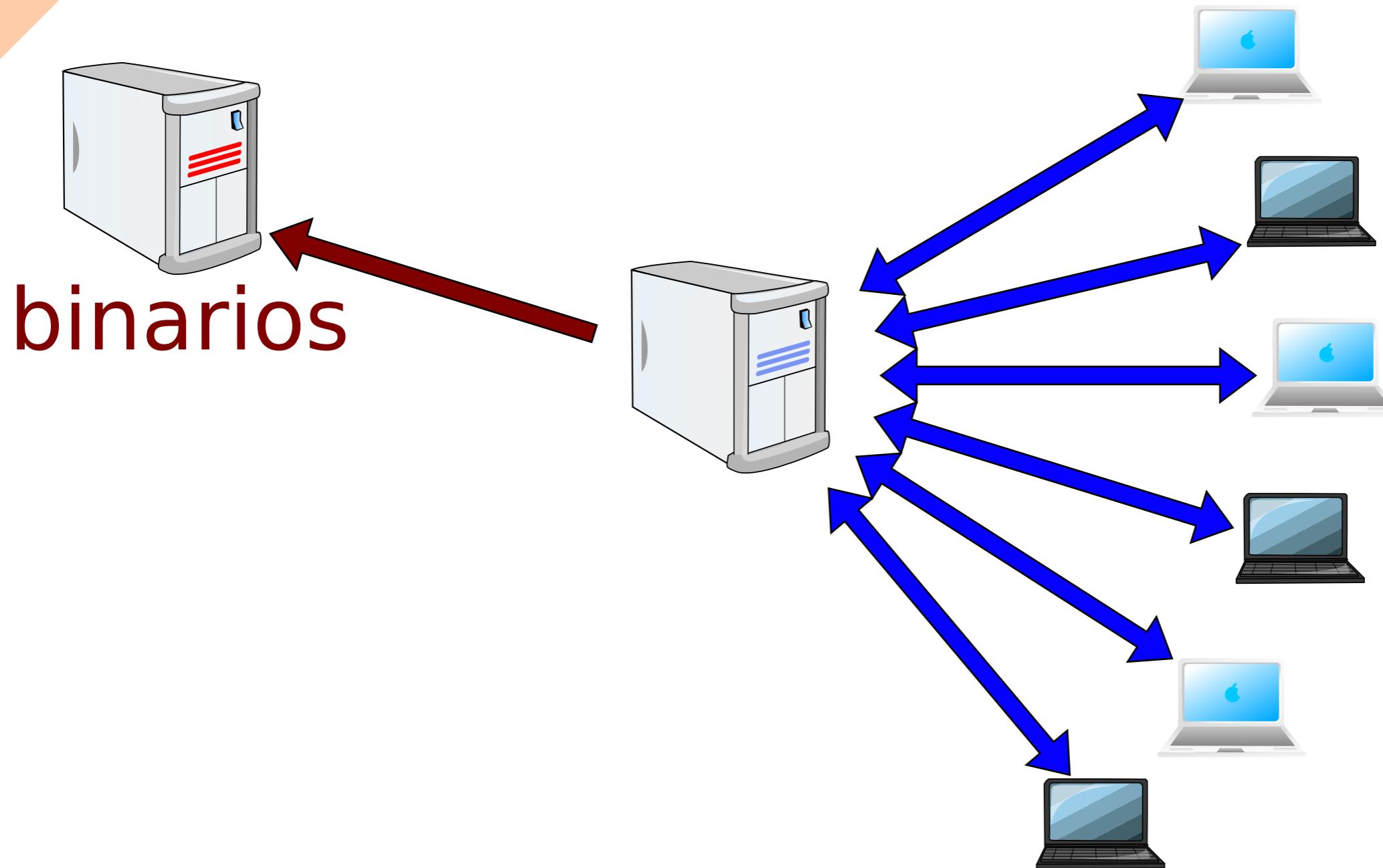
# Publicando a otro repositorio



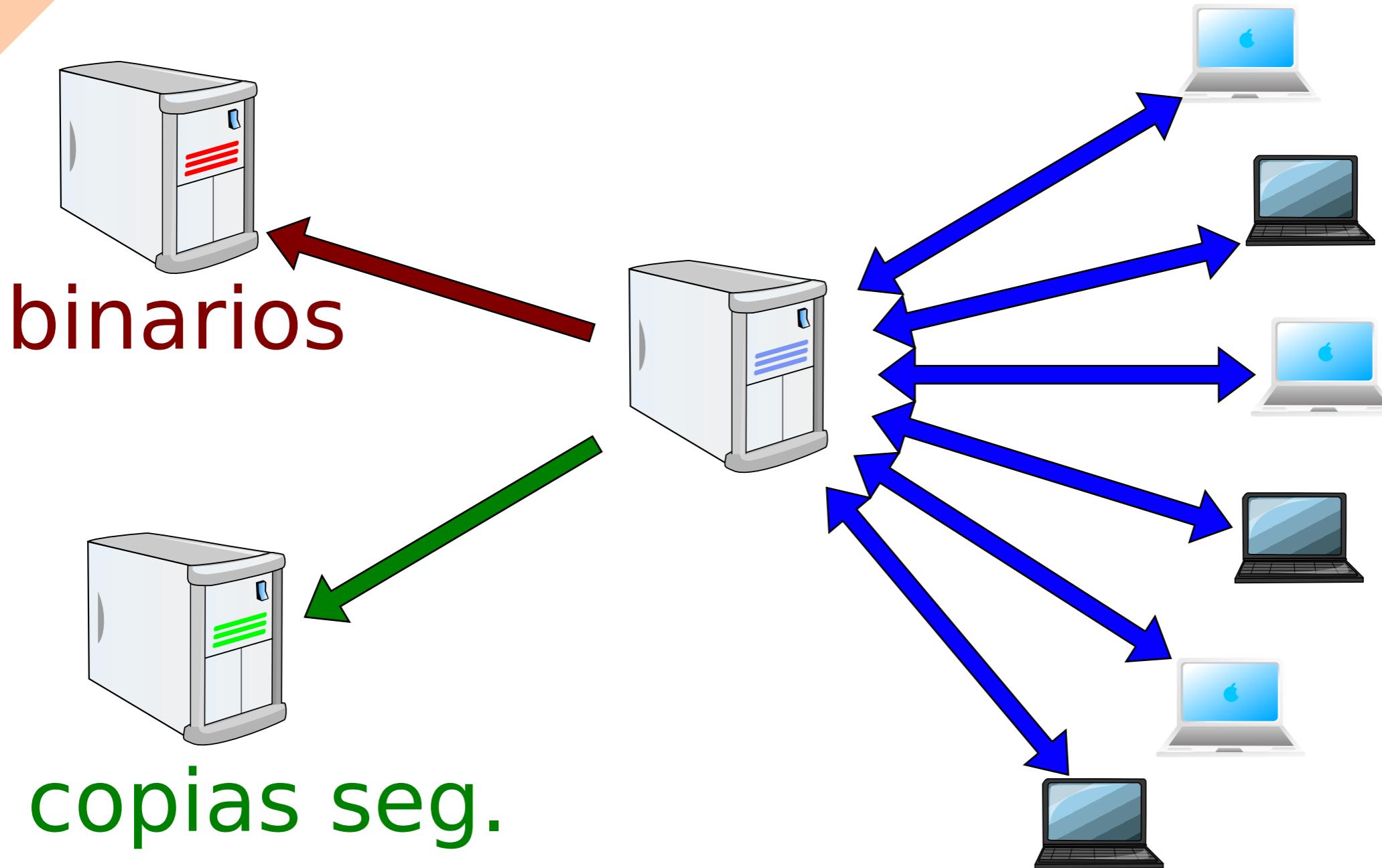
# Centralizado



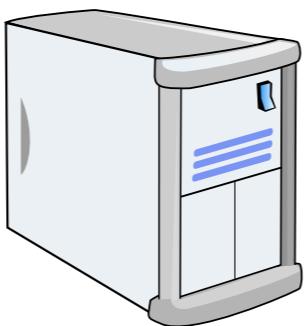
# Centralizado



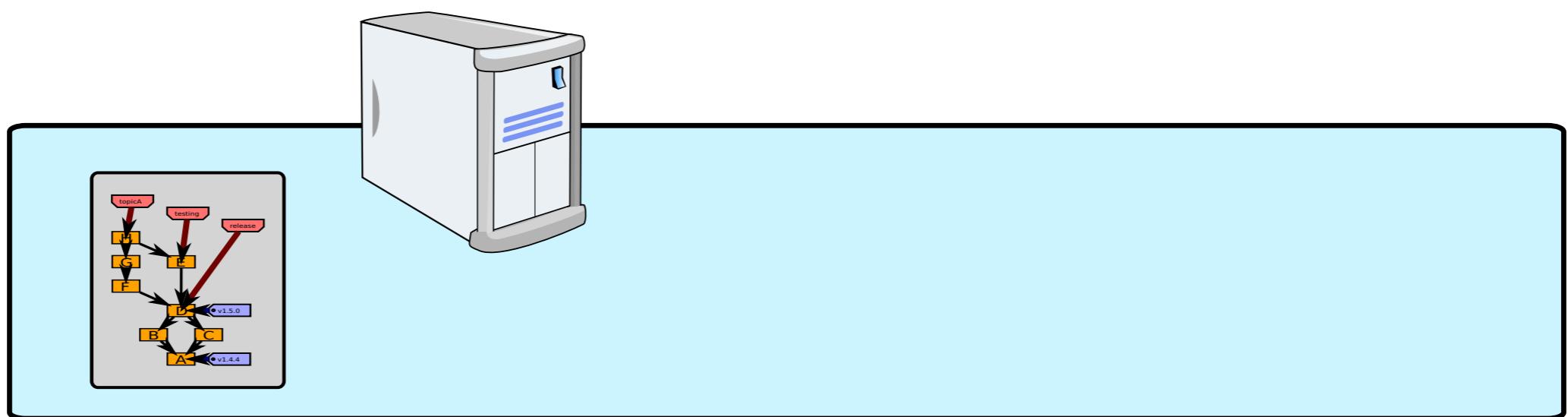
