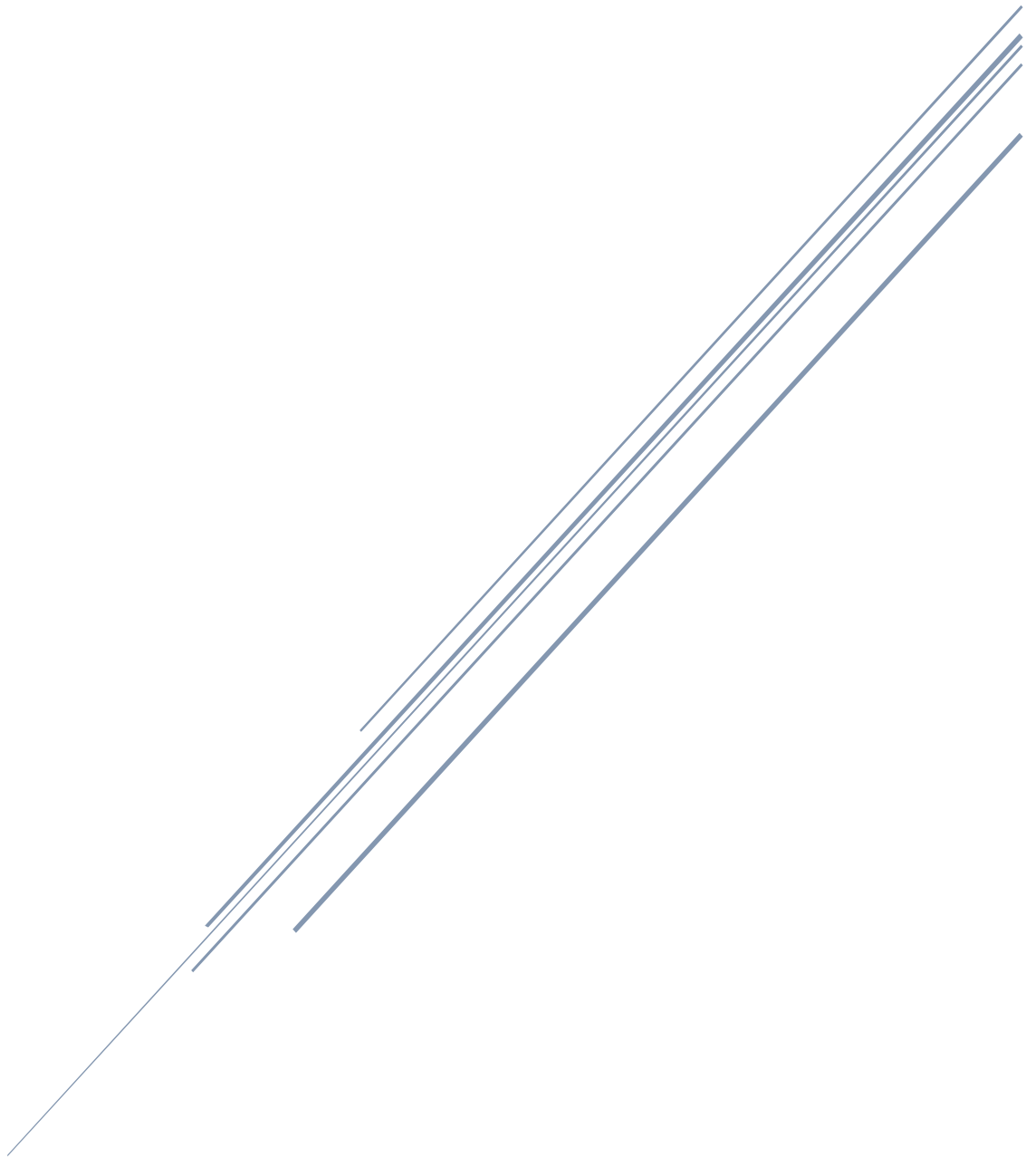


ENTORNOS DE DESARROLLO

UD1 - Práctica 4. GitGame



Decroly Santander
DAW 1 Hecho por: David Valle

Índice

Introducción 2

Desarrollo 3

NIVELES APARTADO INTRO 3

NIVELES APARTADO ACELERANDO 6

NIVELES APARTADO MOVE 9

NIVELES APARTADO MIXED 12

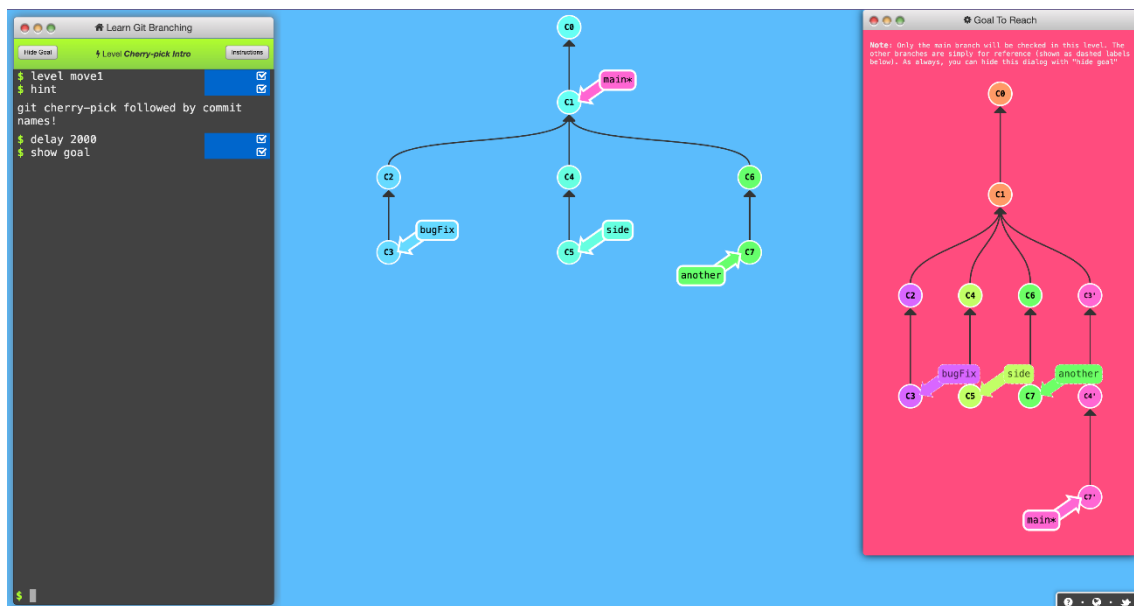
NIVELES APARTADO ADVANCED 17

NIVELES APARTADO REMOTE 21

NIVELES APARTADO REMOTE ADVANCES 29

Introducción

En este trabajo jugaremos a un juego de Git para familiarizarnos con los comandos y poder aprenderlos de una forma “divertida” pasando diferentes niveles con el objetivo de entender como funciona y como podriamos utilizarlo en proyectos reales.



Desarrollo

NIVELES APARTADO INTRO

LVL 1

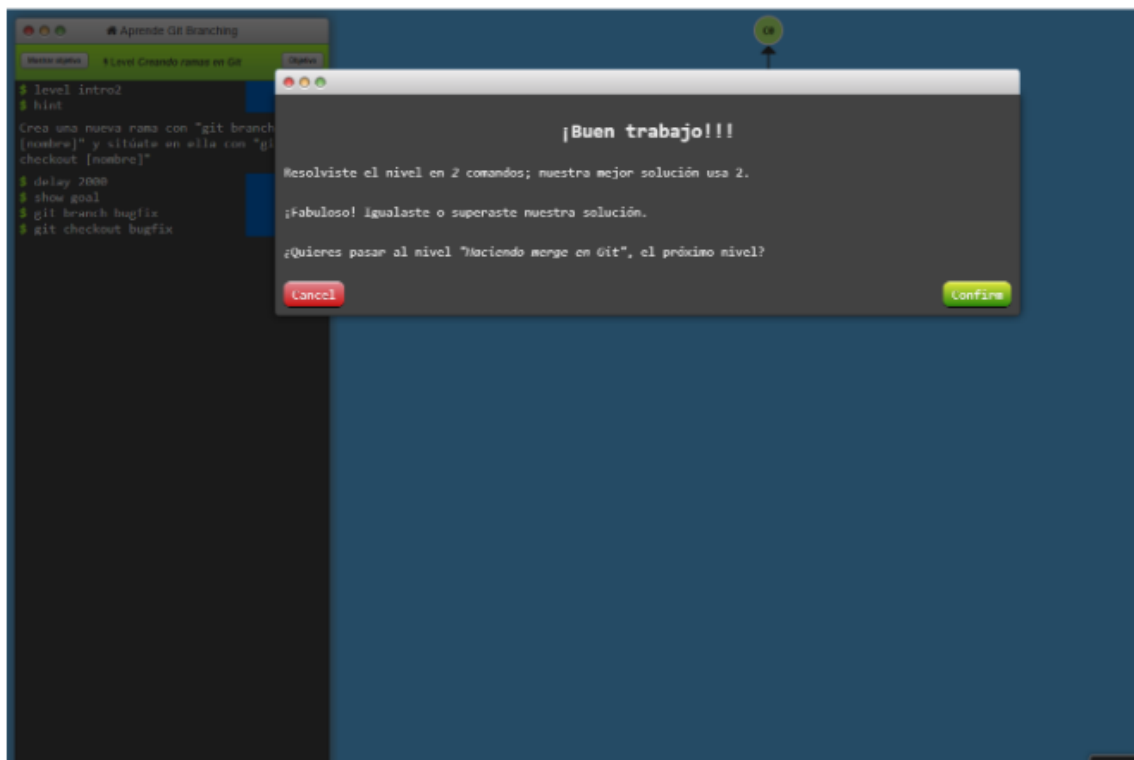
Introducción a los commits:

Cada vez que realicemos algo, con git commit realizaremos una instantánea



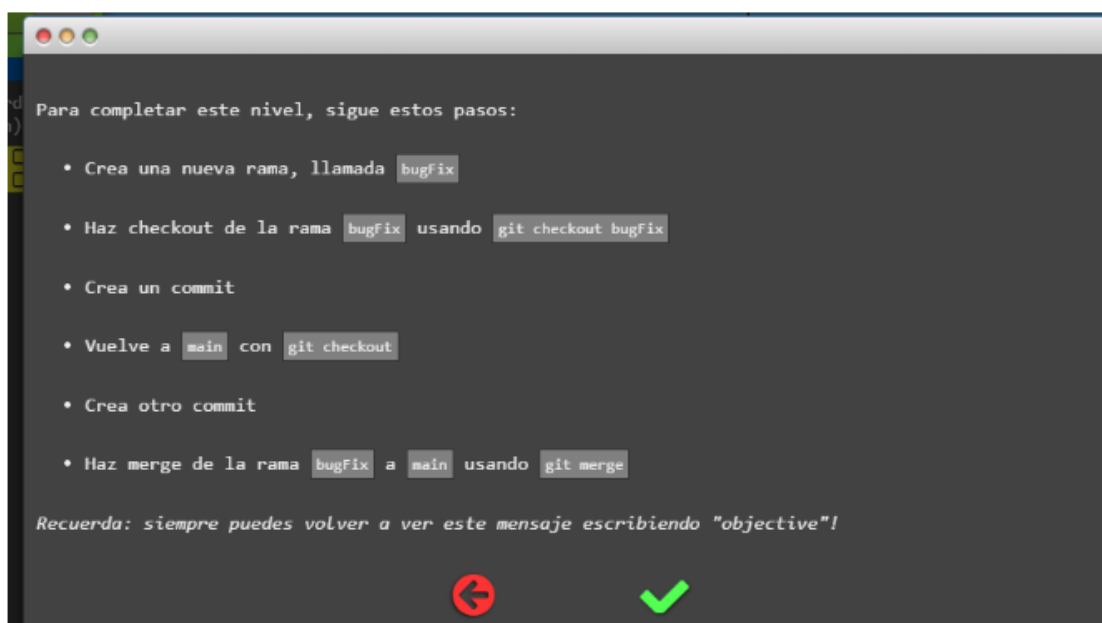
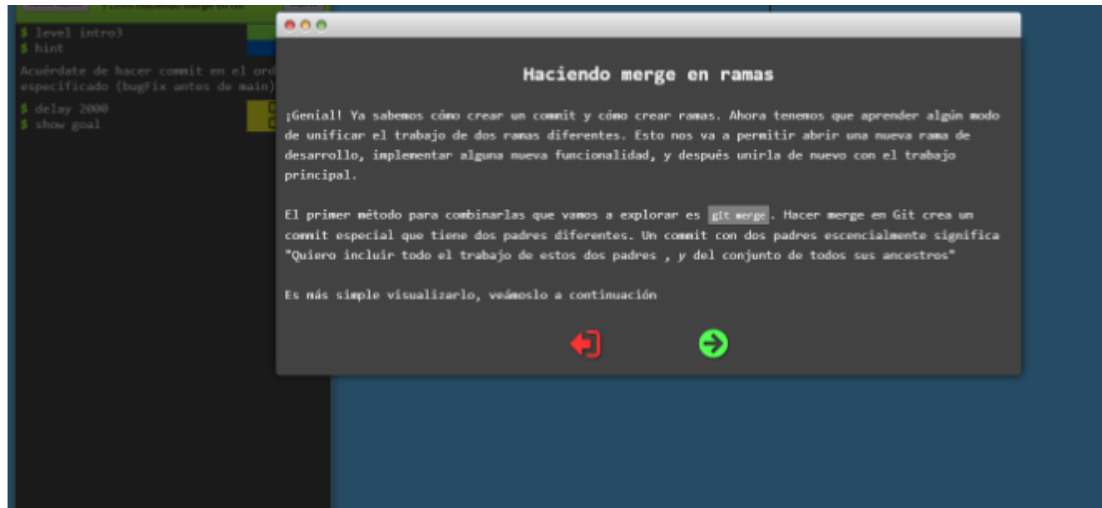
LVL 2

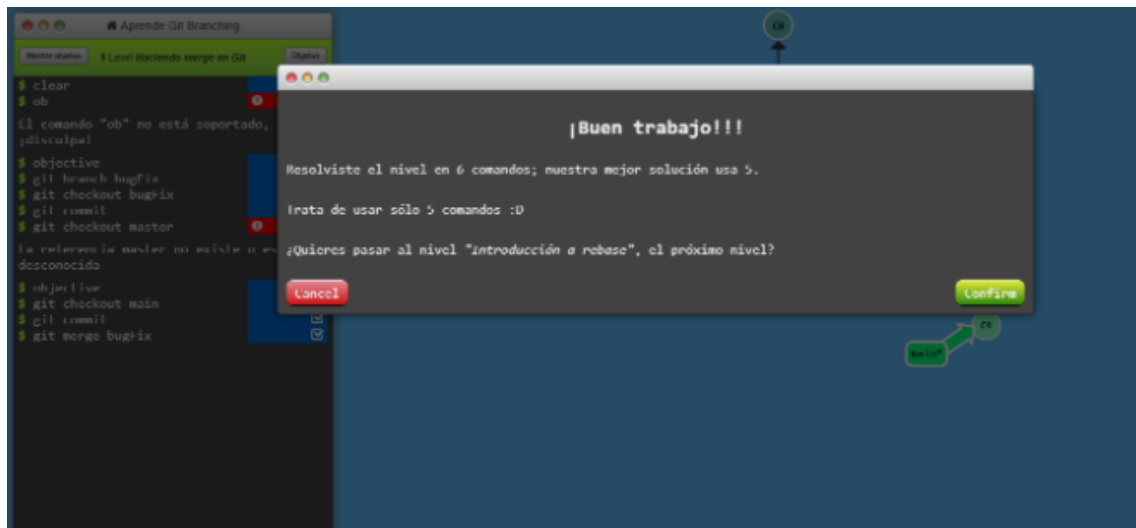
En este ejercicio nos piden crear una rama llamada BugFix y cambiar a esa rama, para ello utilizaremos los siguientes comandos:



LVL 3

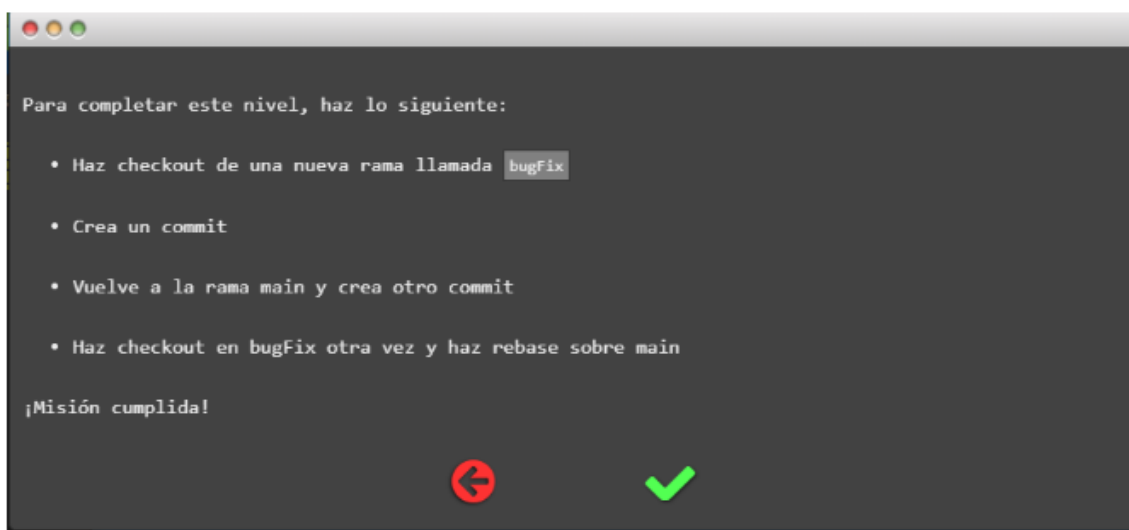
Utilizaremos git merge para crear commits que unan el trabajo de 2 ramas, combinando los cambios en uno solo.

