

**Taller: Aprendiendo Git.**

Julián David Orjuela Parada.

Emmanuel Ortiz.

Algoritmos y programación.

Facultad de Ingeniería, Universidad EAN.

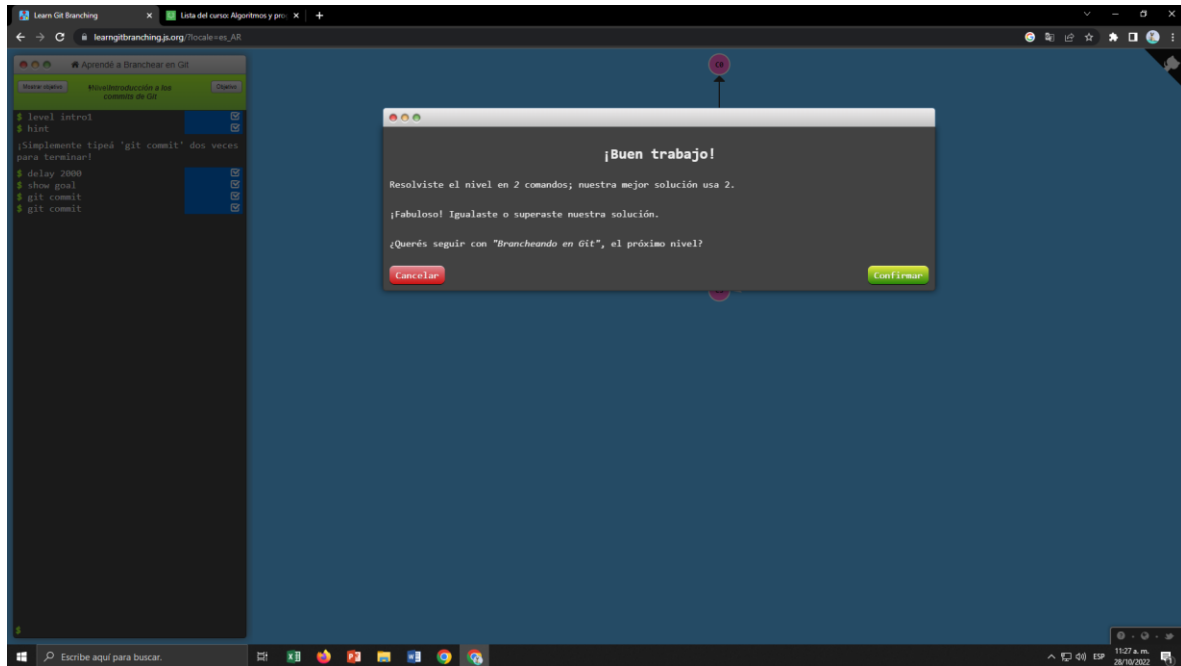
Bogotá D.C.

28 de agosto de 2022.