# Centralizado



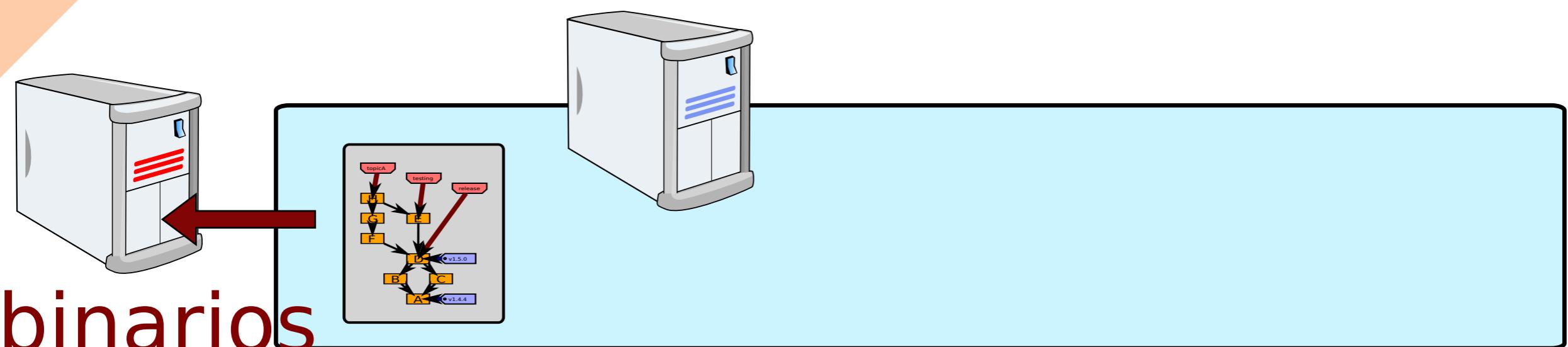
# Un integrador



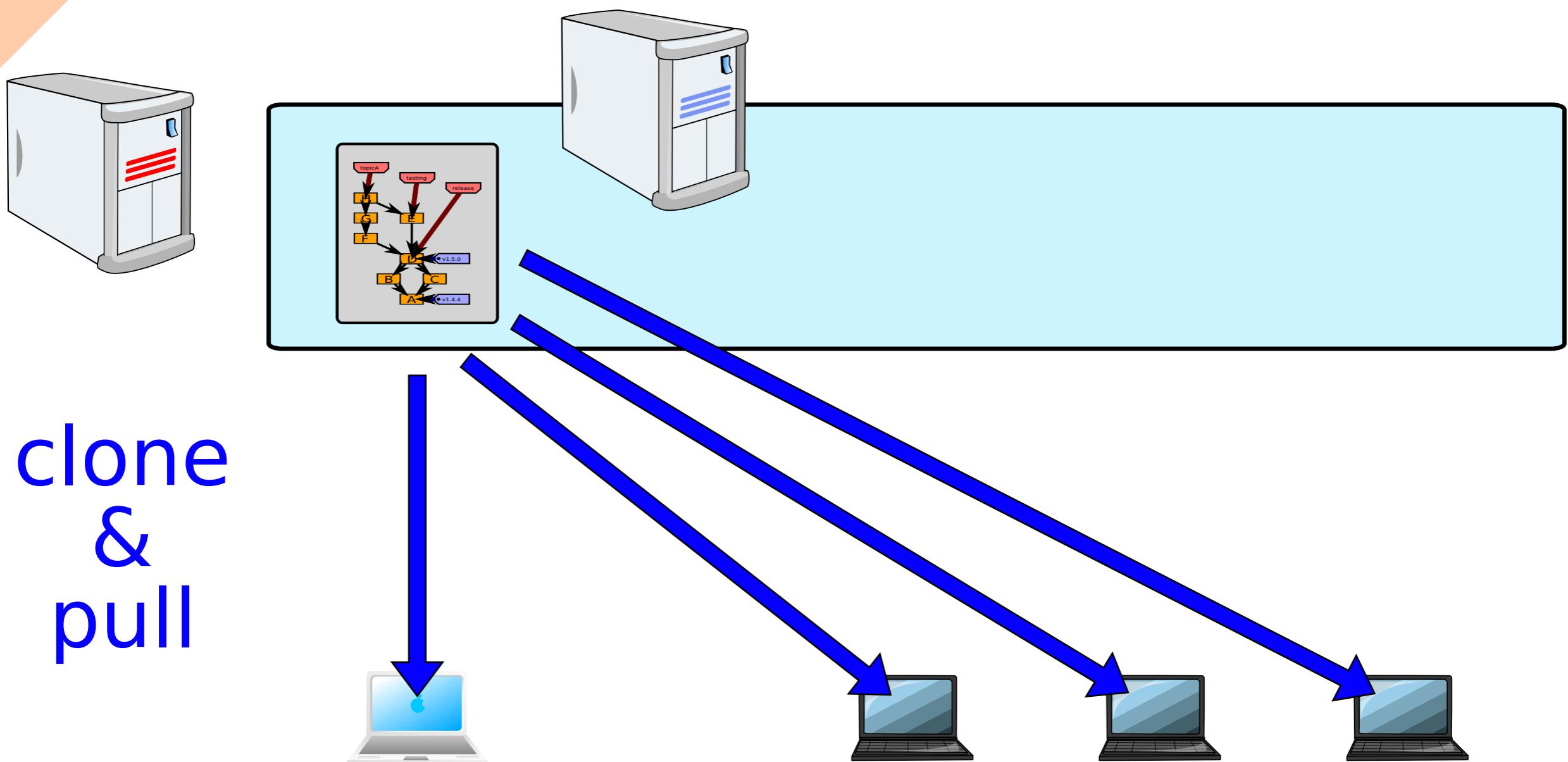
# Un integrador



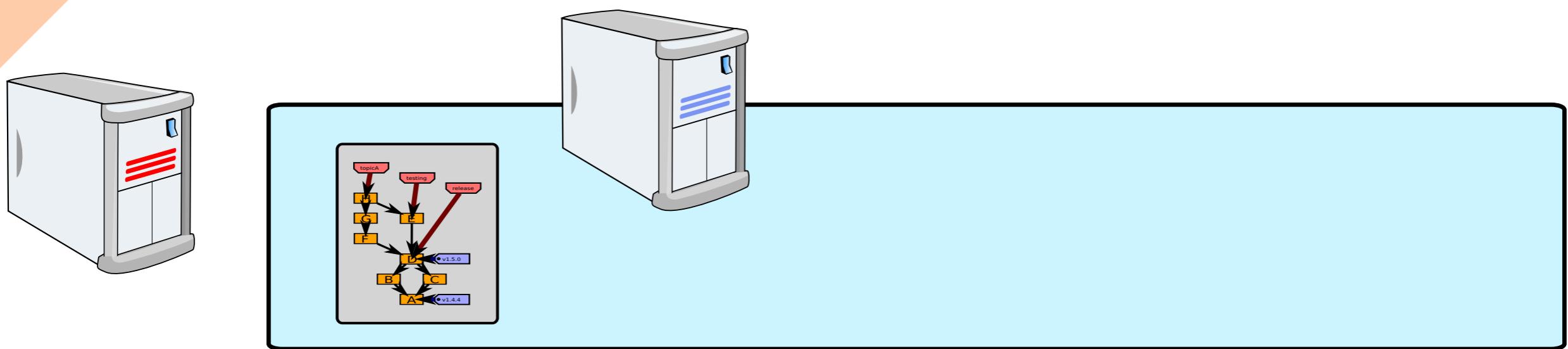