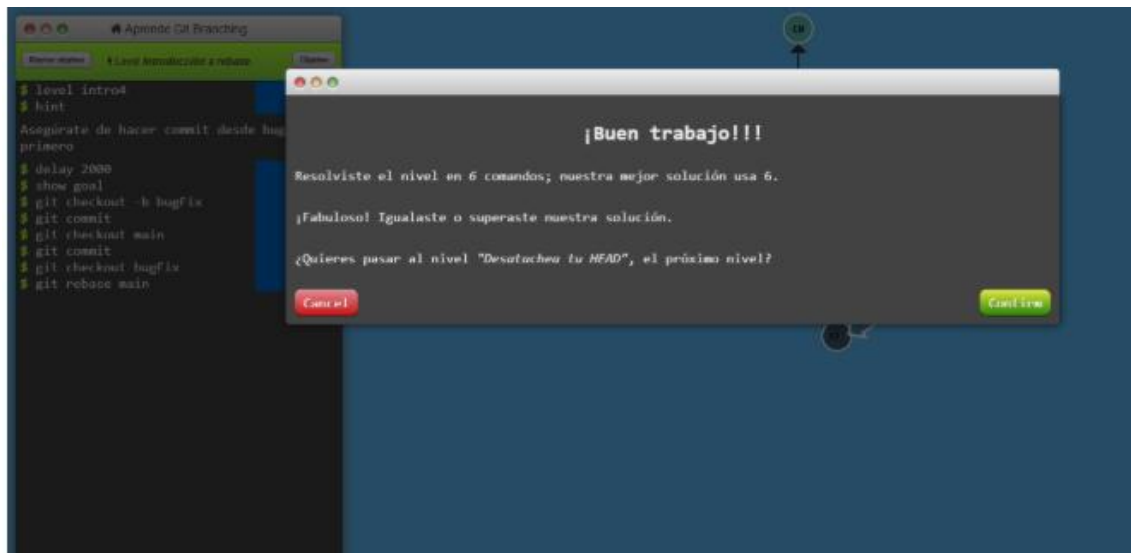




LVL 4

En este nivel utilizaremos rebase que es otra forma de poder unir, seleccionaremos varios commits, haremos una copia y lo colocaremos en el main para dejarlo más ordenado.

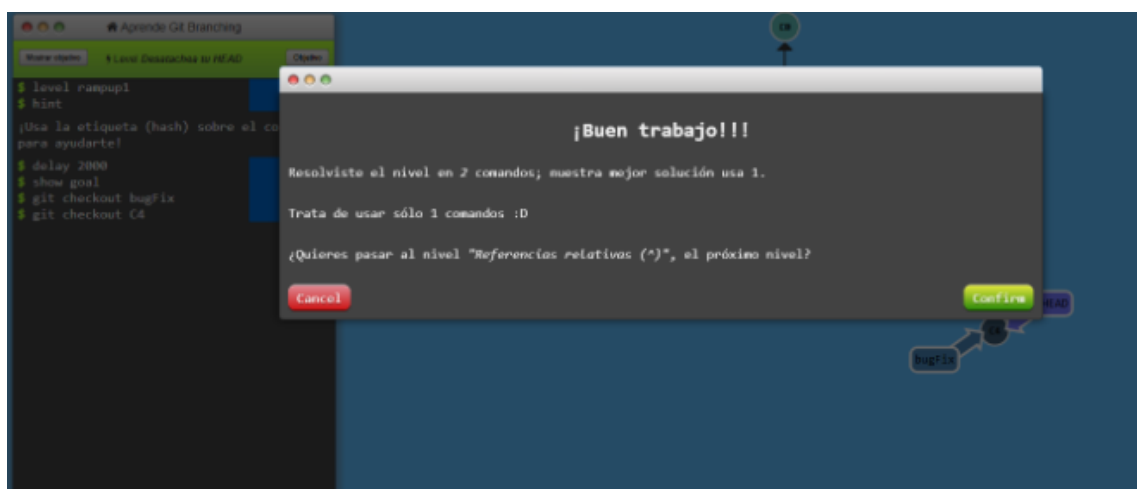
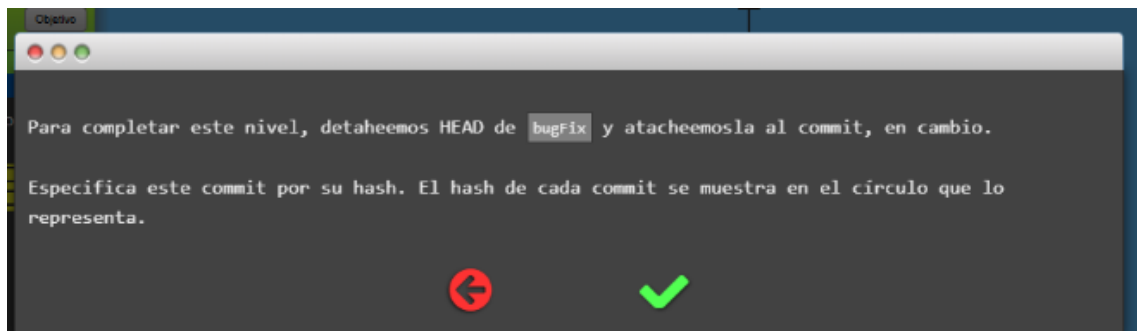


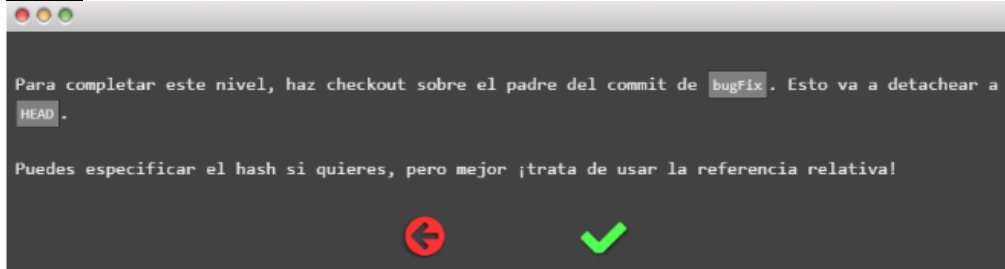


NIVELES APARTADO ACELERANDO

LVL 1

Moveremos el HEAD a otra parte para poder hacer un commit utilizando git checkout para cambiar la rama. Como el head esta en BugFix y tiene el commit C4, necesitaremos revisar el commit de la siguiente forma:

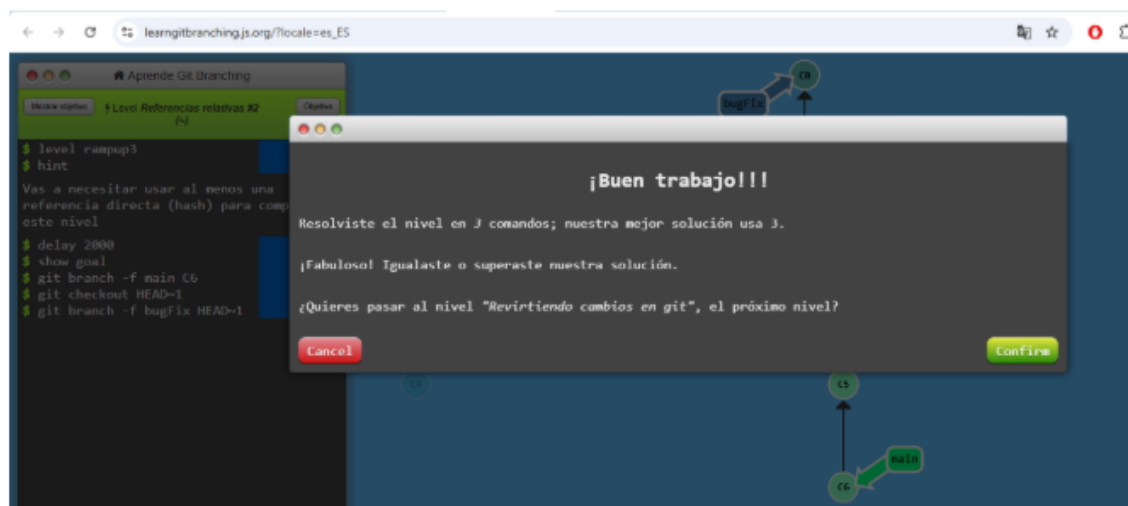
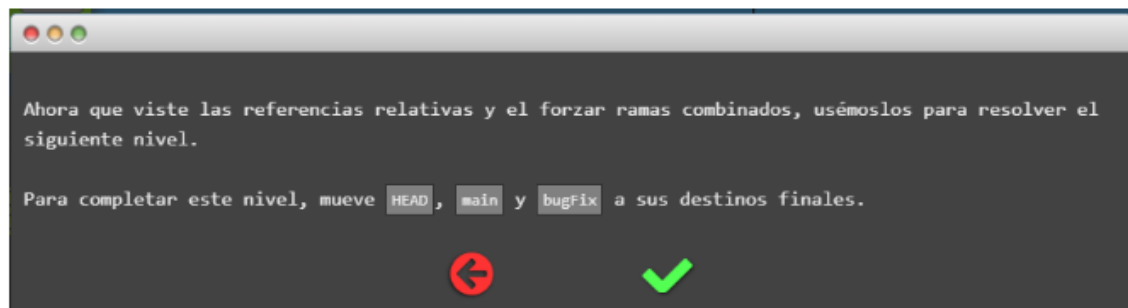


LVL 2

Aquí, necesitamos separar el HEAD C3, para ello:

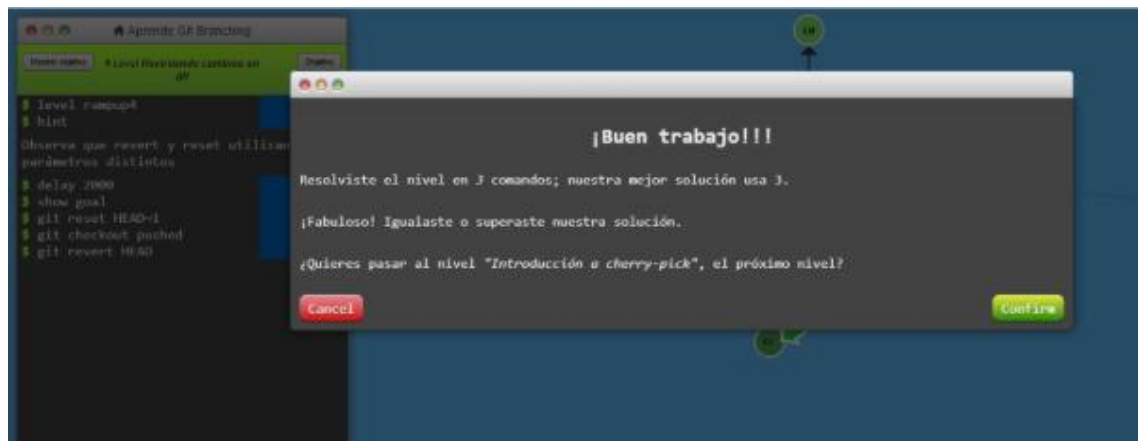
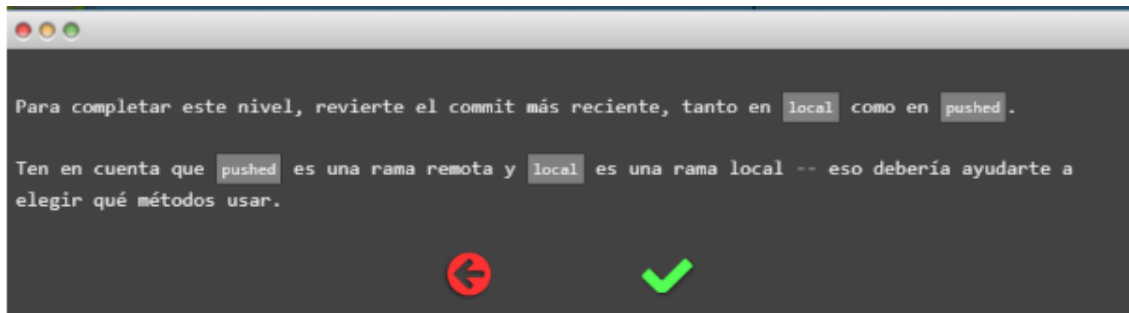
LVL 3

Tendremos que mover la rama a C6, BugFix y HEAD a C1 con los siguientes comandos:



LVL 4

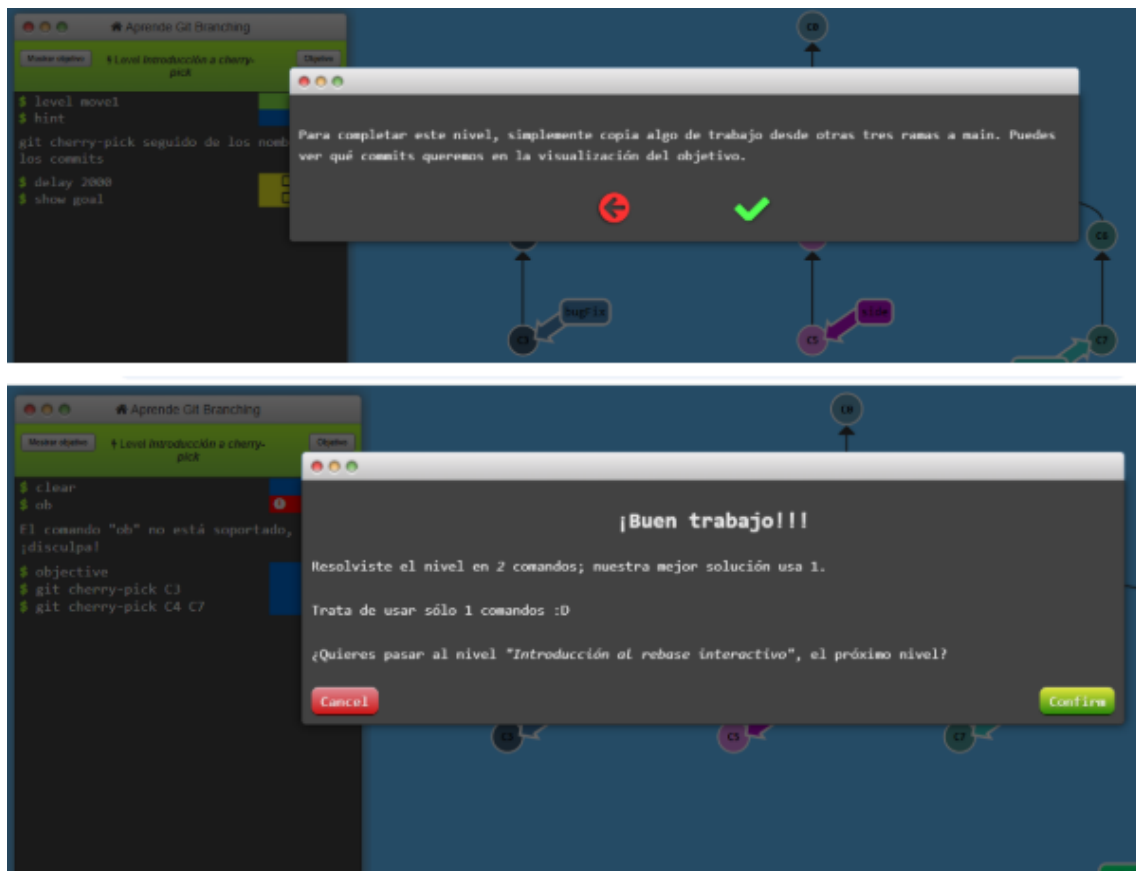
Retrocedemos al commit en la rama actual y mantenemos los cambios, cambiamos a la rama pushed y quitamos los archivos del área staging dejándolo en el directorio de trabajo