# Un integrador



# Un integrador



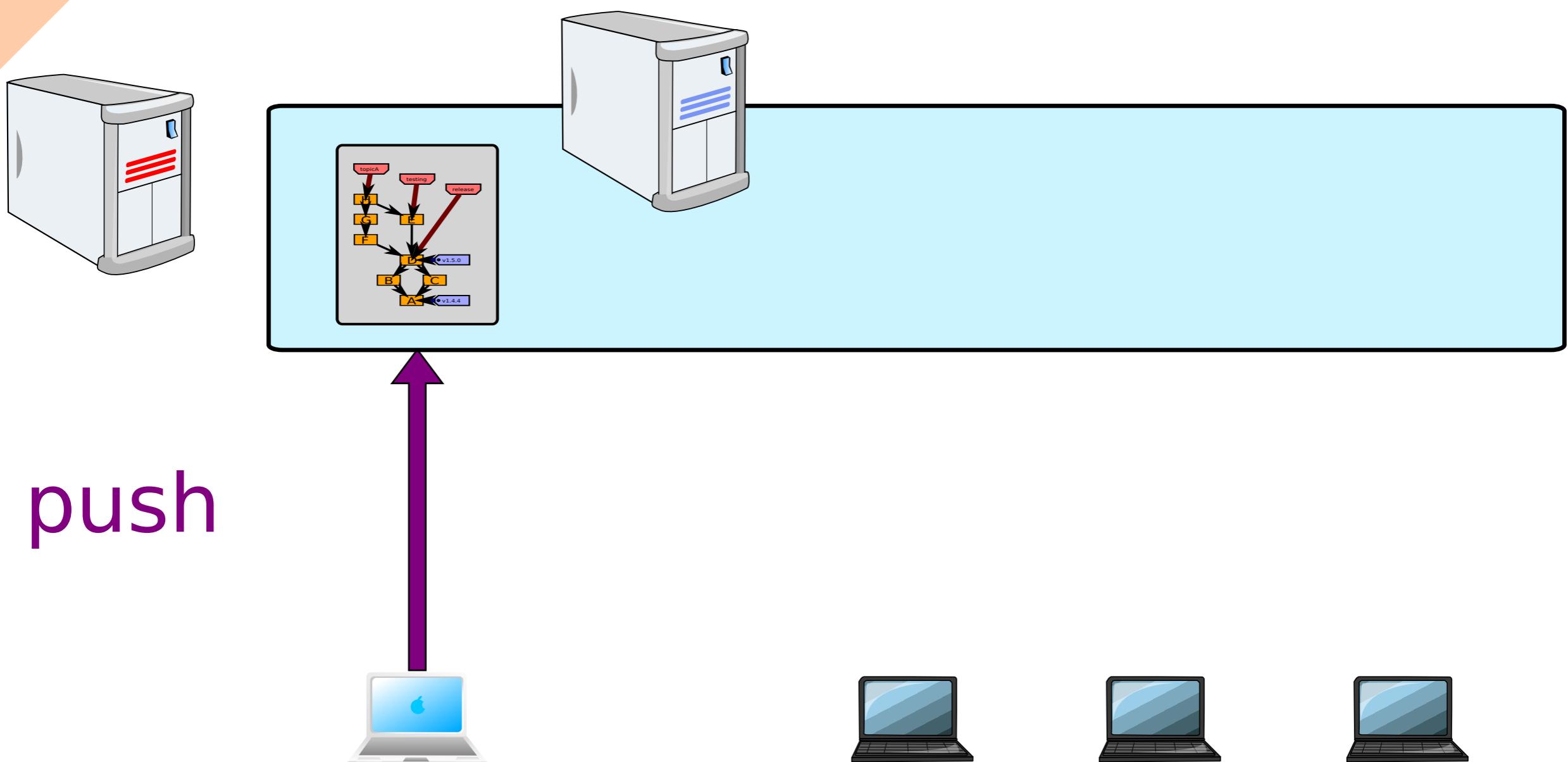
# Un integrador



integrador

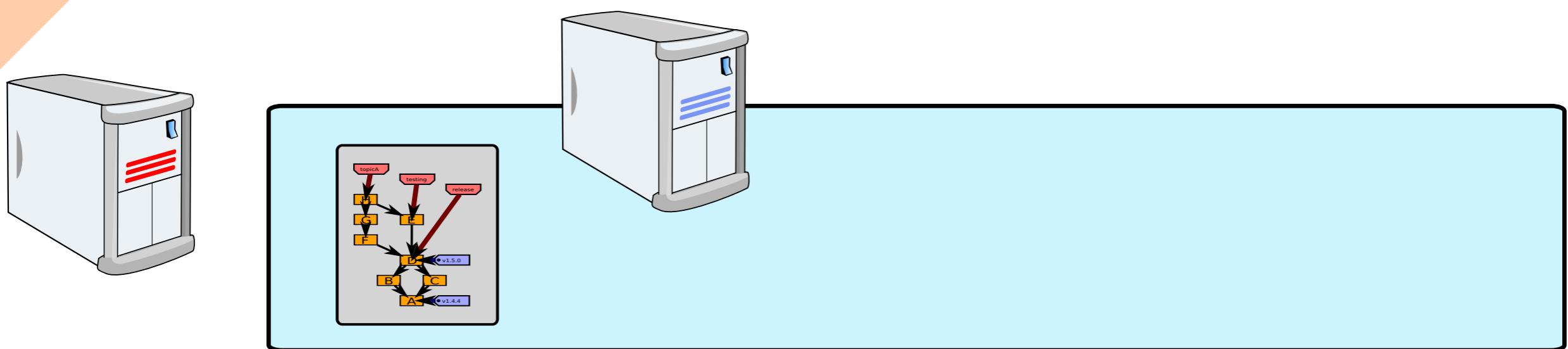


# Un integrador



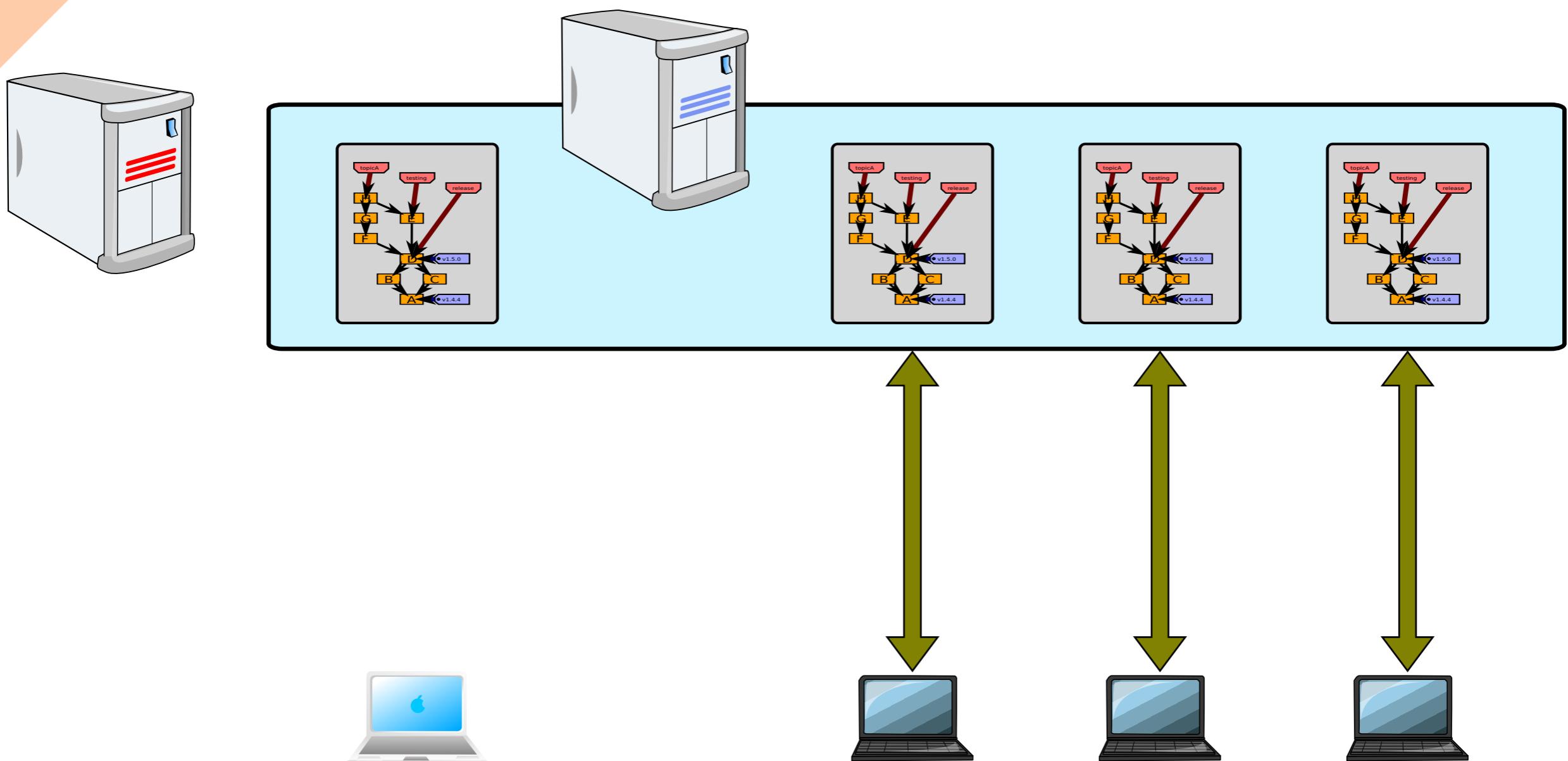
integrador

# Un integrador



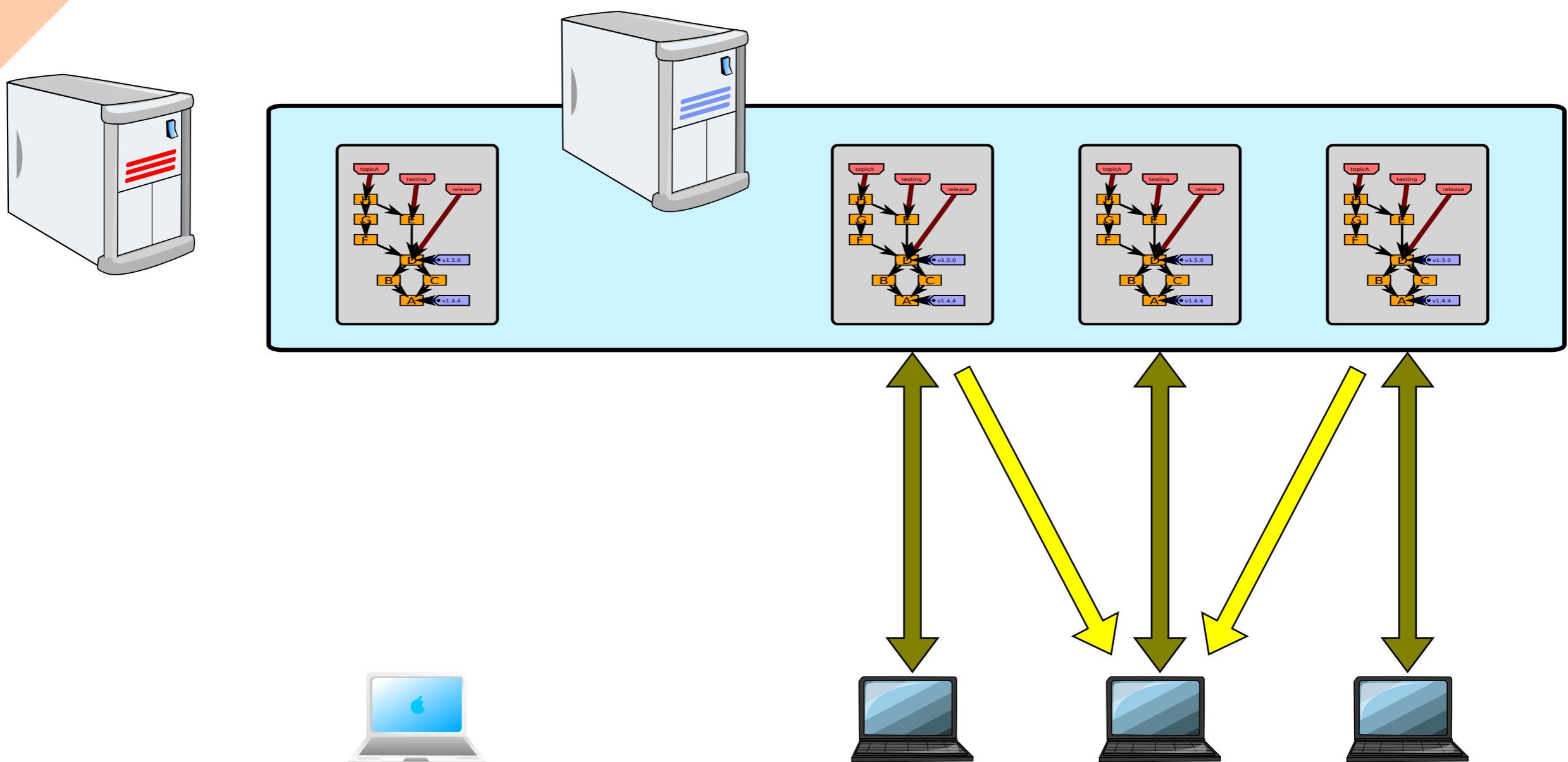
desarrolladores

# Un integrador



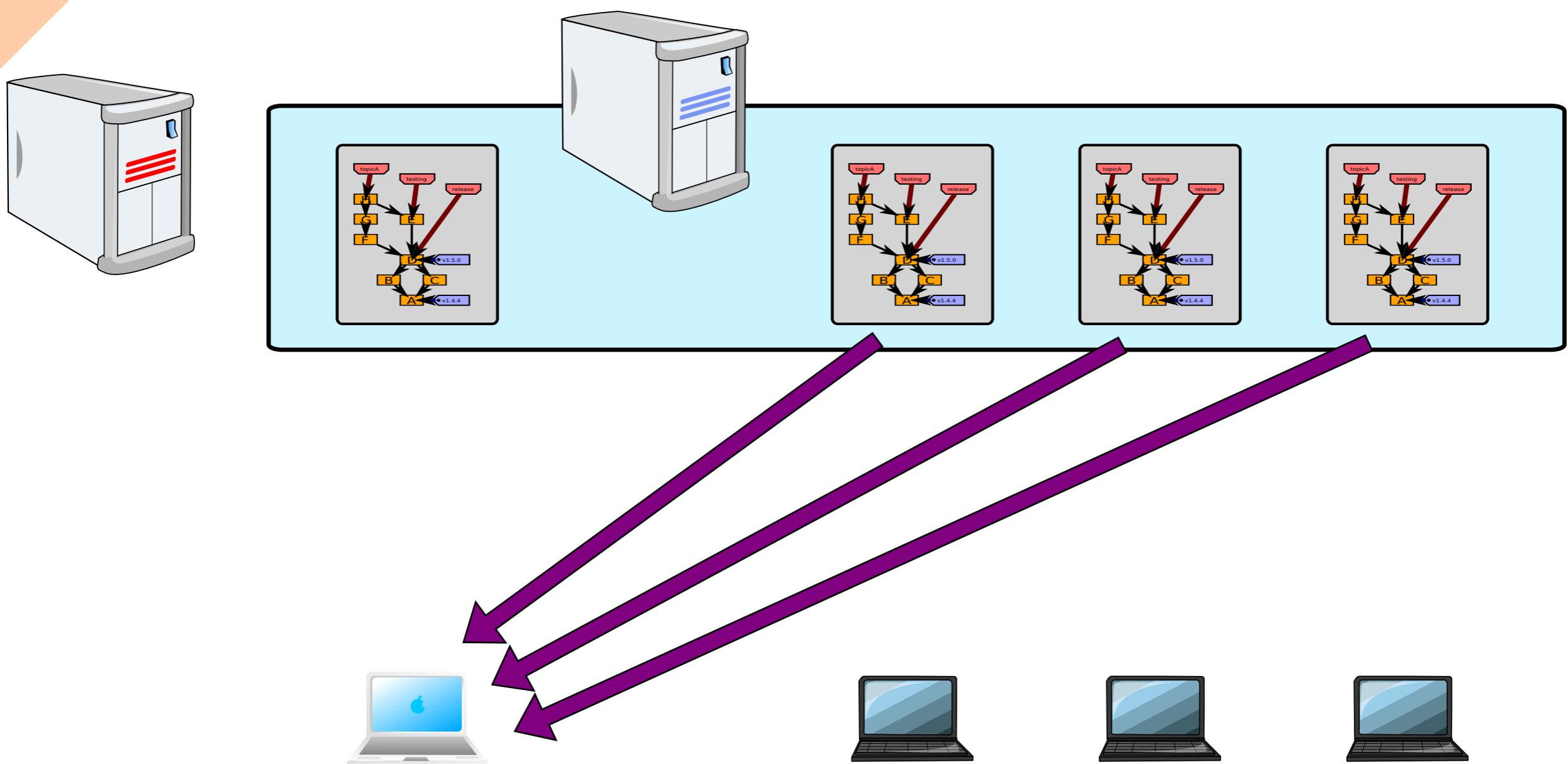
desarrolladores

# Un integrador



desarrolladores

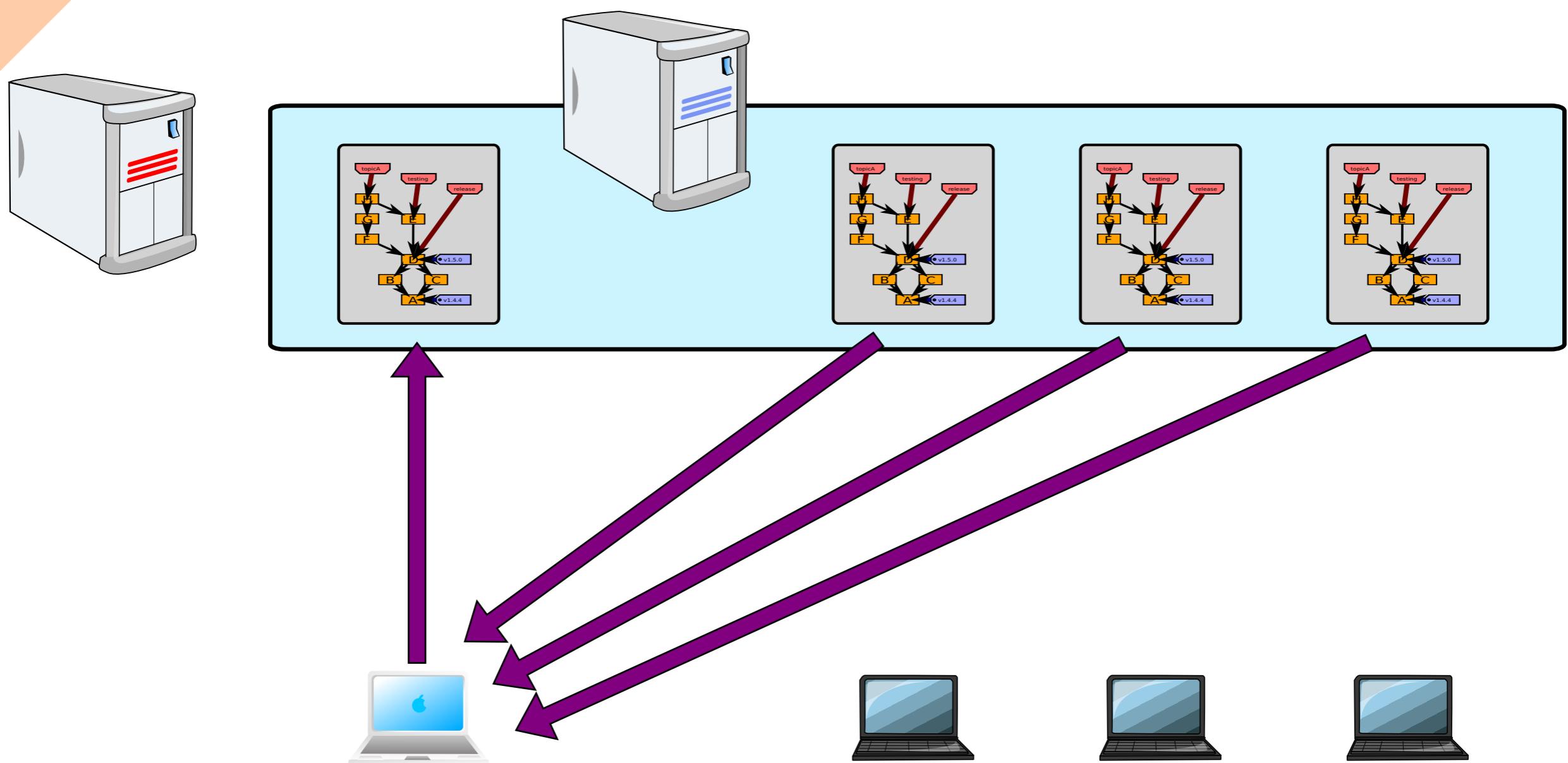