NIVELES APARTADO MOVE

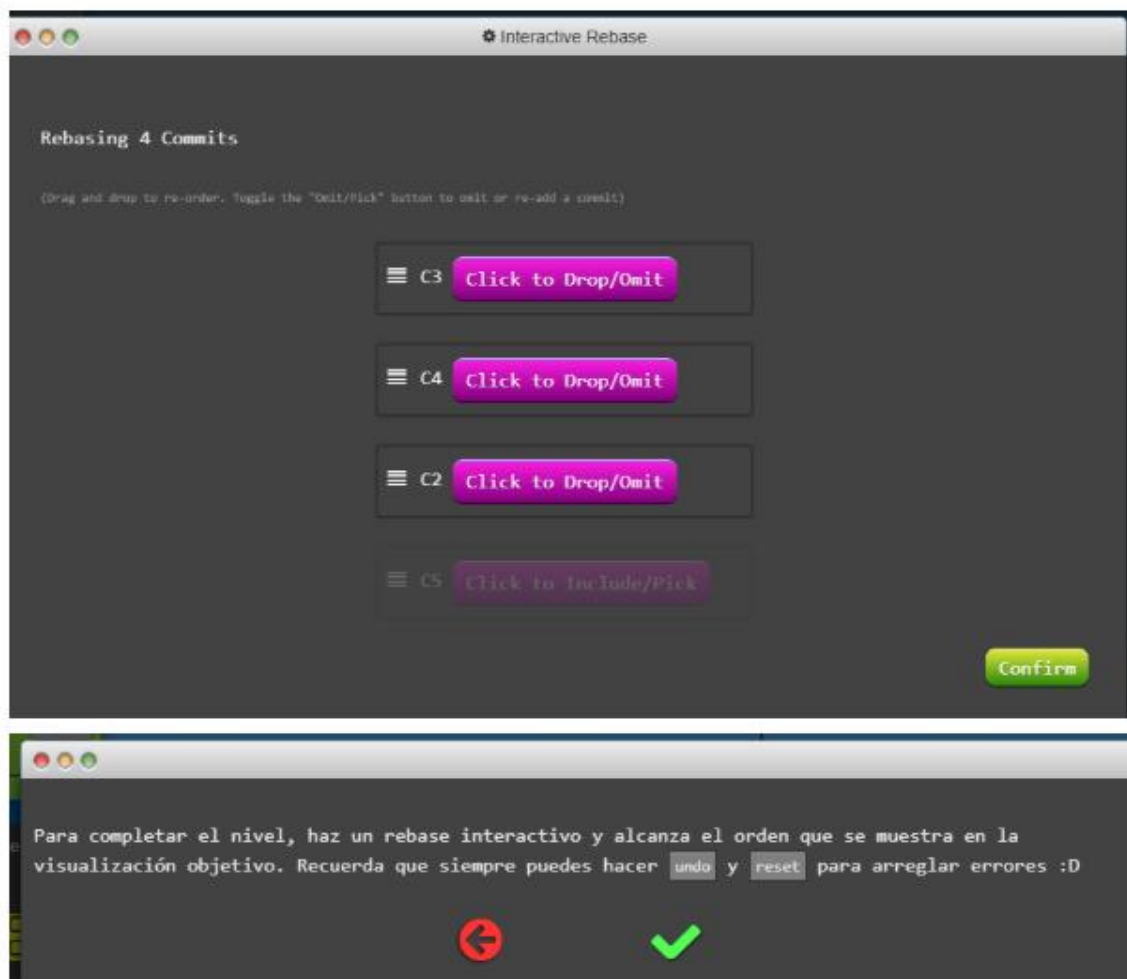
LVL 1

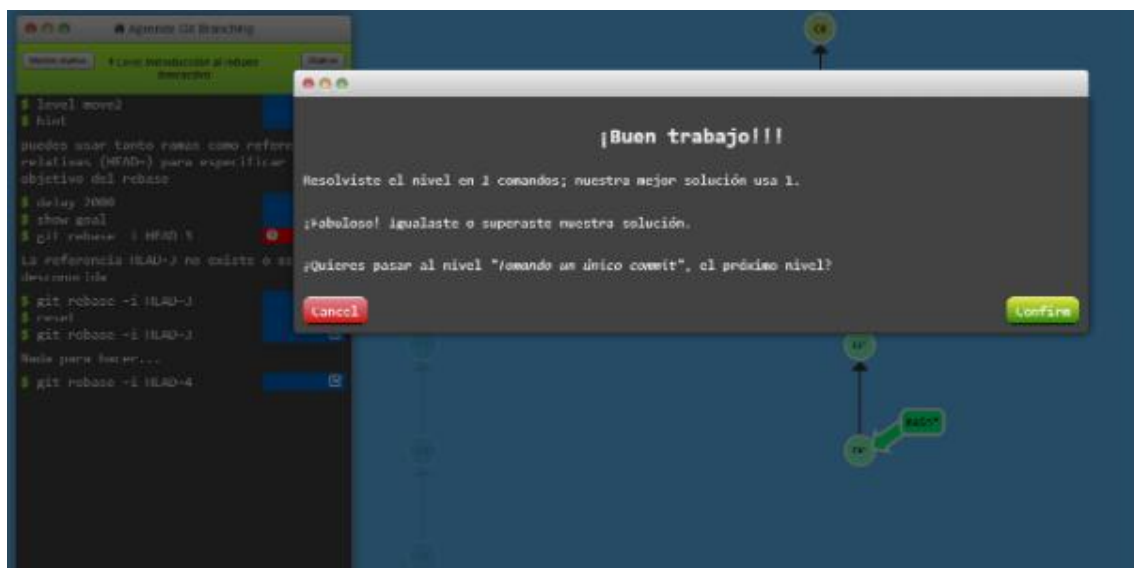
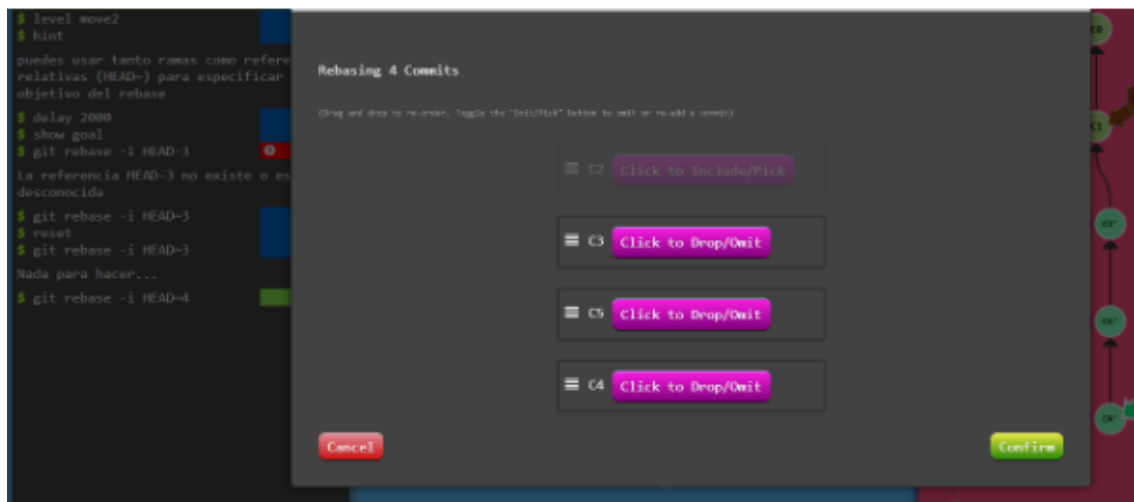
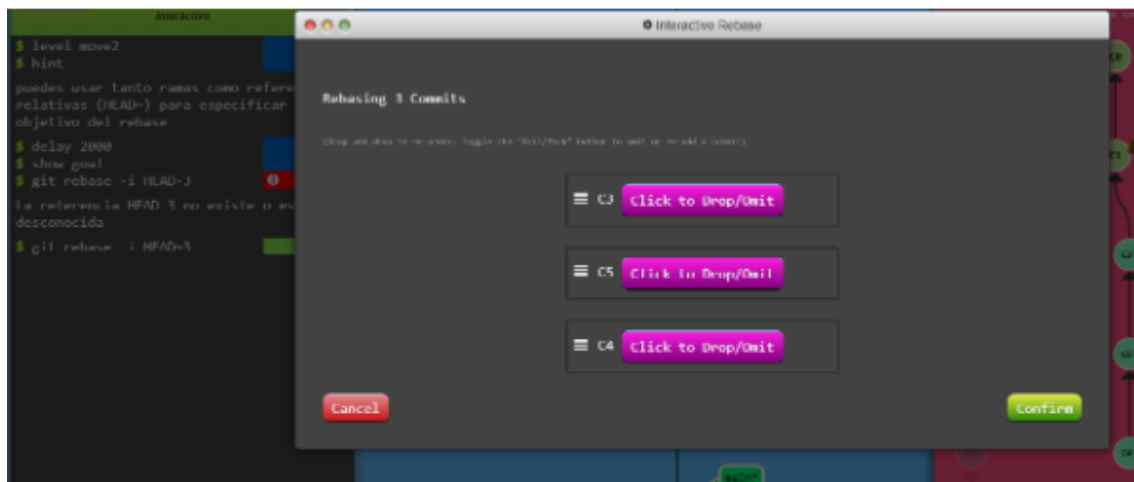
Copiamos el Commit C3 a la rama actual, aplicando cambios y copiamos los commits C4 a C7 también a la rama actual.



LVL 2

Hacemos git rebase y omitimos C5. Colocamos los demás commits y lo confirmamos con los siguientes comandos:

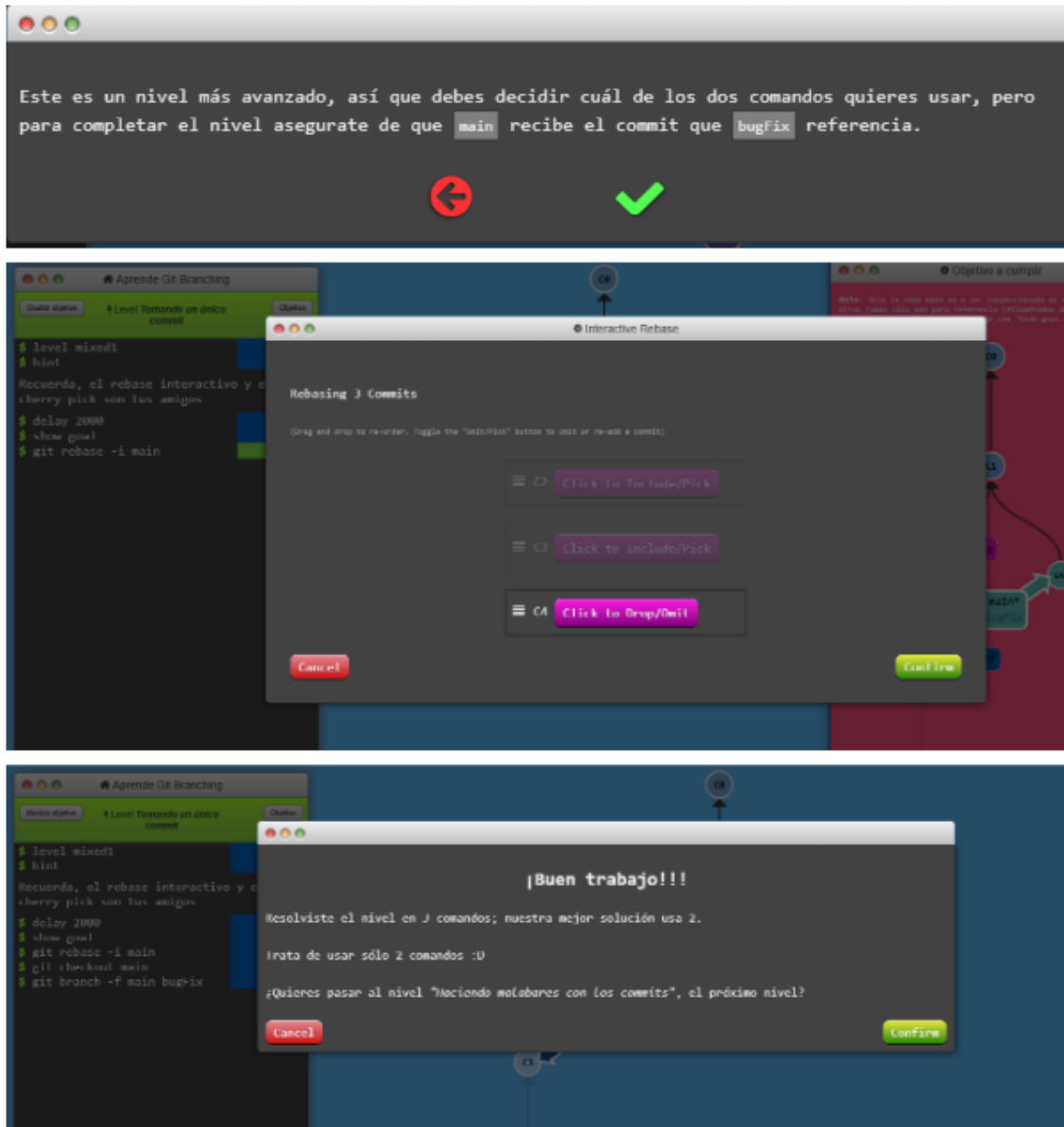




NIVELES APARTADO MIXED

LVL 1

Iniciamos el rebase interactivo y reorganizamos los commits de la rama sobre main. Cambiamos el main, y forzamos que bugFix apunte al commit del main mediante los siguientes comandos:



LVL 2

Iniciamos el rebase sobre los 2 commits de la rama actual, los ordenamos, modificamos el ultimo commit, volvemos al rebase sobre los 2 commits y los movemos a la rama actual para que se aplique en el main.