# Un integrador



integrador

desarrolladores

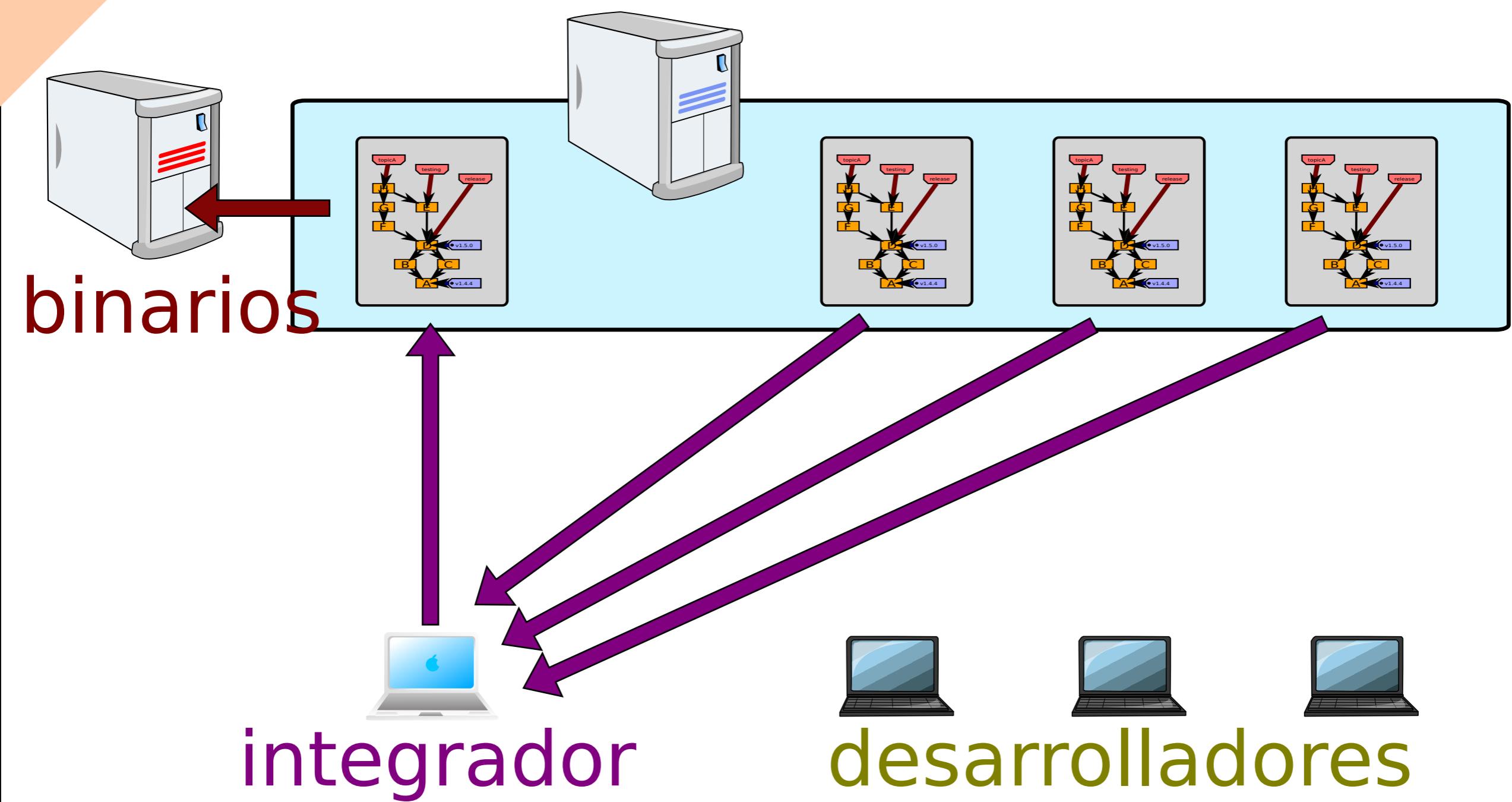
# Un integrador



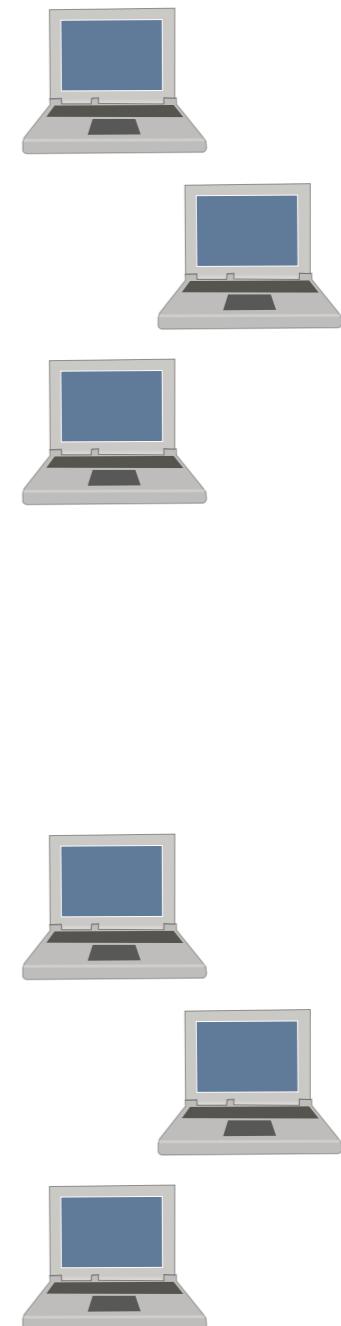
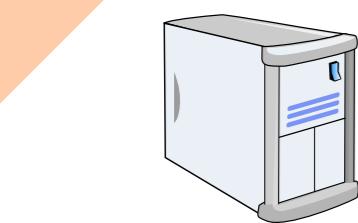
integrador