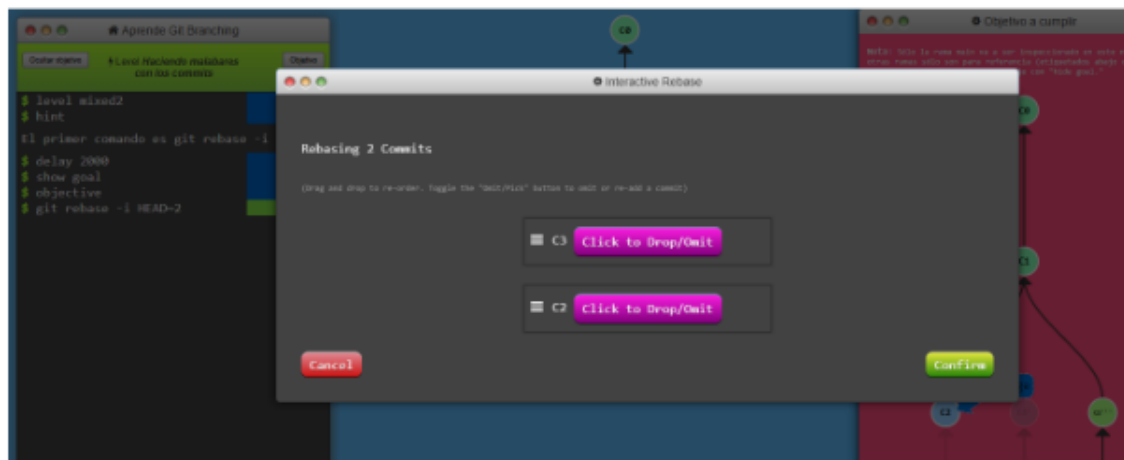
Superaremos esta dificultad haciendo lo siguiente:

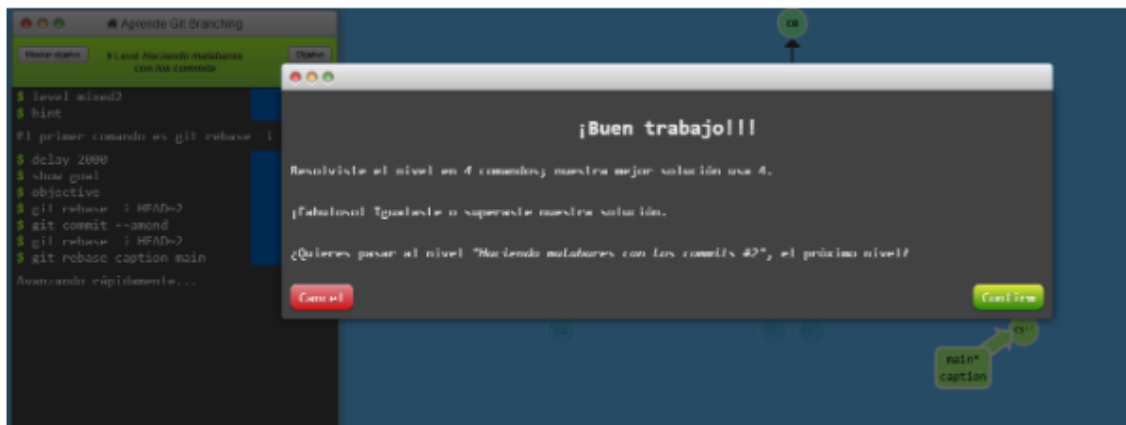
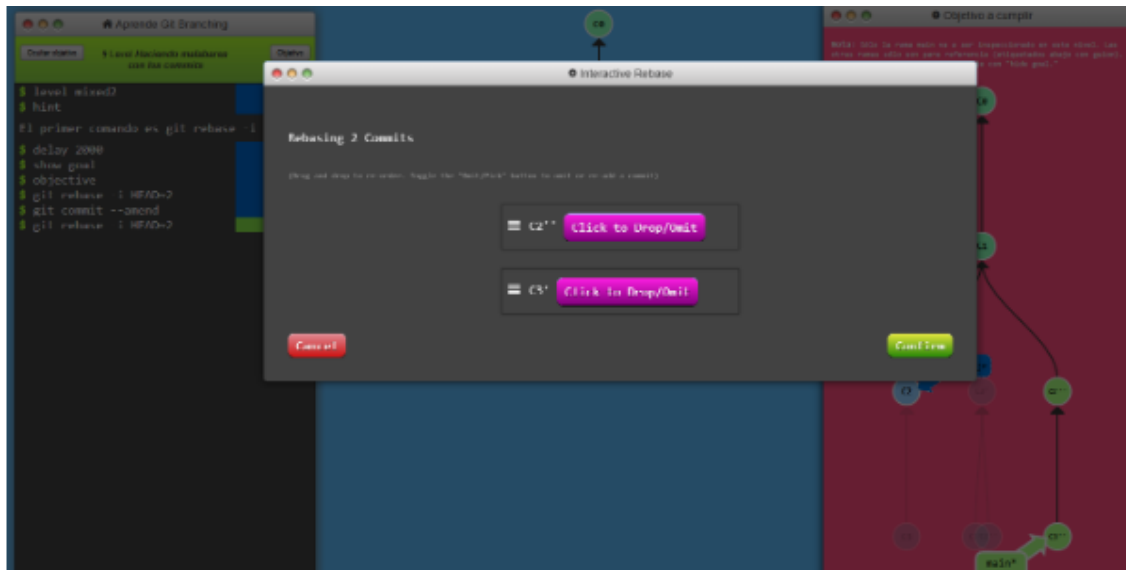
- Vamos a reordenar los commits para que el que queremos cambiar quede arriba de los demás con `git rebase -i`
- Vamos a hacer `git commit --amend` para aplicar la ligera modificación
- Después vamos a reordenar los commits a como estaban con `git rebase -i`
- Finalmente, vamos a mover main a esta parte actualizada de nuestro árbol de commits para terminar el nivel (usando el método que prefieras)

Hay varias maneras de lograr este objetivo en general (ya te veo haciéndole ojitos al cherry-pick), y veremos algunos más después, pero por ahora concentrémonos en esta técnica.

Por último, presta atención al estado final -- como movemos los commits dos veces, ambos quedan con un apóstrofe. El commit que corregimos tiene un apóstrofe extra, y así nos queda nuestro árbol final

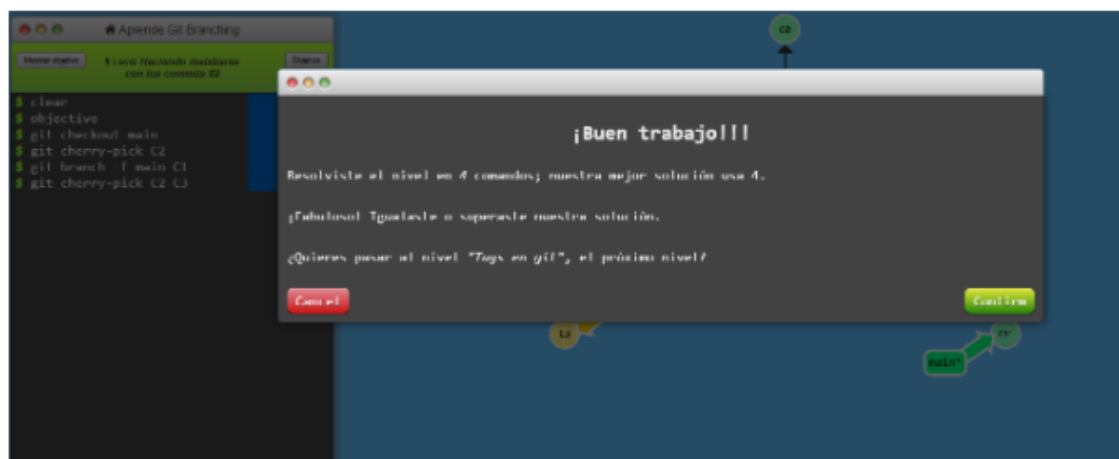
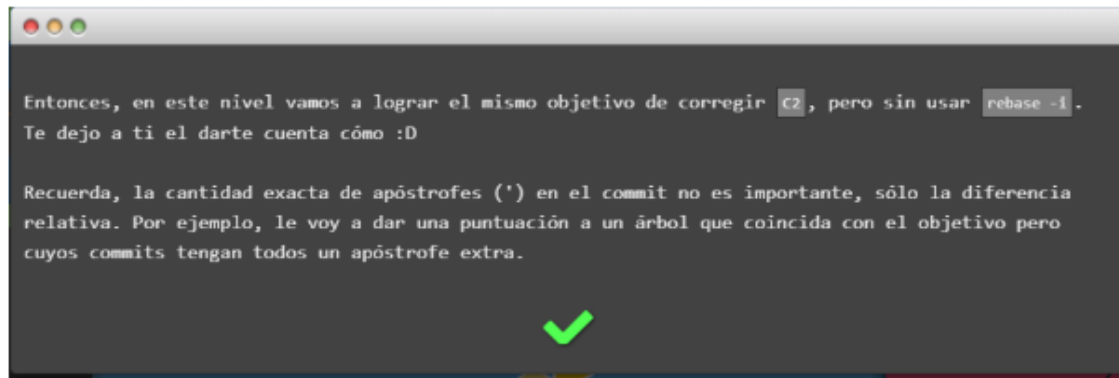
Habiendo dicho eso, puedo comparar los niveles basándome ahora en la estructura y las diferencias relativas de apóstrofes. Mientras que tu rama `main` tenga la misma estructura y diferencias relativas de apóstrofes, te voy a dar el puntaje completo.





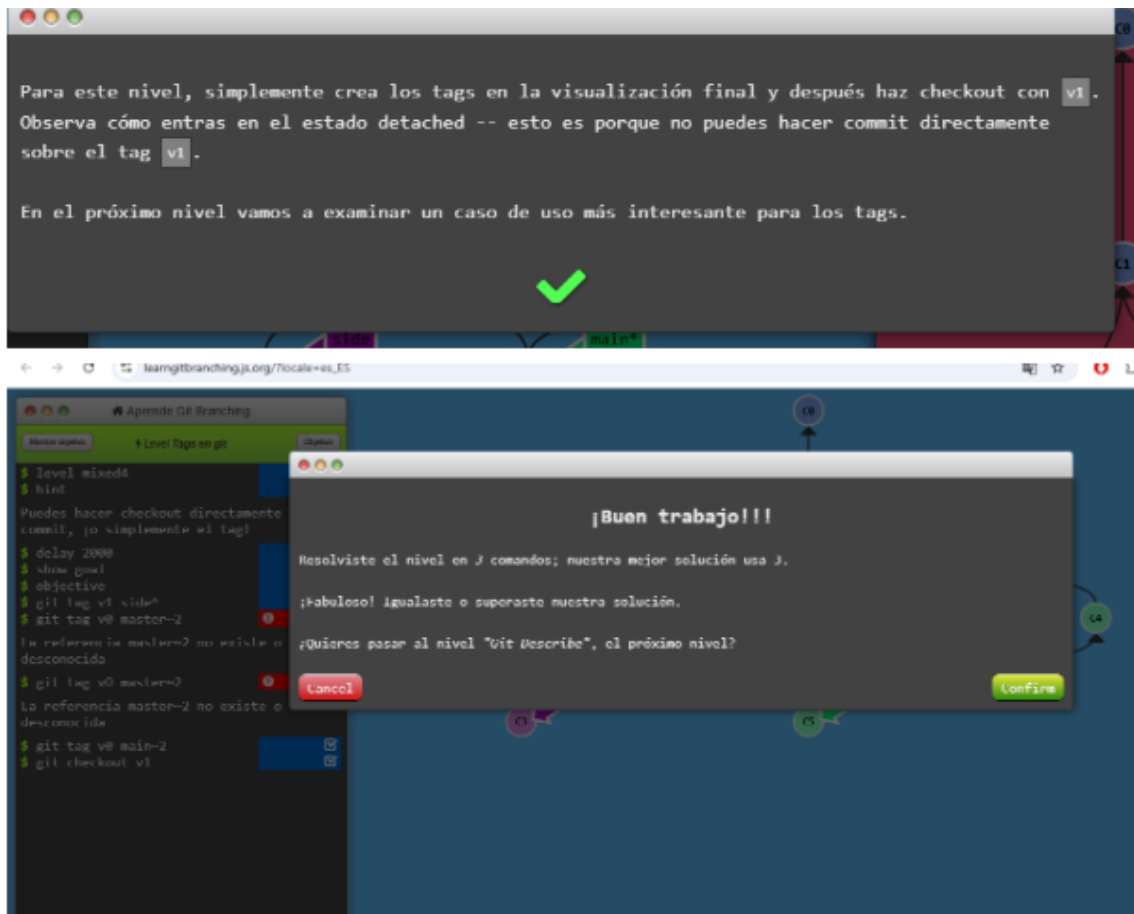
LVL 3

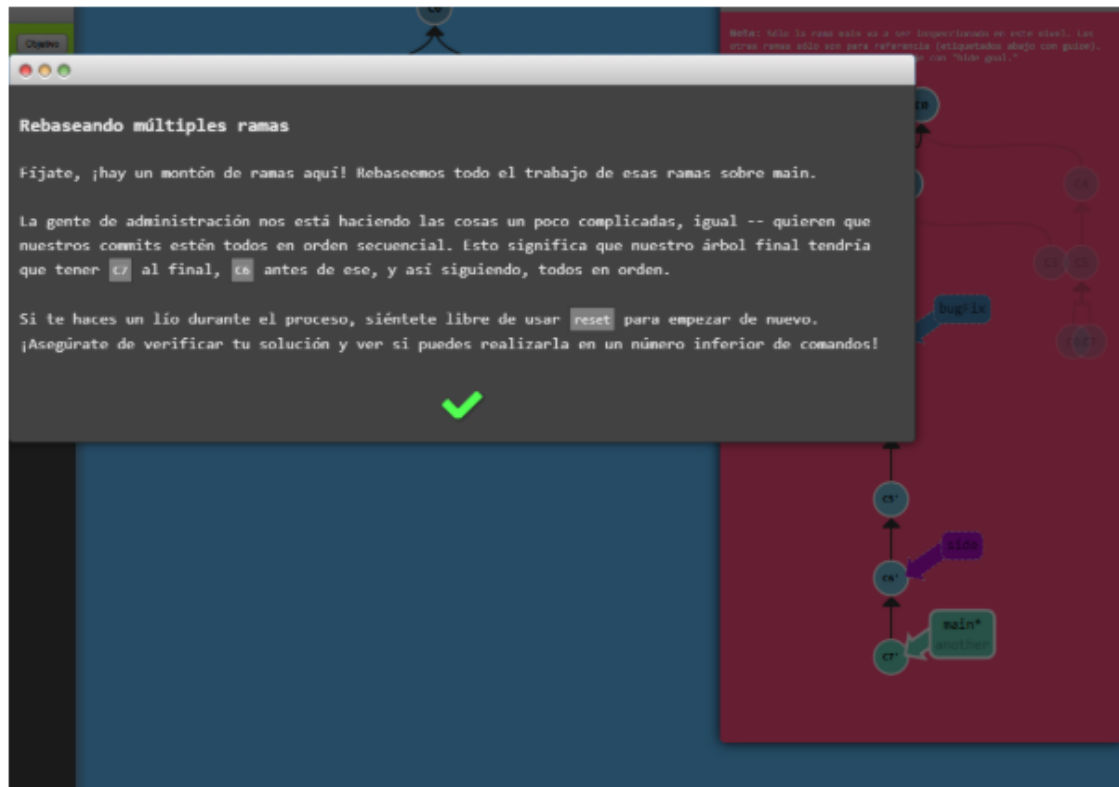
Cambiamos a main, copiamos el commit C2 a main con un nuevo commit, forzamos a main para que apunte a C1 y copiamos los commits C2 y C3 a main en orden.



LVL 4

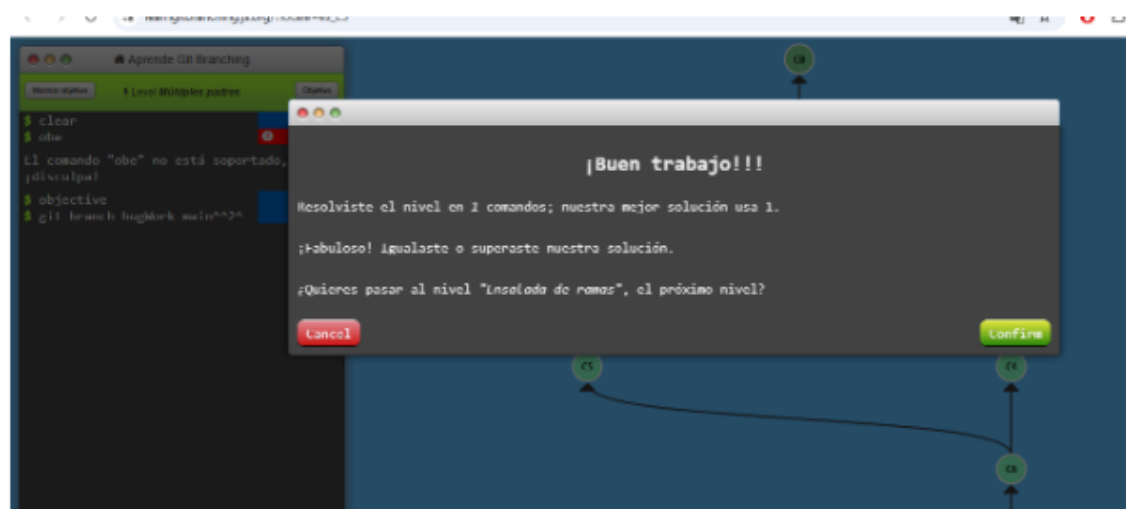
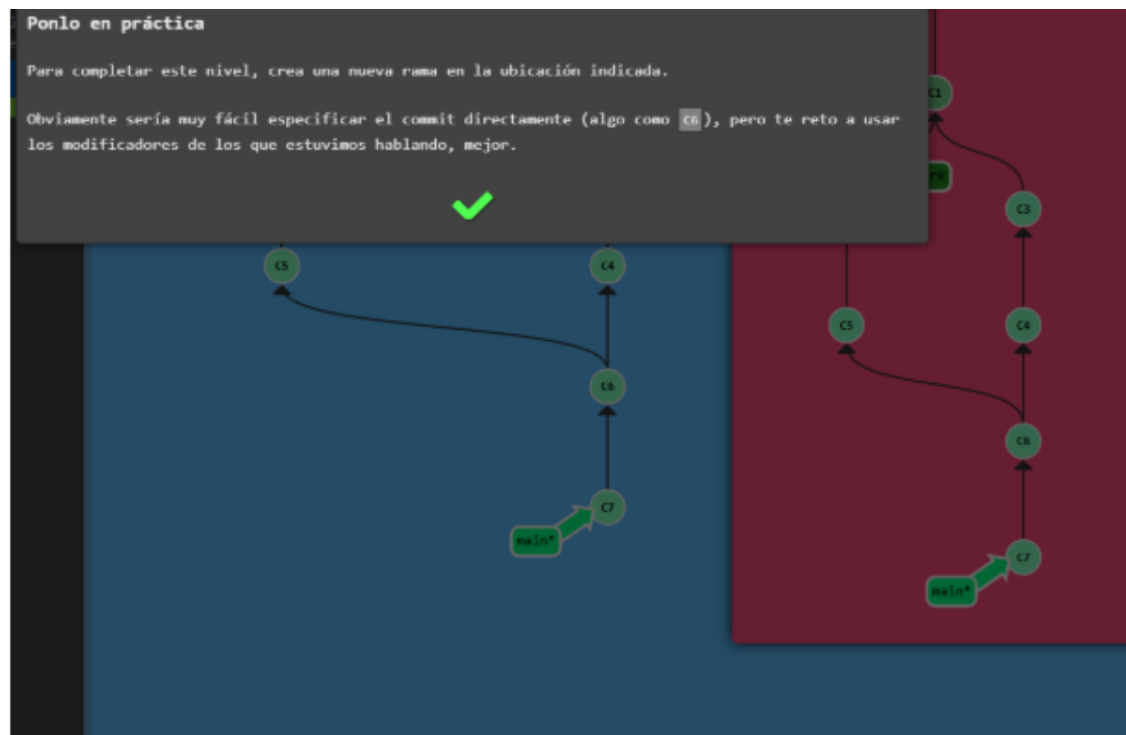
Creamos etiquetas v1 y v0, apuntamos al commit padre a v1 y v0 al commit de atrás y cambiamos head a la etiqueta v1.





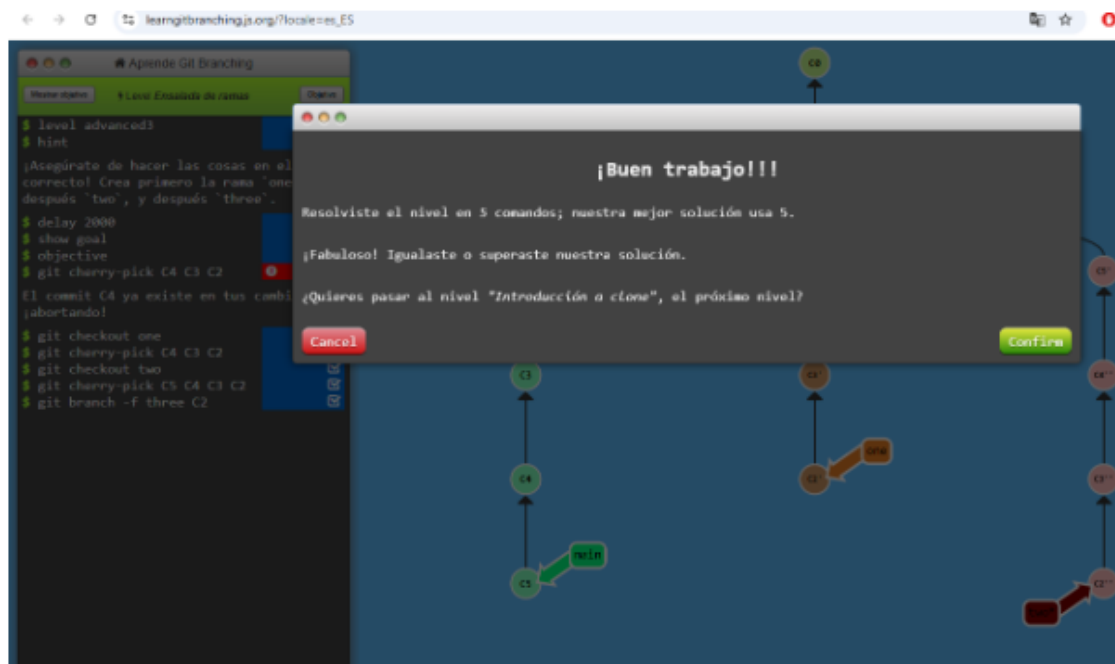
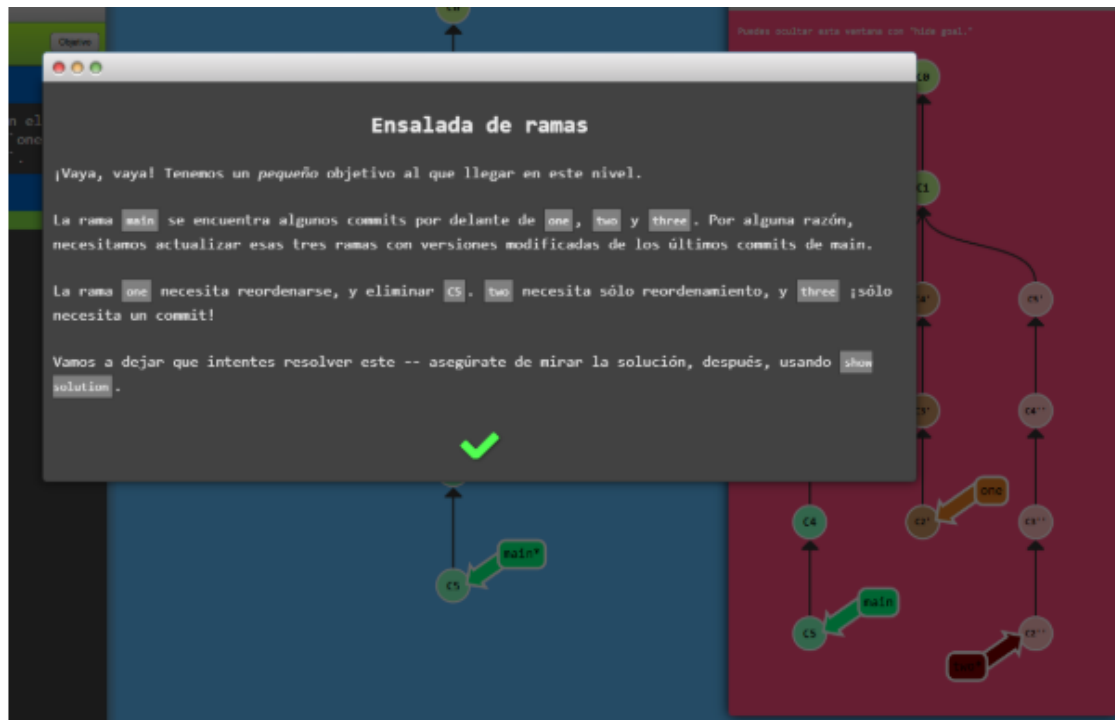
LVL 2

Creamos una rama llamada bugMark que apunte al commit 2 pasos atrás del main, básicamente subes el nivel en el historial de main y empiezas la rama desde ese commit



LVL 3

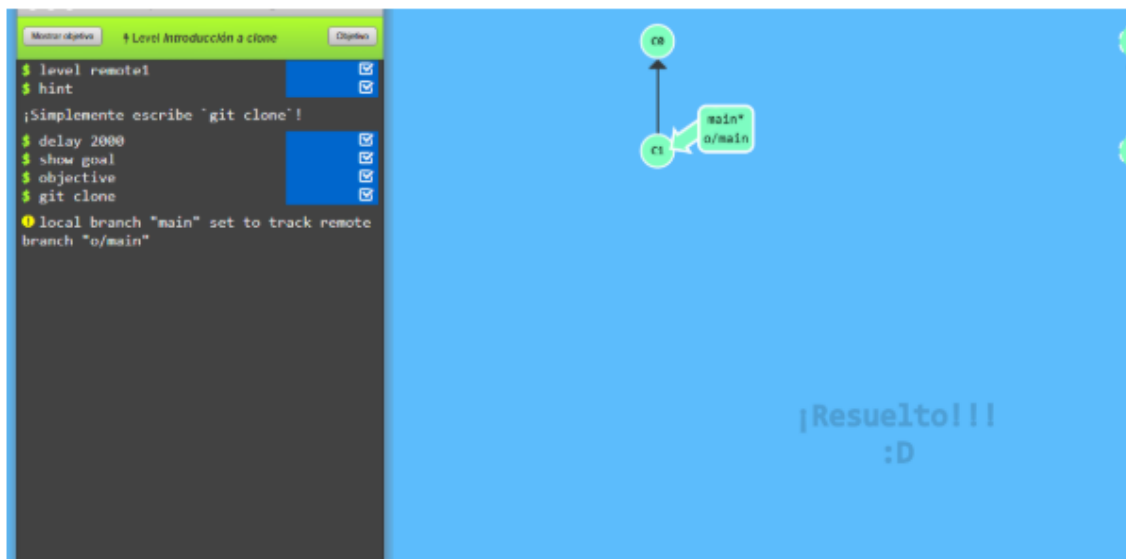
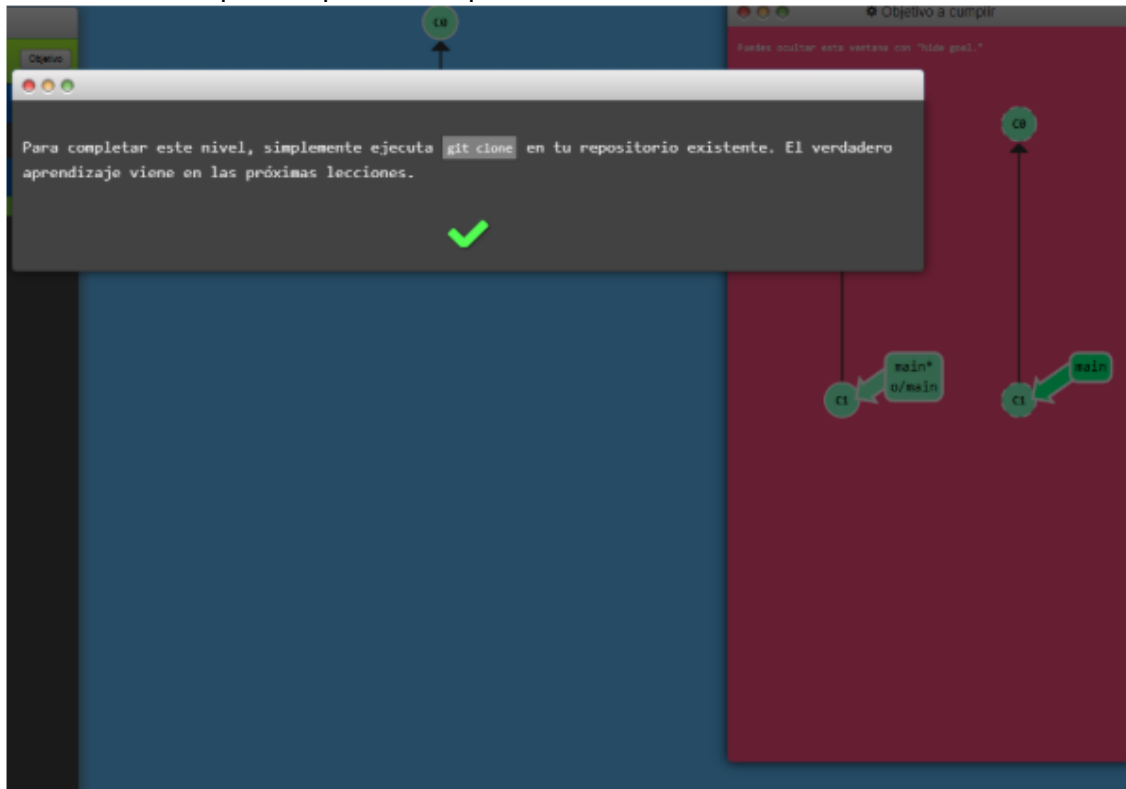
Cambiamos a one, copiamos los commits C4, C3 y C2 a la rama one como nuevos commits, cambiamos a la rama one, copiamos los commits C5 C4 C3 y C2 a la rama two y forzamos que la rama three apunte a C2 sobrescribiendo esa rama.



NIVELES APARTADO REMOTE

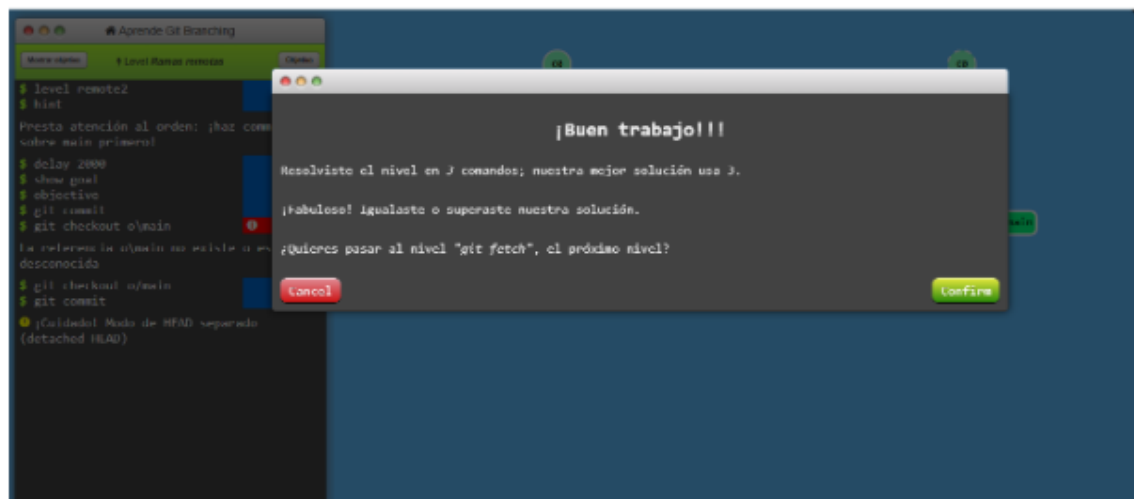
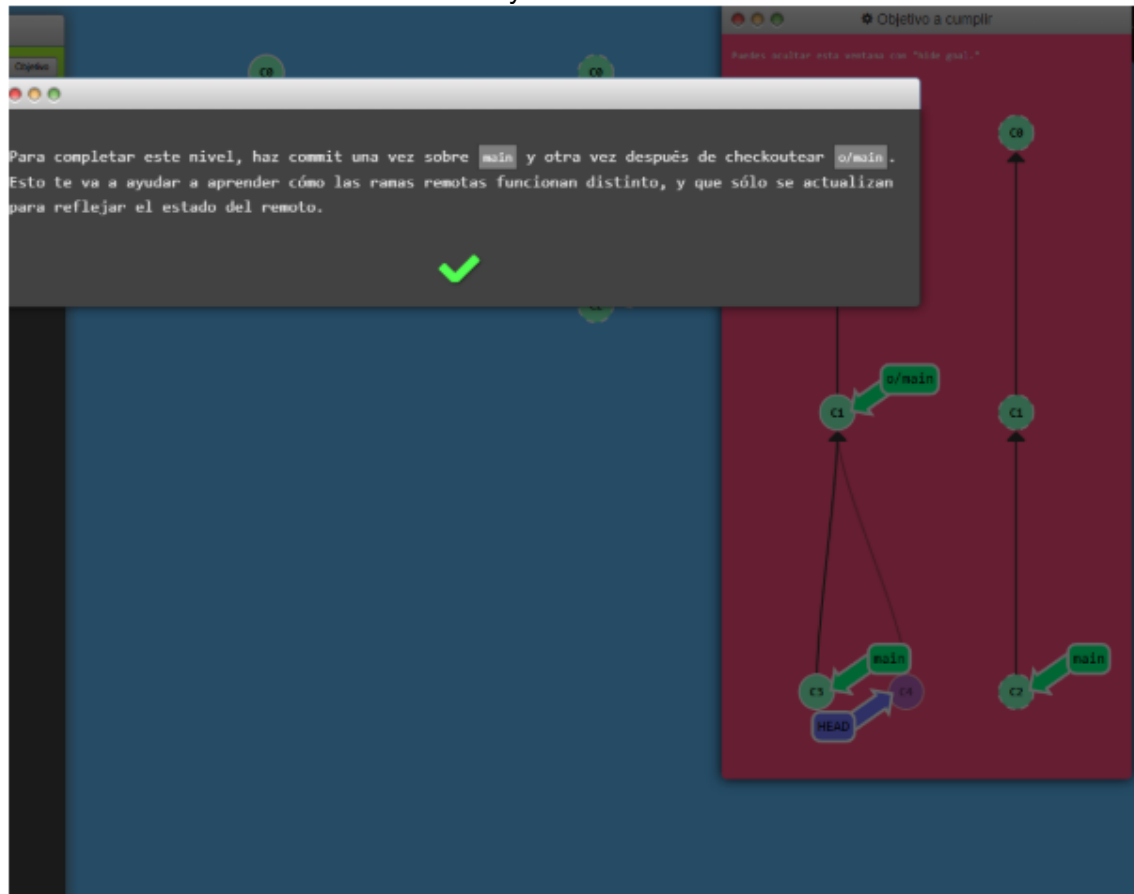
LVL 1