desarrolladores

# Un integrador



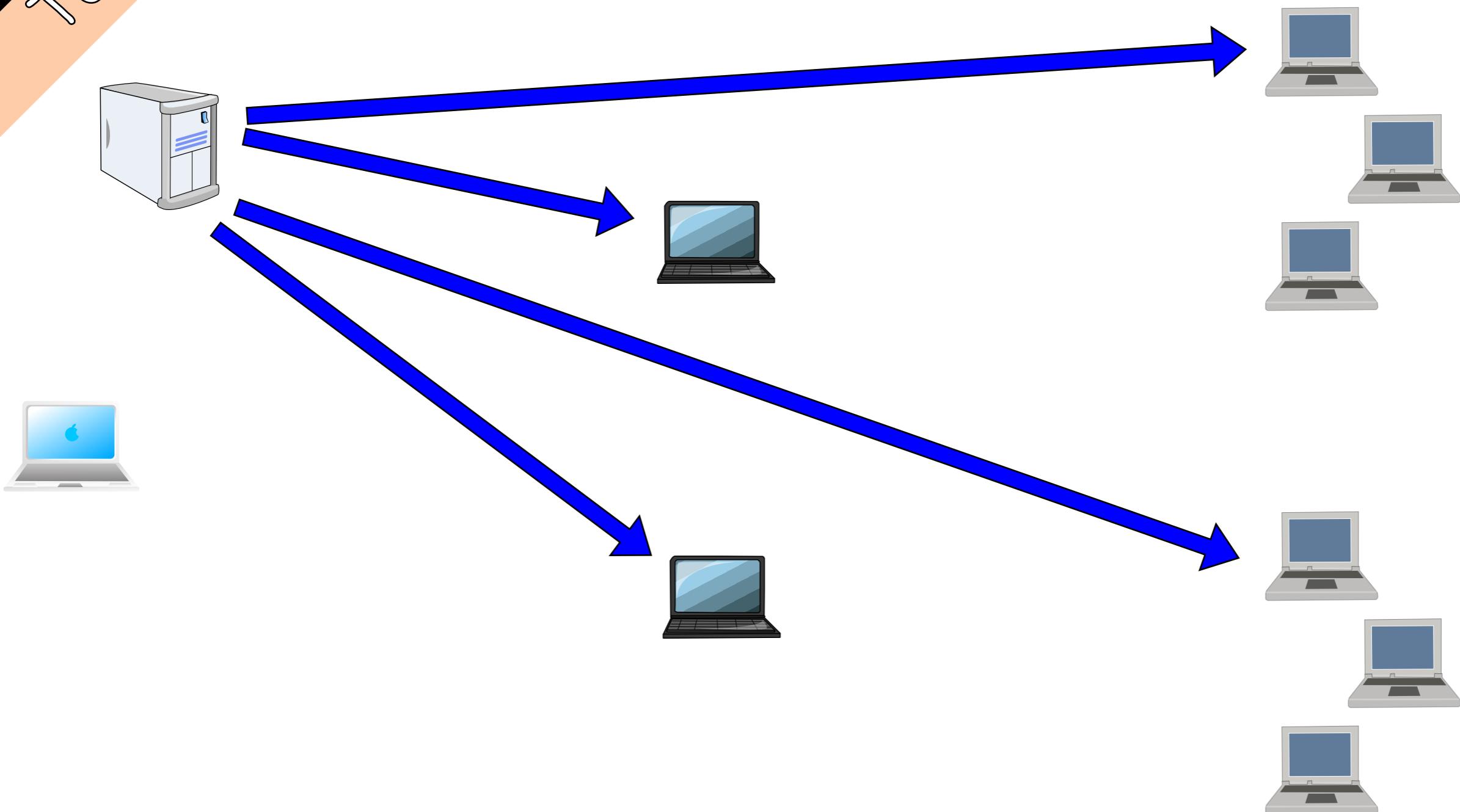
# Dictador y Tenientes



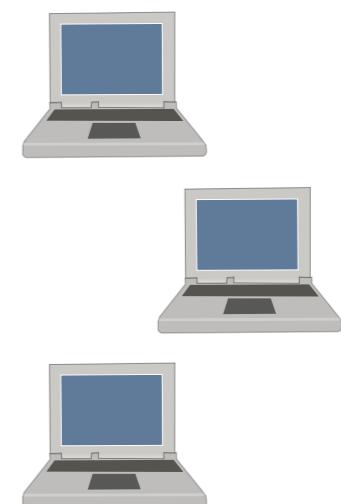
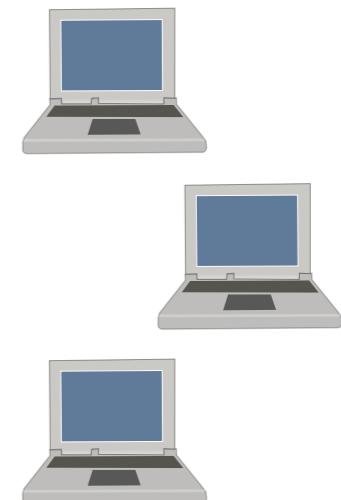
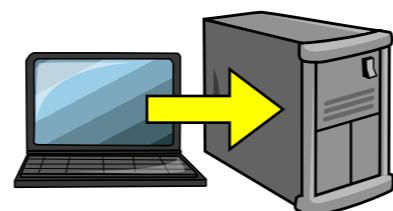
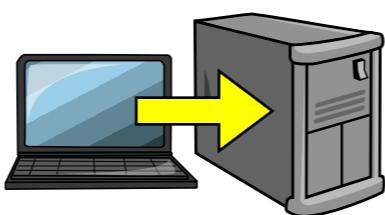
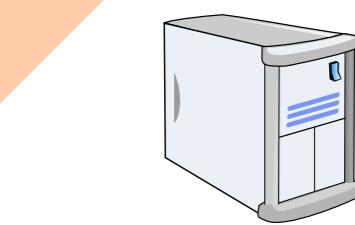
# Dictador y Tenientes