Hacemos una copia completa del repositorio



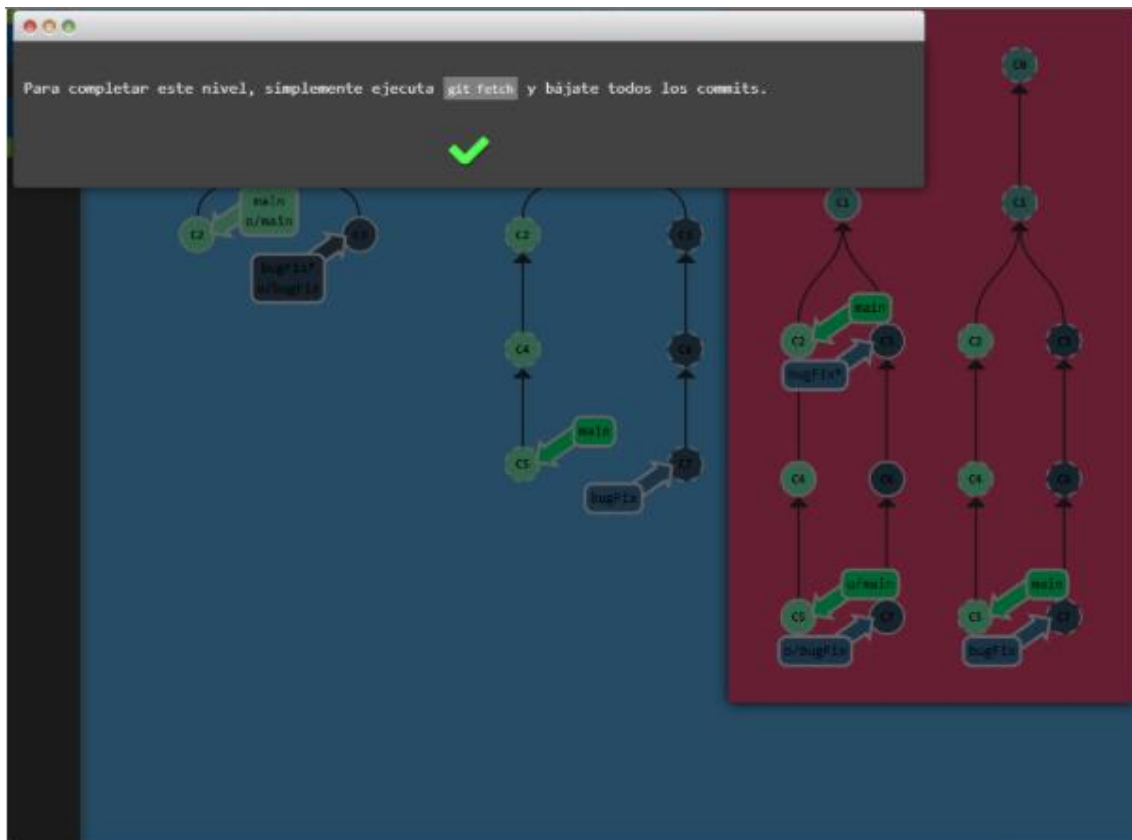
LVL 2

Creamos un nuevo commit con los cambios del área staging de la rama actual, cambiamos a la rama remota de main y creamos un commit en la rama o/main



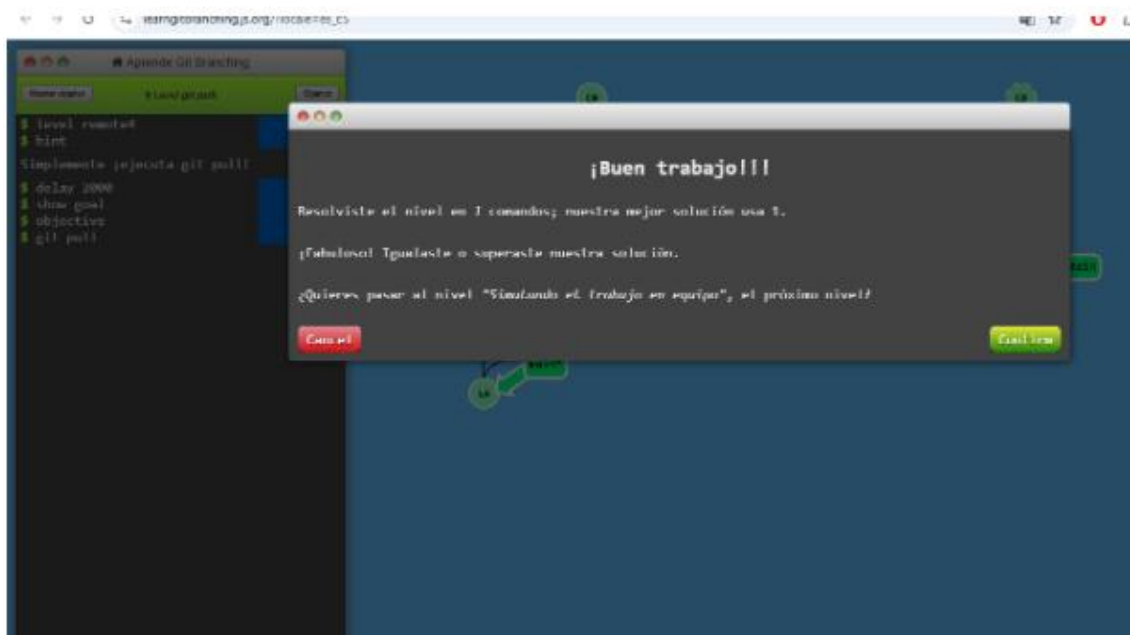
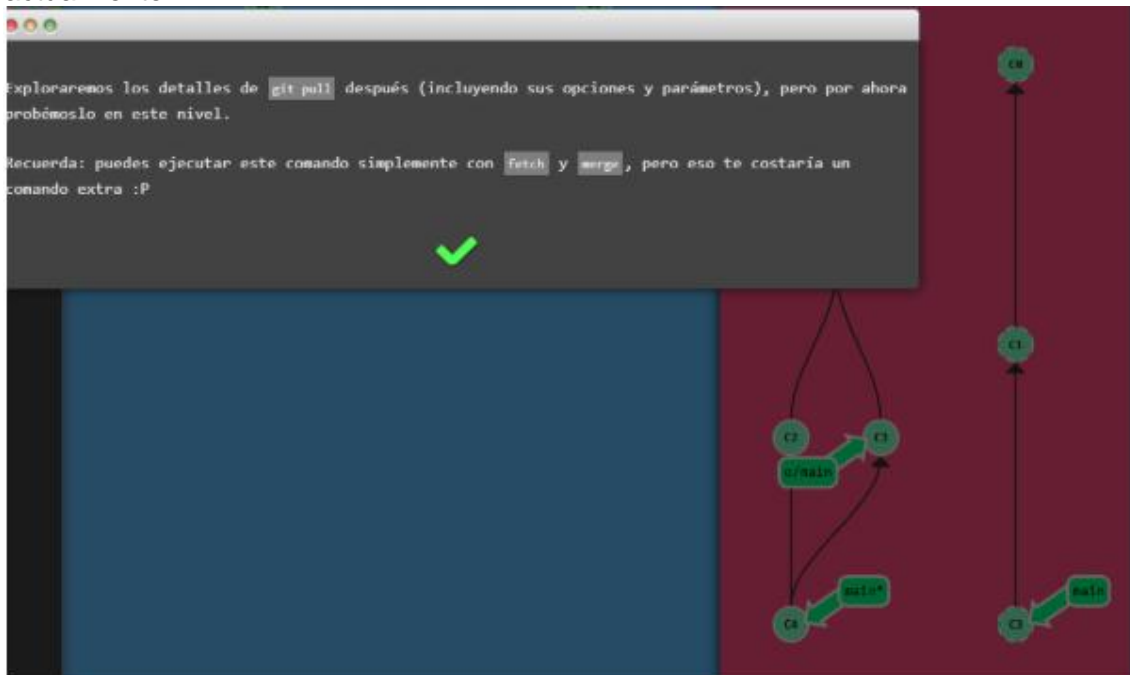
LVL 3

Descargamos los cambios del repositorio remoto al local para actualizar la rama remota sin afectar al trabajo actual.

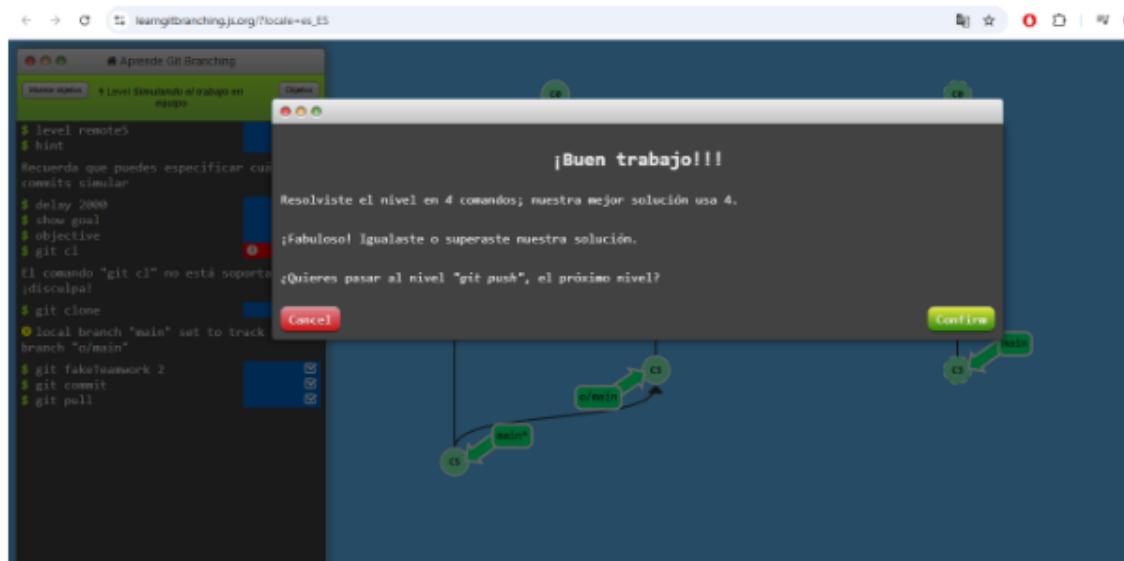
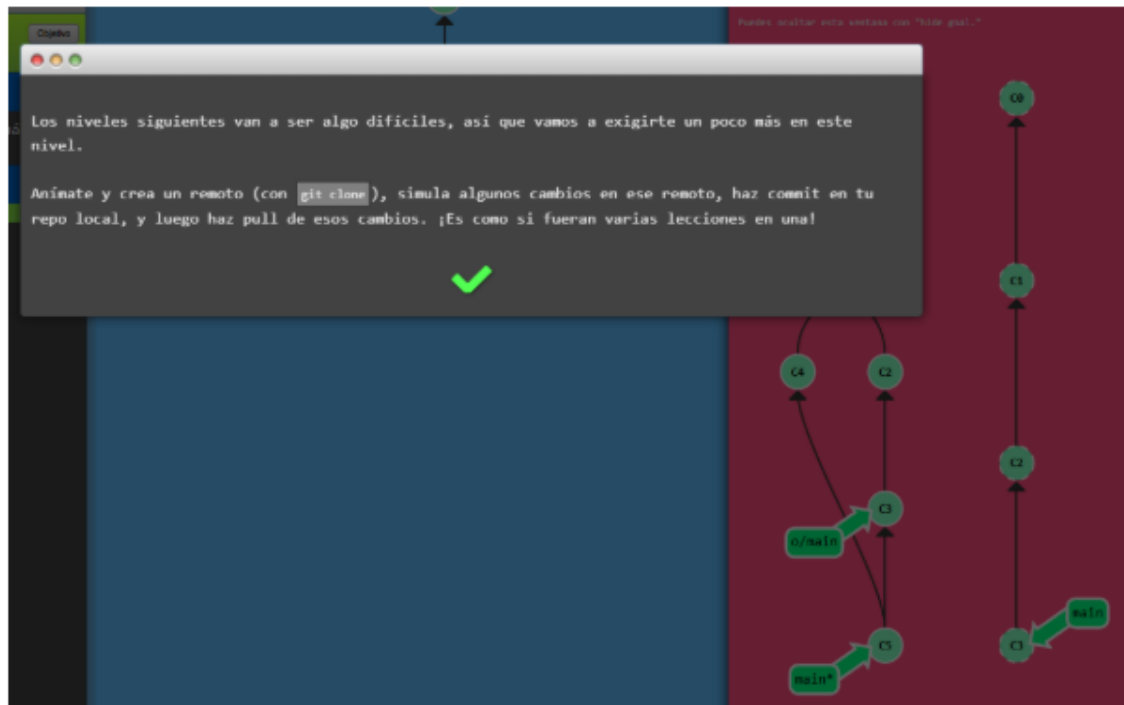


LVL 4

Descargamos los datos del repositorio remoto y los ponemos en la rama que estemos actualmente.

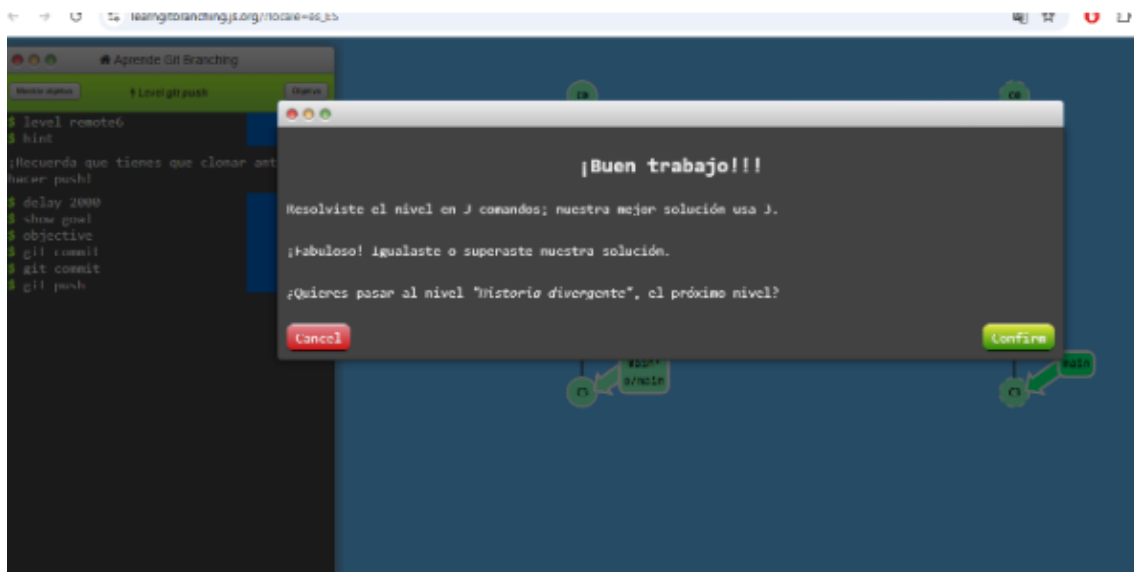
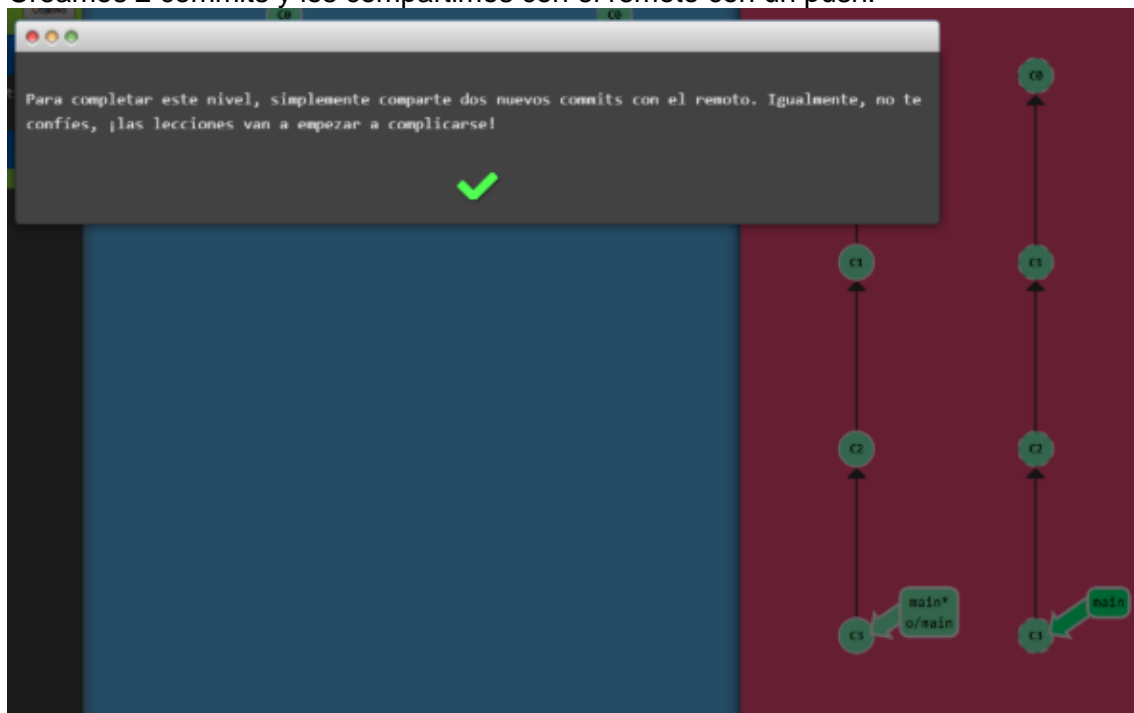
LVL 5

Clonamos un repositorio remoto a nuestra rama actual (local), creamos un nuevo commit con los cambios que tengamos en el área staging y descargamos los cambios del repositorio remoto fusionándolos en nuestra rama actual.



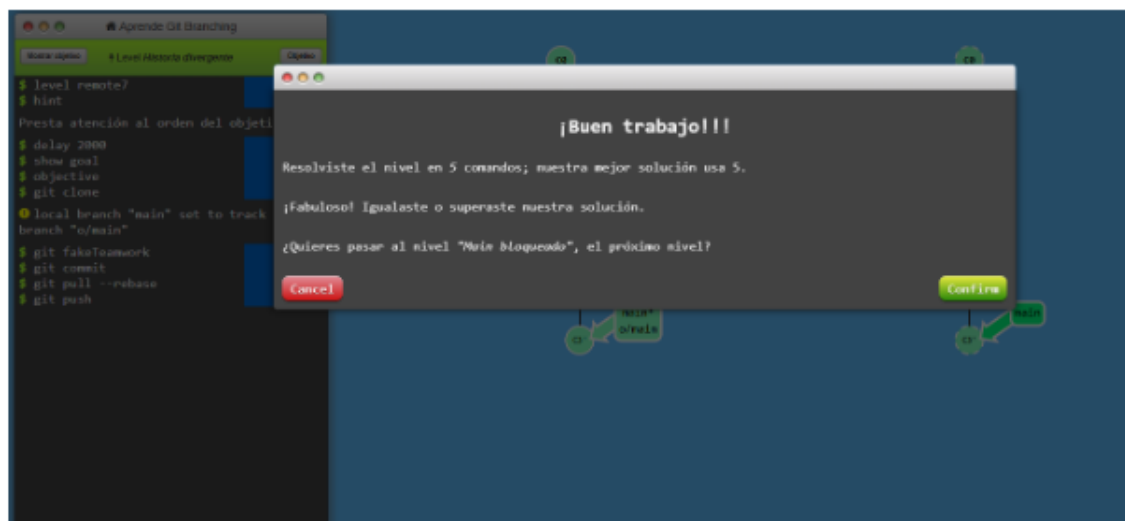
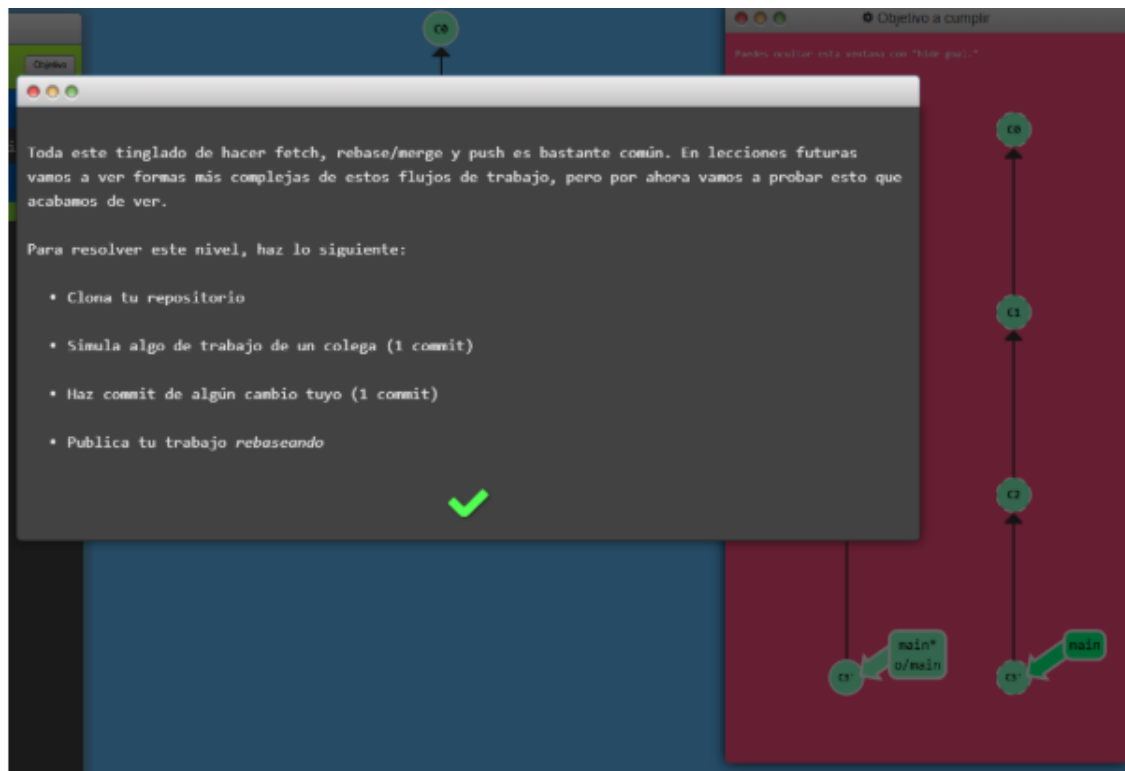
LVL 6

Creamos 2 commits y los compartimos con el remoto con un push.



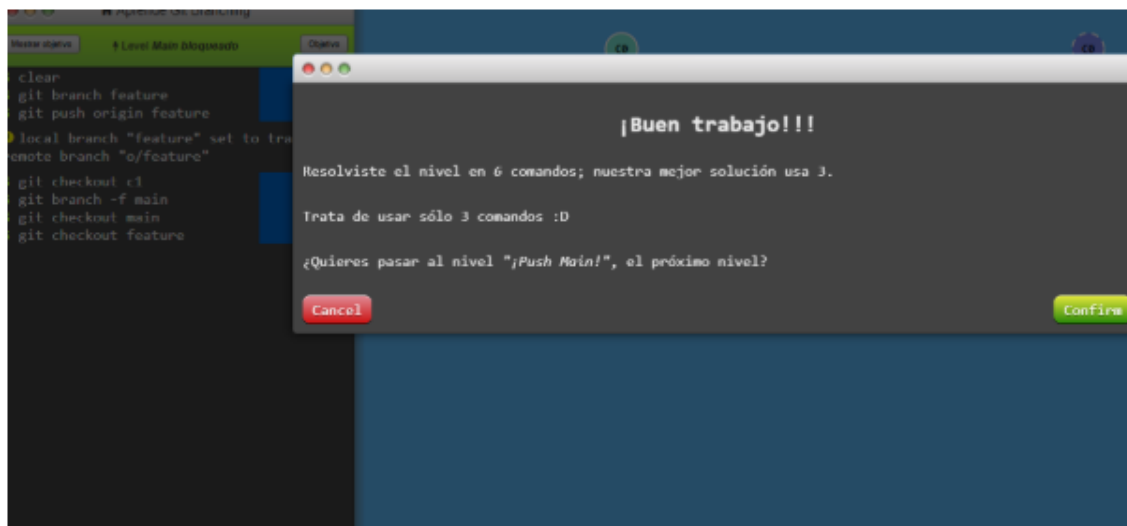
LVL 7

Clonamos el repositorio, hacemos pull en el remoto, creamos un commit con los cambios, traemos los cambios del remoto y los aplicamos y por último enviamos los commits locales al remoto, actualizando la rama.



LVL 8

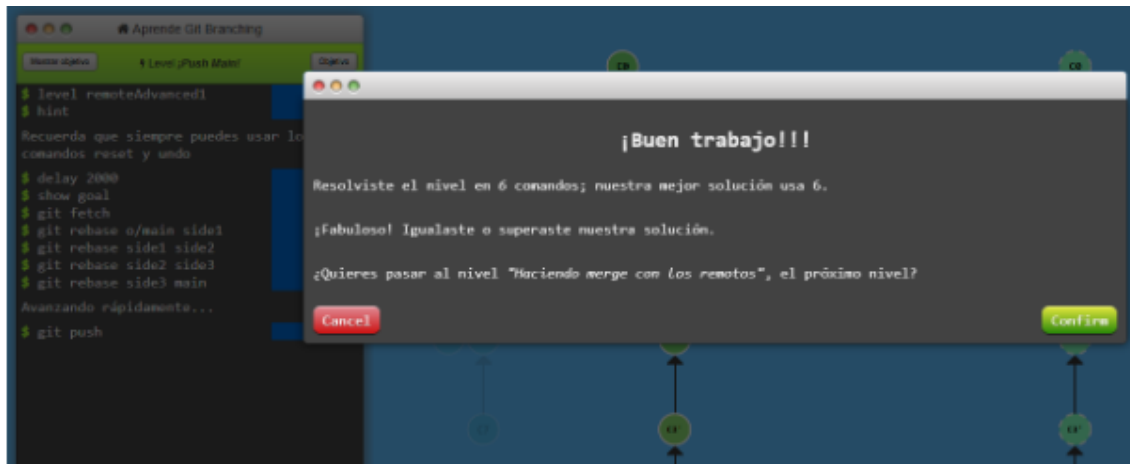
Creamos una rama llamada feature, subimos la rama feature al remoto origin, movemos head al commit c1, forzamos que main apunte a c1, cambiamos main y cambiamos la rama feature.



NIVELES APARTADO REMOTE ADVANCES

LVL 1

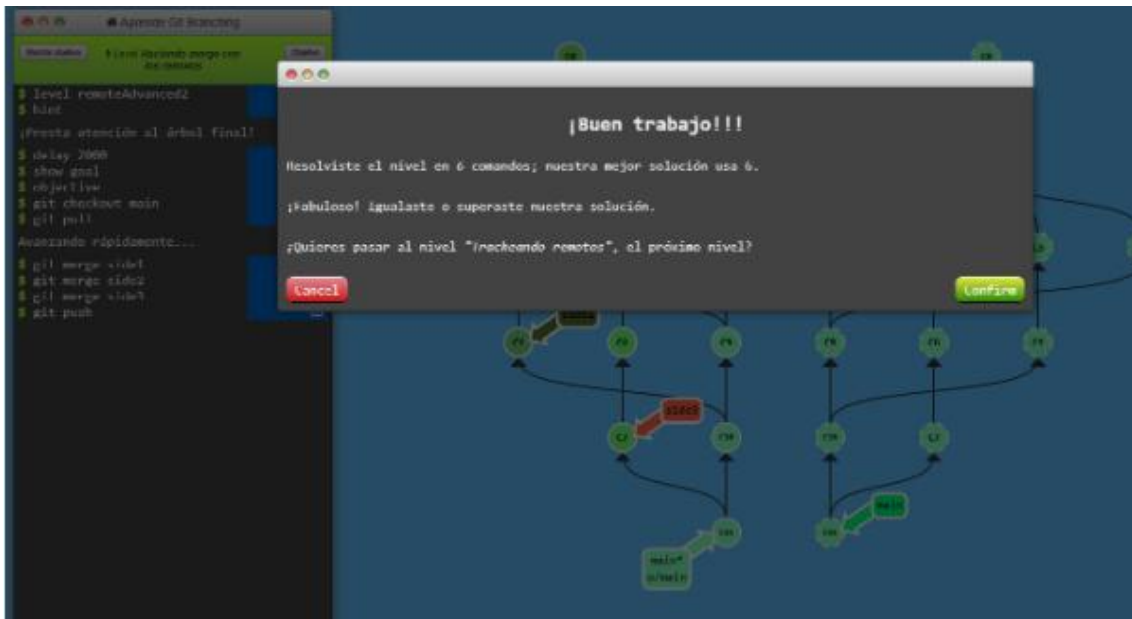
Descargamos los cambios del repositorio, movemos los commits side1 a al main remoto, side 2 a side1, side 3 a side2 y main sobre side3, por último, enviamos los commits al remoto.



LVL 2

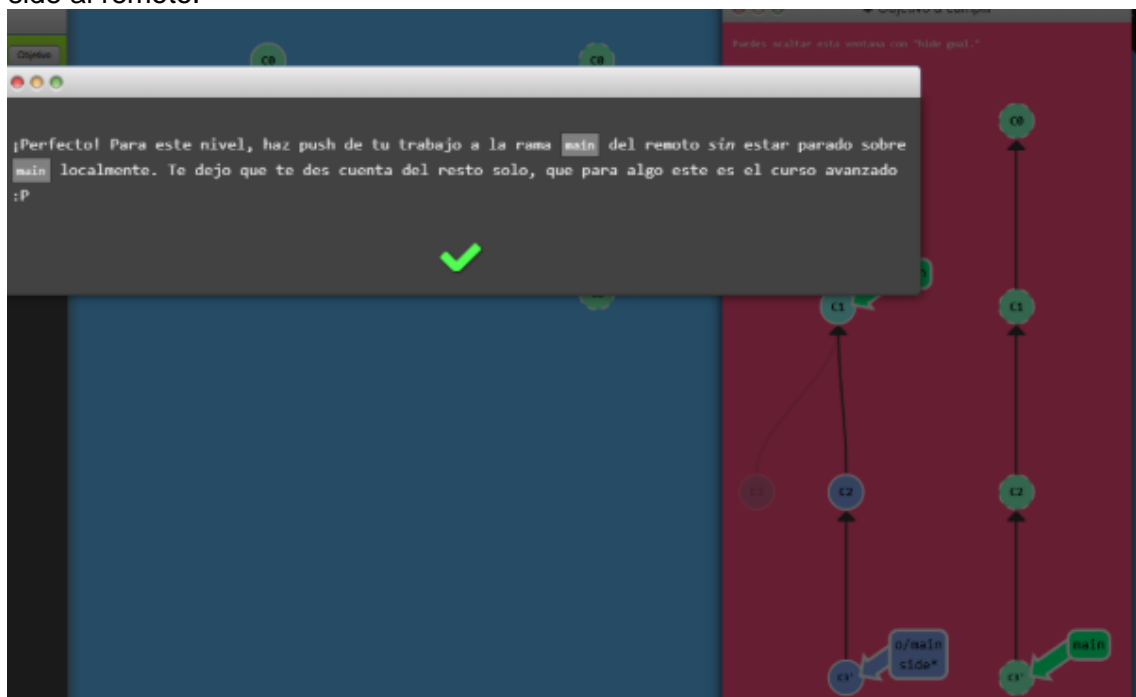
Cambiamos a la rama main, fusionamos las ramas pertinentes y enviamos los commits al remoto.