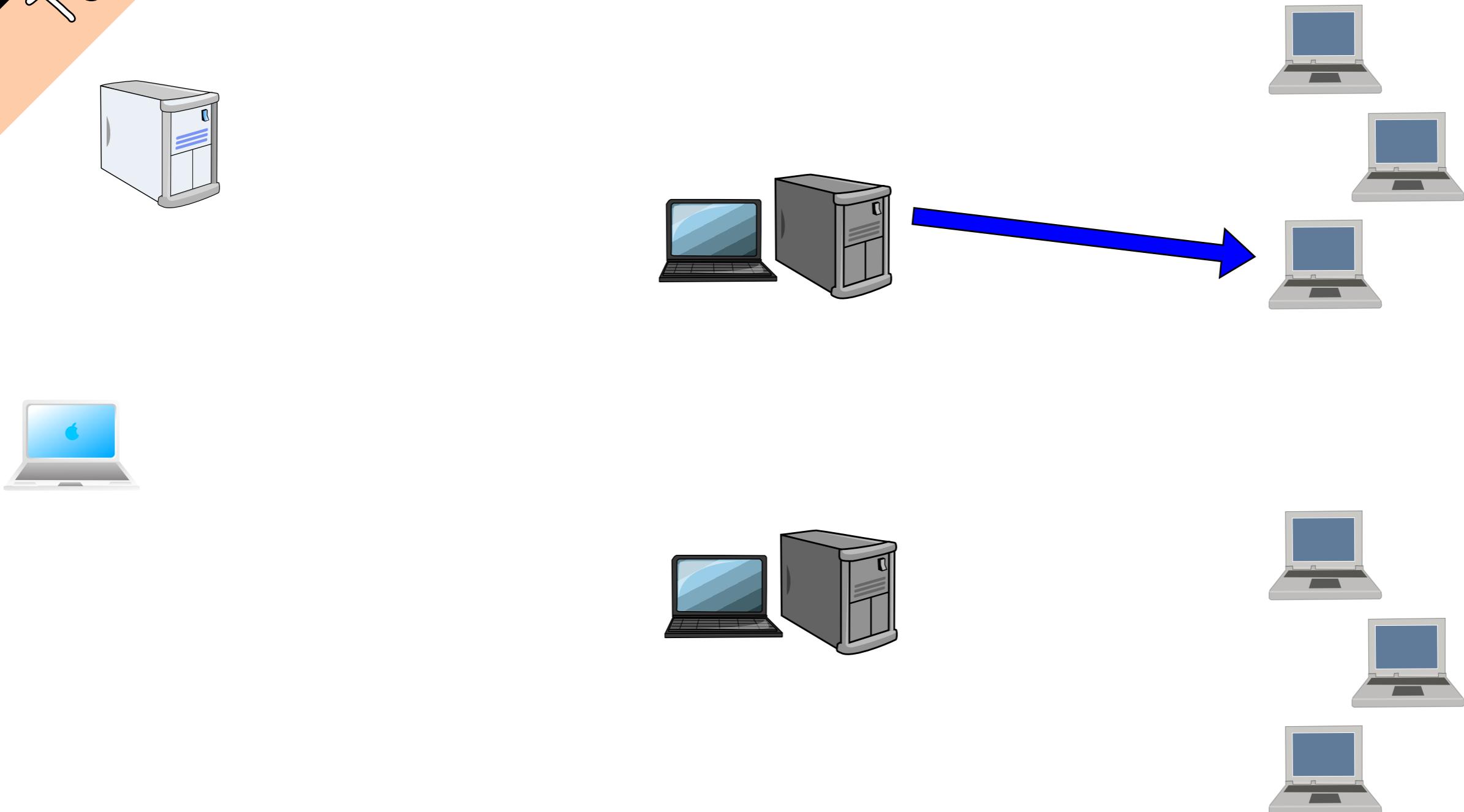
# Dictador y Tenientes



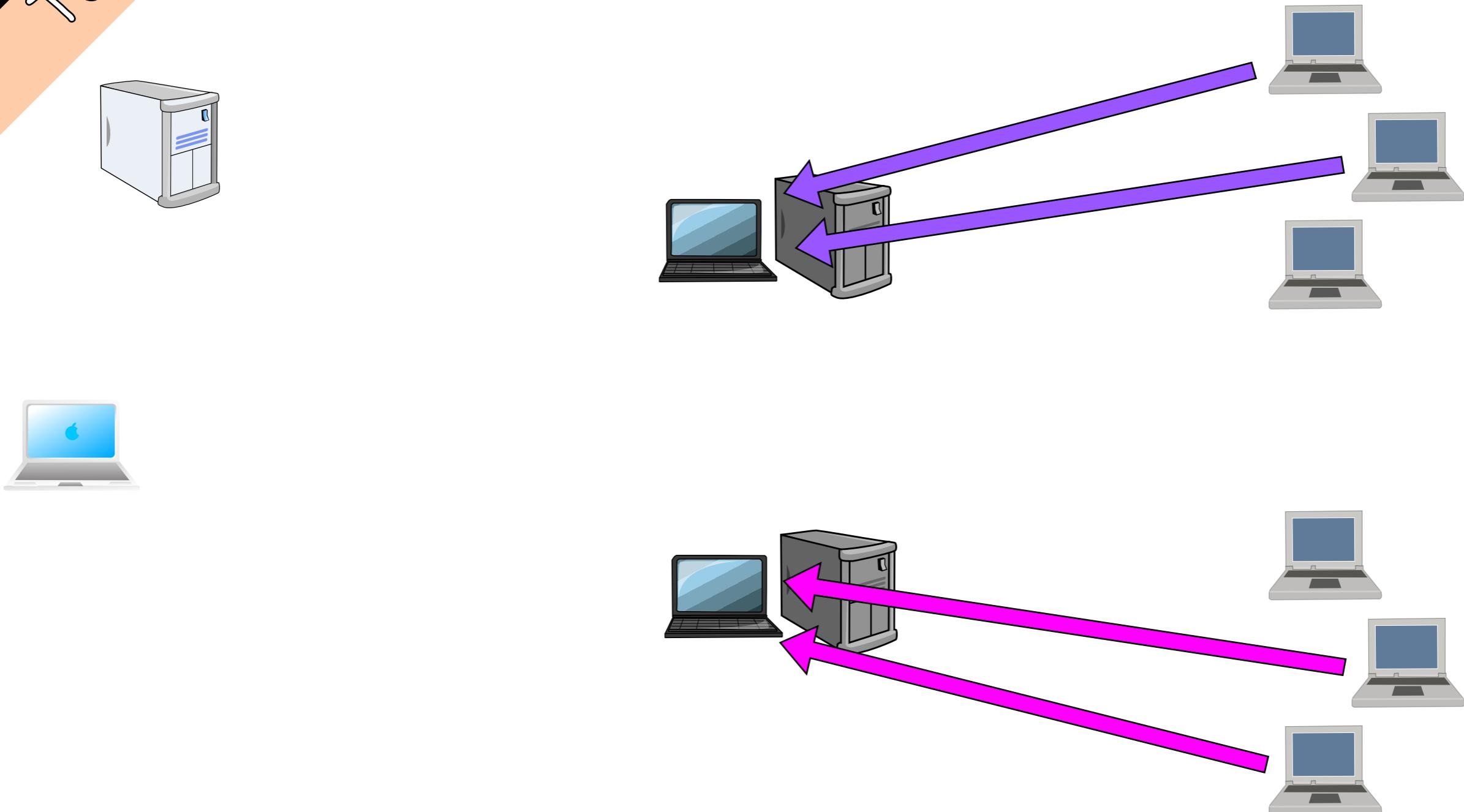
# Dictador y Tenientes



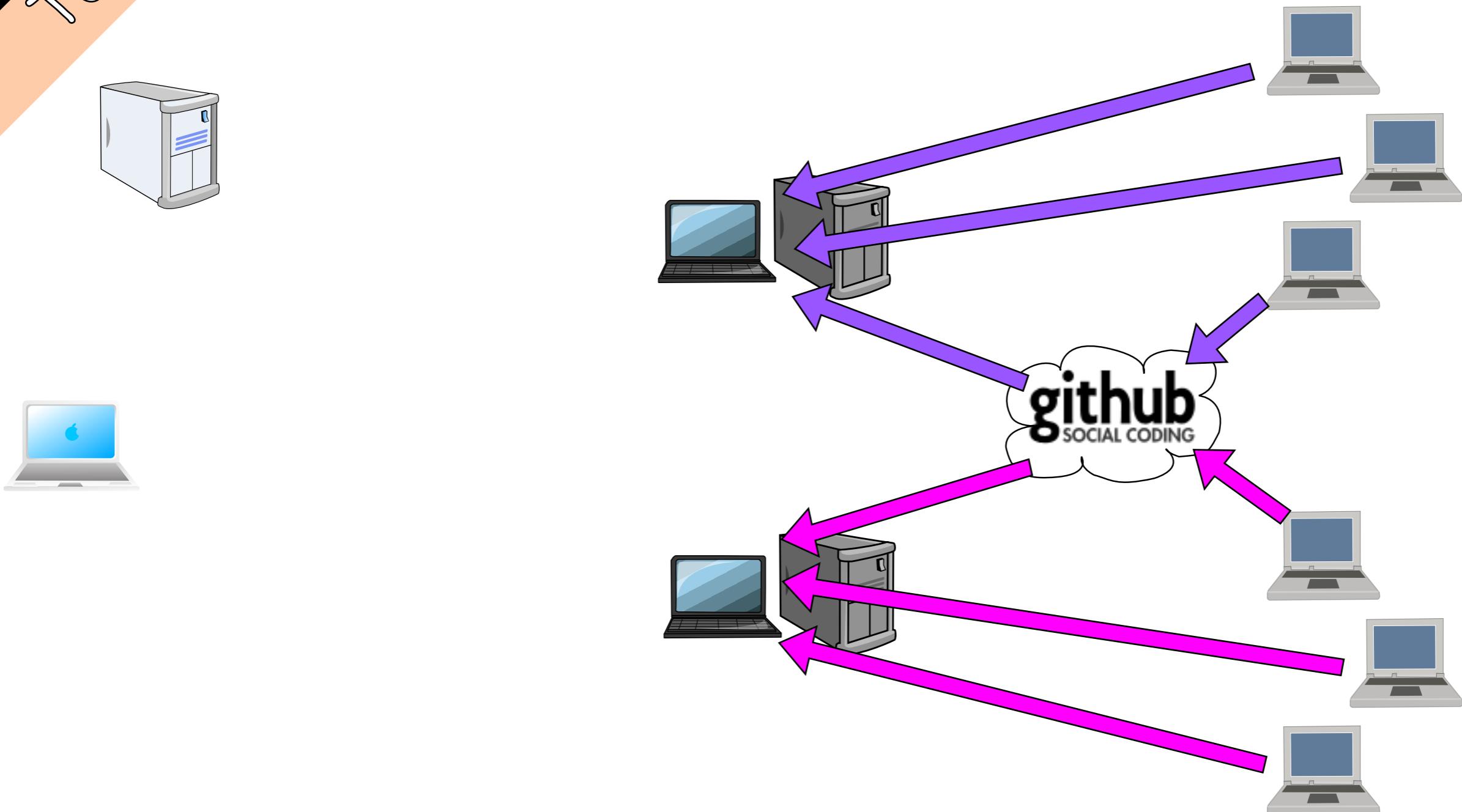
# Dictador y Tenientes



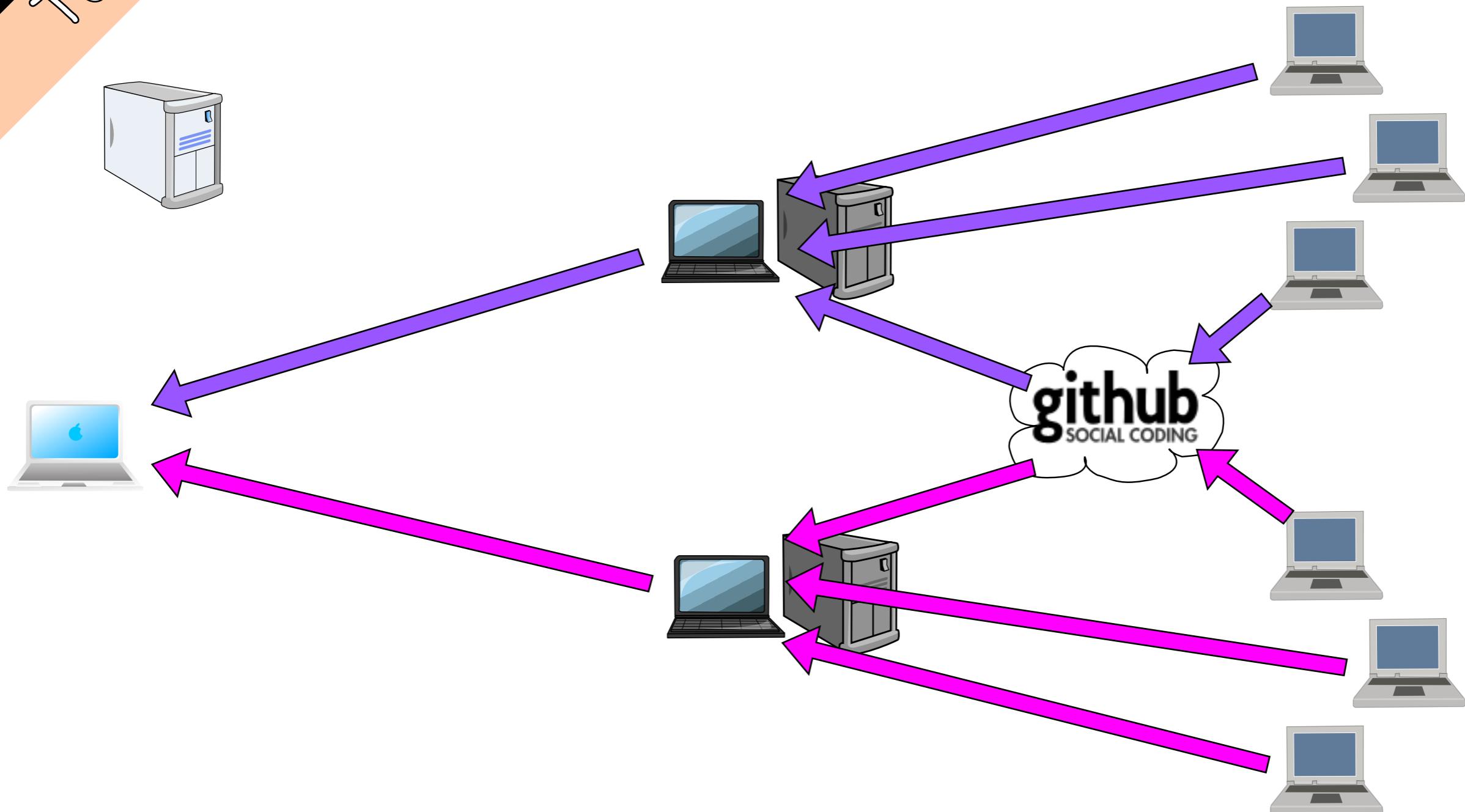
# Dictador y Tenientes



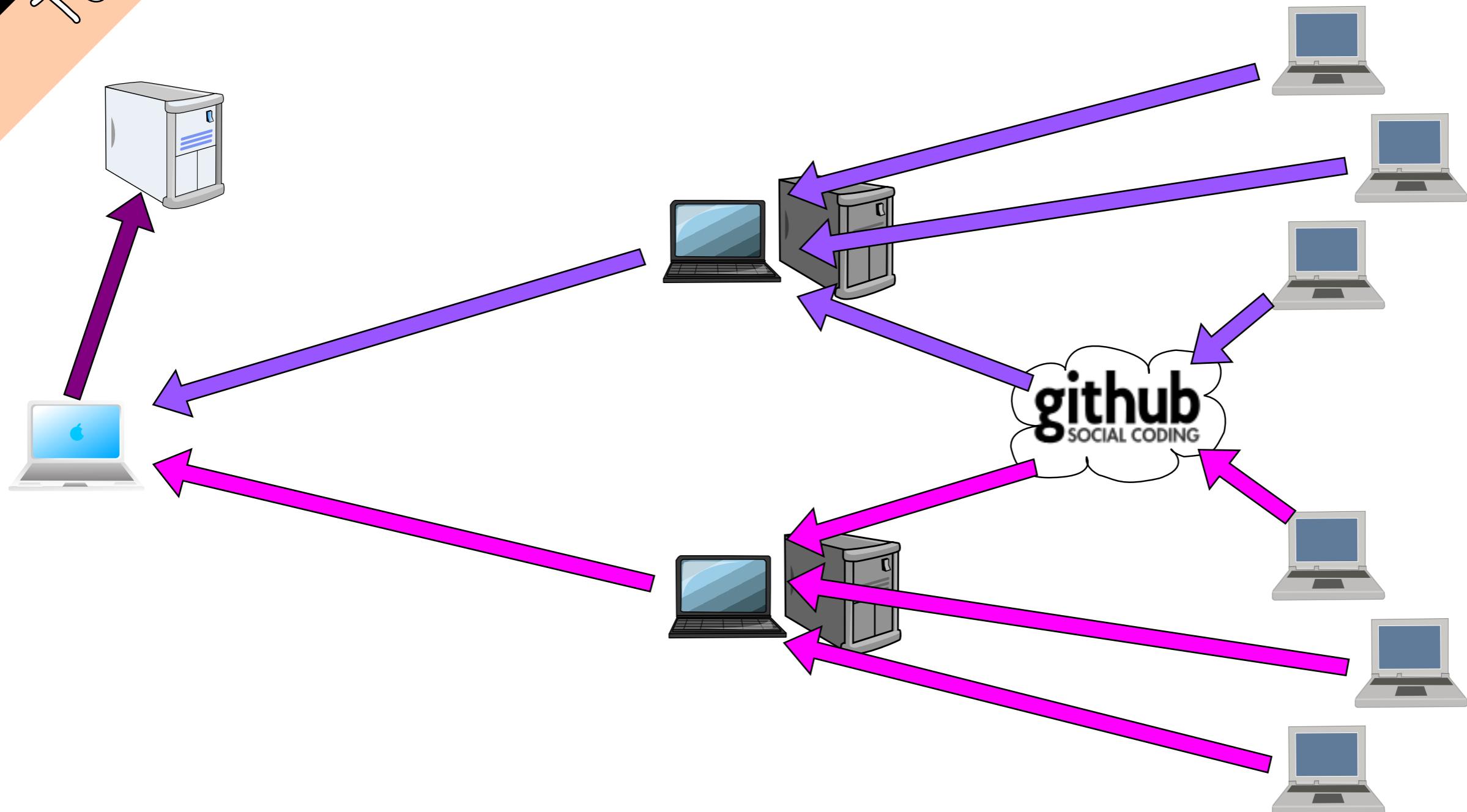
# Dictador y Tenientes



# Dictador y Tenientes



# Dictador y Tenientes



# Uso avanzado

# Interoperar con Subversion

- Queremos usar Git para trabajar en un proyecto que usa SVN
- ¿Cómo sacamos una copia?

# Interoperar con Subversion

- Queremos usar Git para trabajar en un proyecto que usa SVN
- ¿Cómo sacamos una copia?

`git svn clone <url repo>`

# Interoperar con Subversion

- Queremos usar Git para trabajar en un proyecto que usa SVN
- ¿Cómo sacamos una copia?

```
git svn clone -s <url repo>
```

# Interoperar con Subversion

- Queremos usar Git para trabajar en un proyecto que usa SVN
- ¿Cómo sacamos una copia?

`git svn clone -s <url repo>`

- ¿Cómo actualizamos?

# Interoperar con Subversion

- Queremos usar Git para trabajar en un proyecto que usa SVN
- ¿Cómo sacamos una copia?

`git svn clone -s <url repo>`

- ¿Cómo actualizamos?

`git svn rebase`

# Interoperar con Subversion

- Queremos usar Git para trabajar en un proyecto que usa SVN
- ¿Cómo sacamos una copia?

`git svn clone -s <url repo>`

- ¿Cómo actualizamos?

`git svn rebase`

- ¿Cómo enviamos?

# Interoperar con Subversion

- Queremos usar Git para trabajar en un proyecto que usa SVN
- ¿Cómo sacamos una copia?

`git svn clone -s <url repo>`

- ¿Cómo actualizamos?

`git svn rebase`

- ¿Cómo enviamos?

`git svn dcommit`

# Interoperar con Subversion

- Queremos usar Git para trabajar en un proyecto que usa SVN
- ¿Cómo sacamos una copia?

`git svn clone -s <url repo>`

- ¿Cómo actualizamos?

`git svn rebase`

- ¿Cómo enviamos?

`git svn dcommit --rmdir`

# Rebase interactivo

- Normalmente "rebase" replantea todas las revisiones sobre otra, tal y como estaban.
- Hay una variante interactiva que nos deja decidir cómo se replantean las revisiones de la rama.
- Potente, pero ¡no lo uséis en ramas publicadas!

# Rebase interactivo

Limpiemos la rama **ejemplo-rebase-i**.

# Rebase interactivo

Limpiemos la rama **ejemplo-rebase-i**.

```
git checkout ejemplo-rebase-i
```

# Rebase interactivo

Limpiemos la rama **ejemplo-rebase-i**.

```
git checkout ejemplo-rebase-i
```

```
git log
```

```
65d1471 Revisión fuera de orden
18b31b4 commit a medias (2/2)
ab312ba commit a medias (1/2)
a2febb6 mensaje que no sirve para nada
d7fcace commit inútil
584c66b primera revisión
```

# Rebase interactivo

Limpiemos la rama **ejemplo-rebase-i**.

```
git checkout ejemplo-rebase-i
```

```
git log
```

```
65d1471 Revisión fuera de orden
18b31b4 commit a medias (2/2)
ab312ba commit a medias (1/2)
a2febb6 mensaje que no sirve para nada
d7fcace commit inútil
584c66b primera revisión
```

Está desorganizada. Vamos a limpiarla.

# Rebase interactivo

Limpiemos la rama **ejemplo-rebase-i**.  
**git rebase -i 584c66b**

# Rebase interactivo

Limpiemos la rama **ejemplo-rebase-i**.  
**git rebase -i 584c66b**

```
pick d7fcace commit inútil
pick a2febb6 mensaje que no sirve para nada
pick ab312ba commit a medias (1/2)
pick 18b31b4 commit a medias (2/2)
pick 65d1471 Revisión fuera de orden

# Rebase 584c66b..65d1471 onto 584c66b
#
# Commands:
#   p, pick = use commit
#   r, reword = use commit, but edit the commit message
#   e, edit = use commit, but stop for amending
#   s, squash = use commit, but meld into previous commit
#   f, fixup = like "squash", but discard this commit's log message
#
# If you remove a line here THAT COMMIT WILL BE LOST.
# However, if you remove everything, the rebase will be aborted.
#
```

# Rebase interactivo

Limpiemos la rama **ejemplo-rebase-i**.  
Lo dejamos así, guardamos y salimos:

```
pick 65d1471 Revisión fuera de orden
e a2febb6 mensaje que no sirve para nada
pick ab312ba commit a medias (1/2)
s 18b31b4 commit a medias (2/2)

# Rebase 584c66b..65d1471 onto 584c66b
#
# Commands:
#   p, pick = use commit
#   r, reword = use commit, but edit the commit message
#   e, edit = use commit, but stop for amending
#   s, squash = use commit, but meld into previous commit
#   f, fixup = like "squash", but discard this commit's log message
#
# If you remove a line here THAT COMMIT WILL BE LOST.
# However, if you remove everything, the rebase will be aborted.
#
```



# Rebase interactivo

Limpiemos la rama **ejemplo-rebase-i**.

Resolvemos un conflicto con:

`git mergetool`

`git commit`

Continuamos el rebase con:

`git rebase -c`

Unimos los dos mensajes, seguimos y terminamos. Todo limpio :-).

# ¿Qué preferís?

- Git en Windows
- interfaz potente CLI: "add -i"
- colaboración sin red: "bundle"
- colaboración por correo: "send-email"
- registros de referencias: "reflogs"
- biseción de defectos: "bisect"
- importación de commits:"cherry-pick"
- demonio git://: "daemon"
- servidor Git por SSH: Gitosis
- "stash", "textconv", etc.

# gracias



contactar con el autor original

**<http://www.jukie.net/~bart/>**

obtener las transparencias del taller original

**<http://2tu.us/wm4>**

contactar con el autor de la versión en castellano

**<http://xmleye.wordpress.com>**

obtener las transparencias en castellano

**<http://gitorious.org/spanish-git-reflections>**

sitio web oficial

**<http://git-scm.com>**