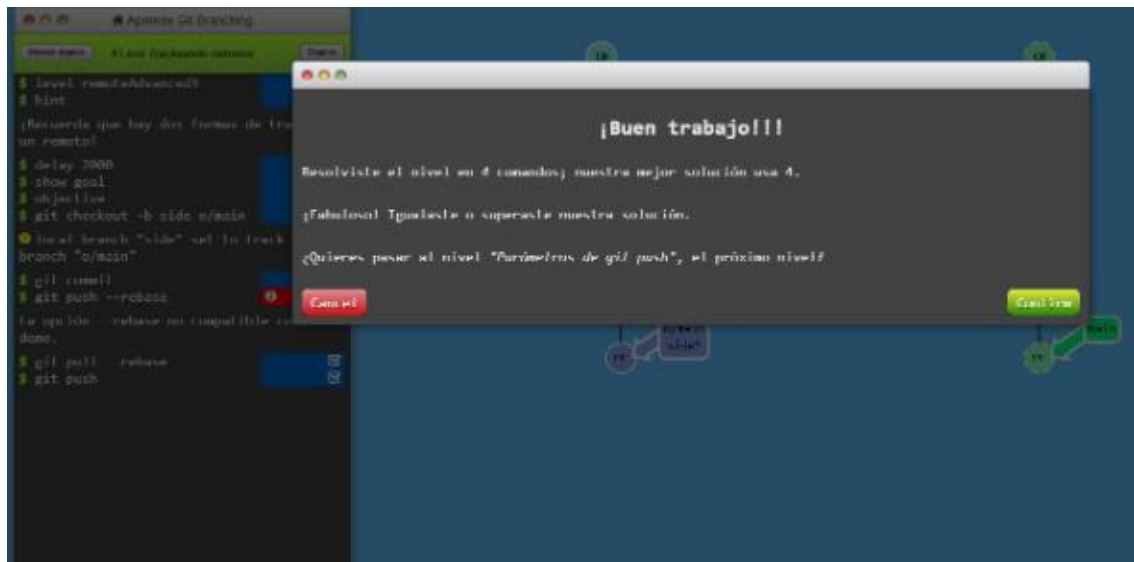




LVL 3

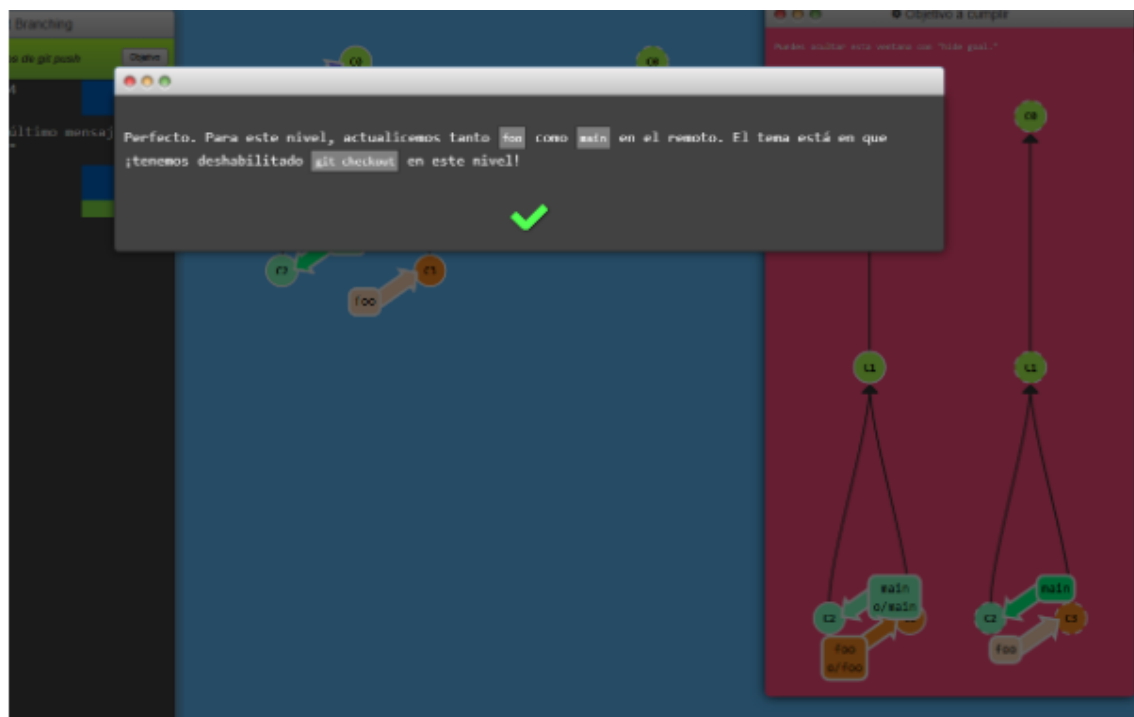
Creamos y cambiamos a una nueva rama side basada en la remota de main, creamos un commit con los cambios de side, traemos los cambios de la remota de reléase y lo fusionamos en la rama actual de side y por último enviamos los commits de la rama side al remoto.

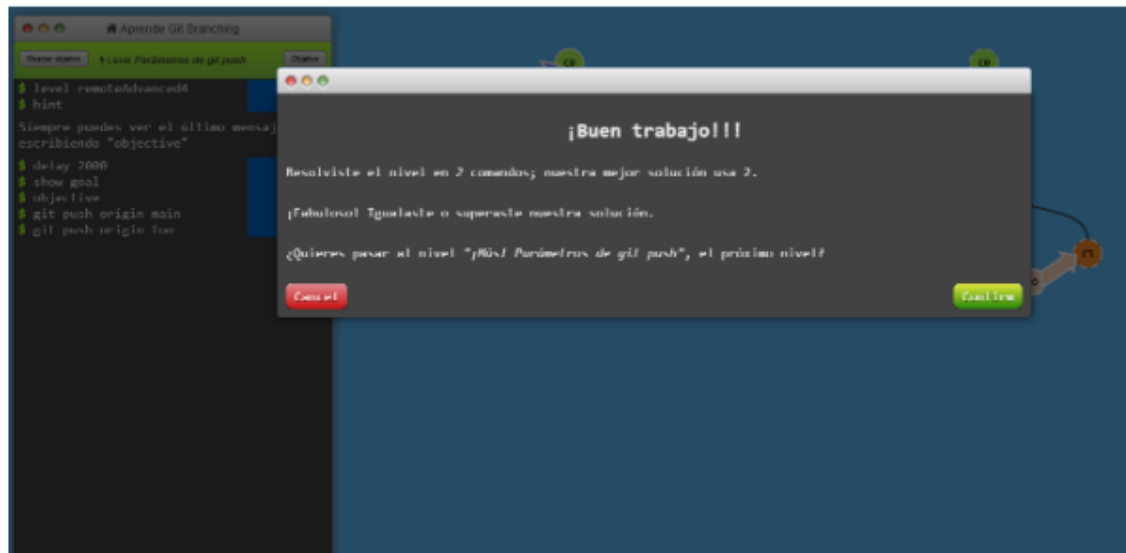




LVL 4

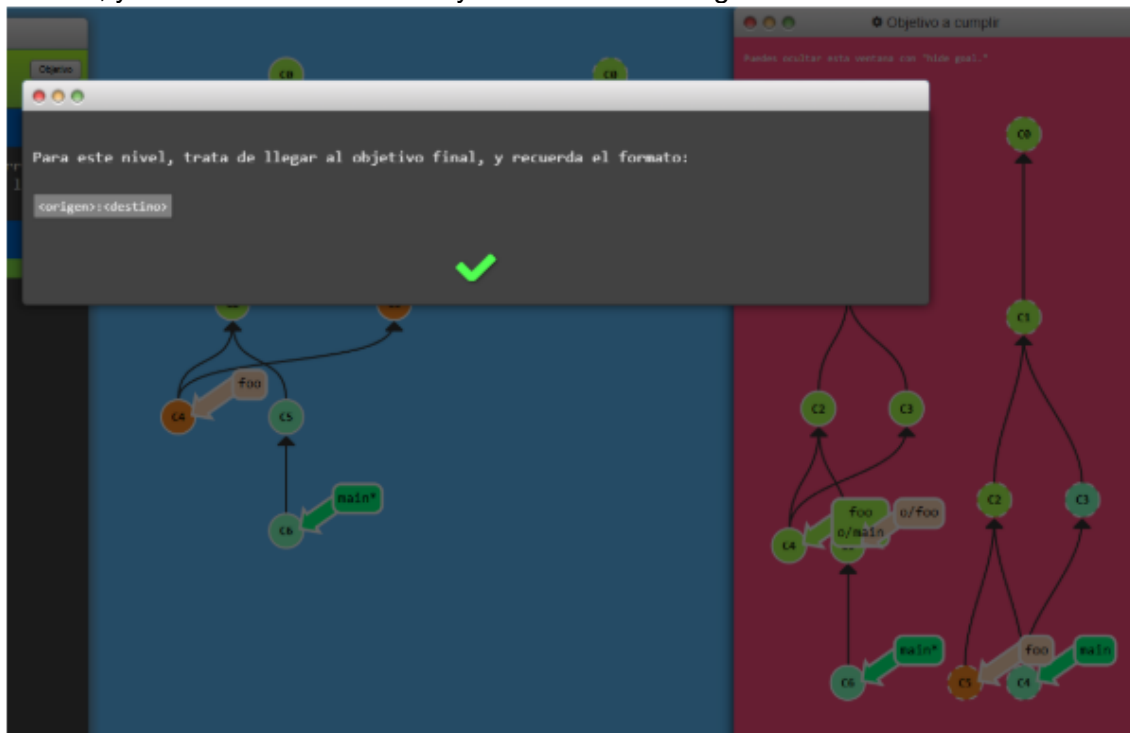
Enviamos los commits del main al remoto origin actualizando el main y enviamos la rama foo al remoto origin actualizando foo del remoto.





LVL 5

Creamos y actualizamos foo en el remoto apuntando al commit del main, no al último commit, y enviamos las ramas foo y main al remoto origin.





LVL 6

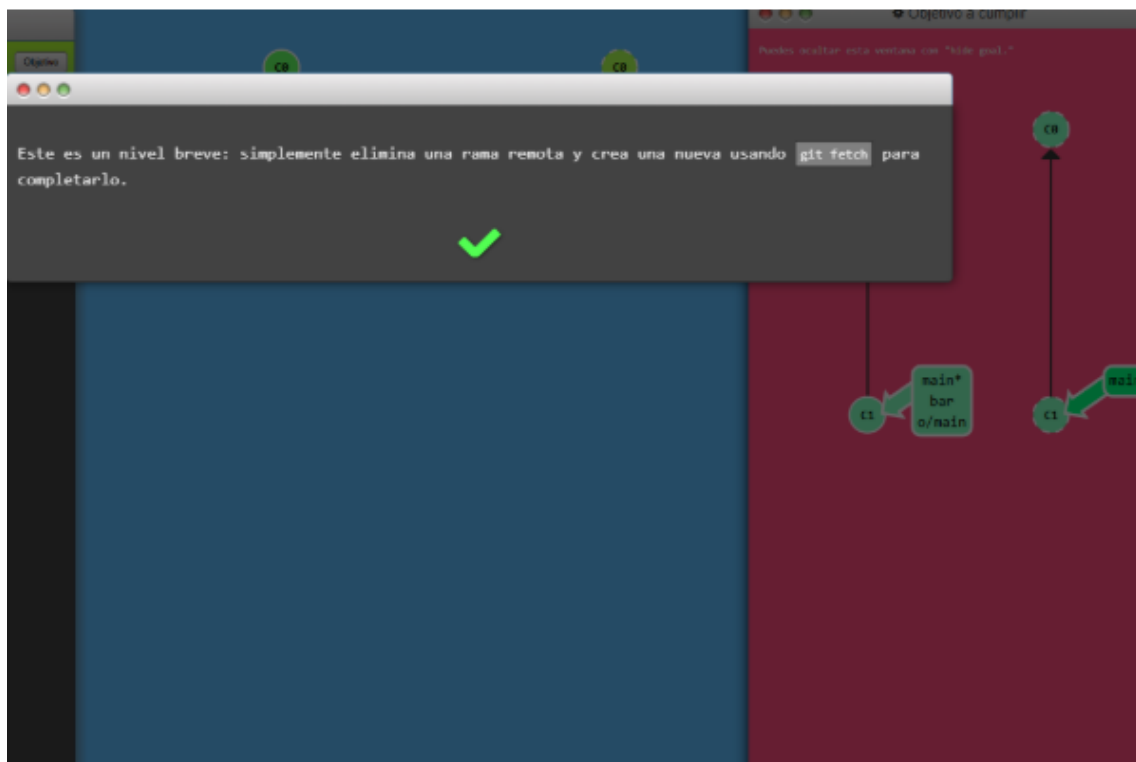
Traemos C3 y C6 del remoto y actualizamos foo y main, cambiamos a la rama foo y fusionamos main en foo en un nuevo commit.

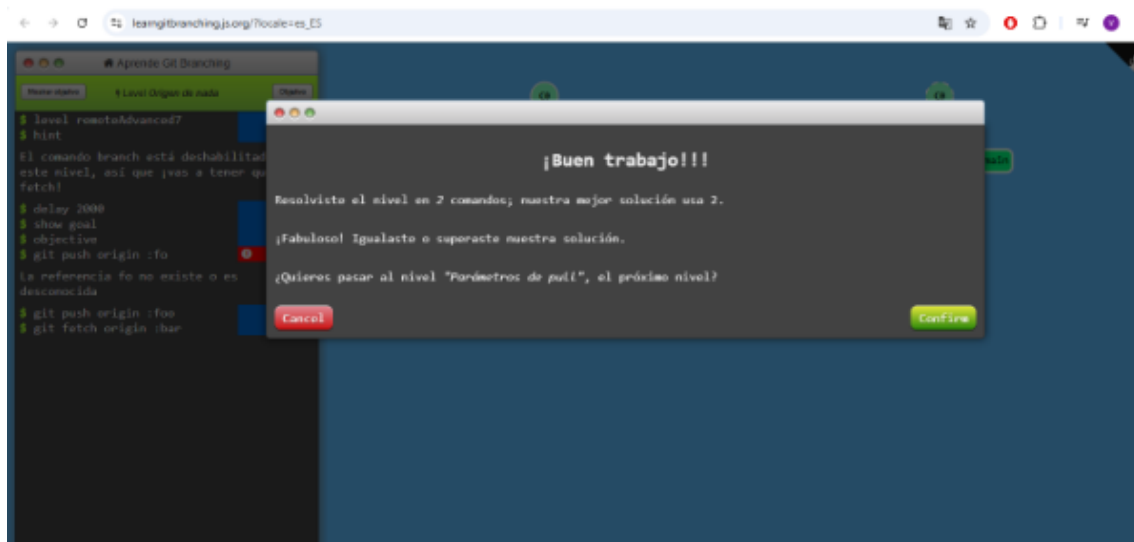




LVL 7

Eliminamos la rama foo del remoto origin y traemos la rama remota bar a la local para actualizarla.





LVL 8

Por último, traemos el commit C3 y C2 al remoto origin y lo fusionamos en el local de foo (el C3), y de side (el C2)