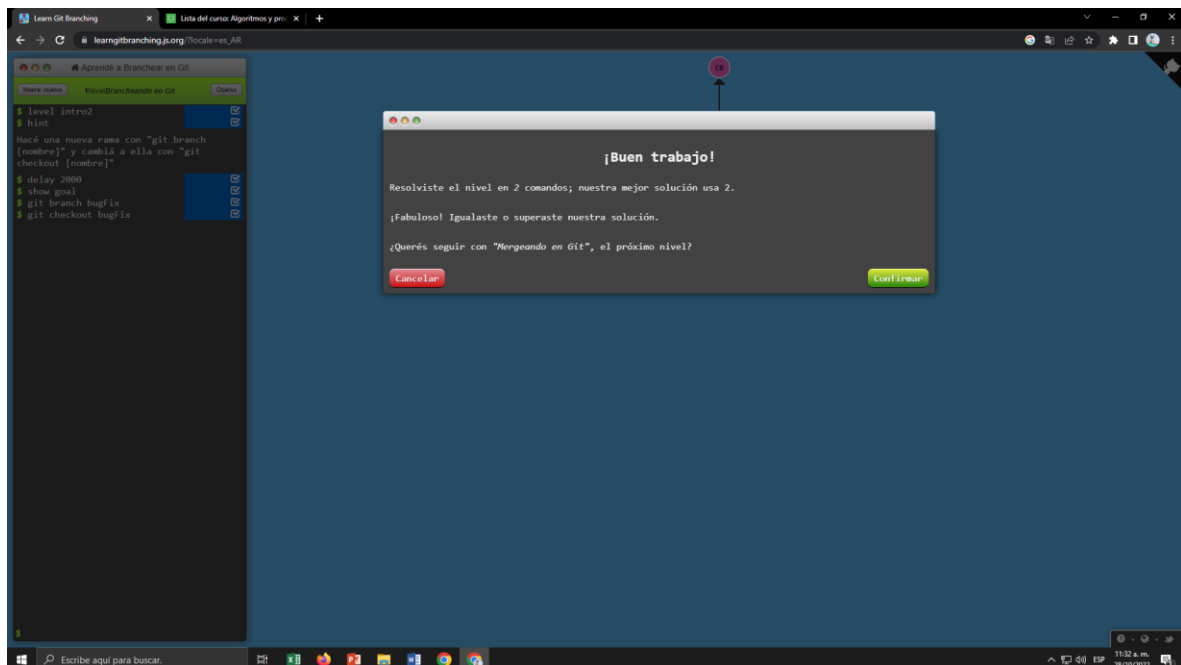
- **Secuencia introductoria:**

## 1) Introducción a los commits de Git:



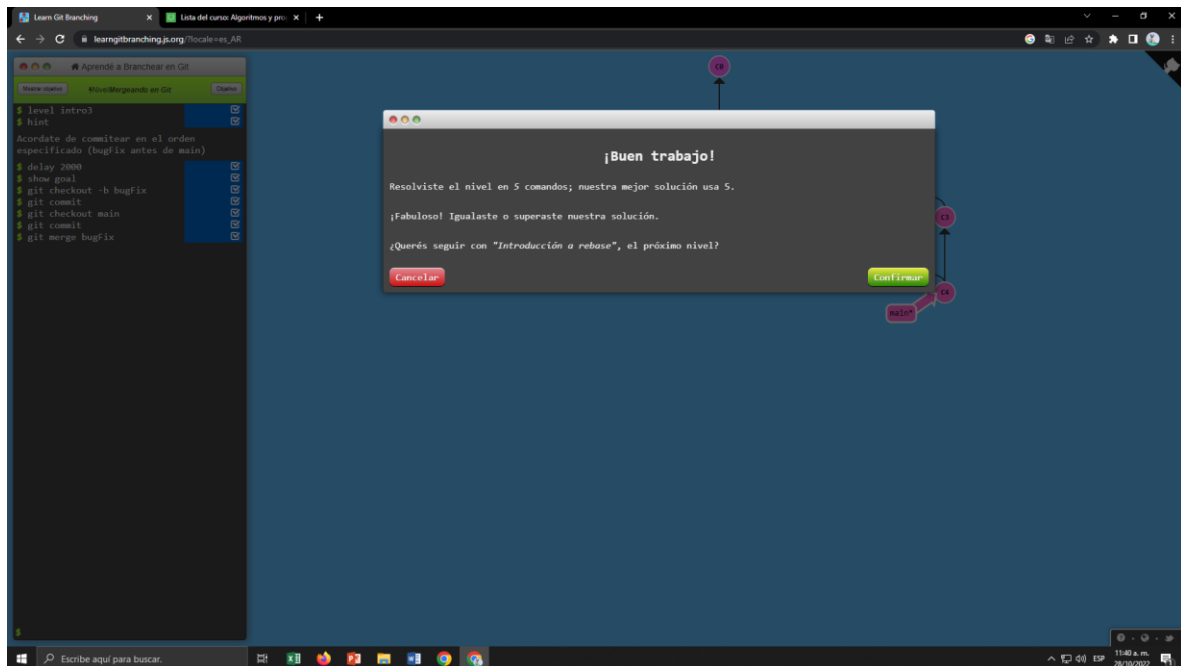
En este caso creamos dos repositorios nuevos del código original, utilizando el comando **git commit** dos veces.

## 2) Ramas en Git:



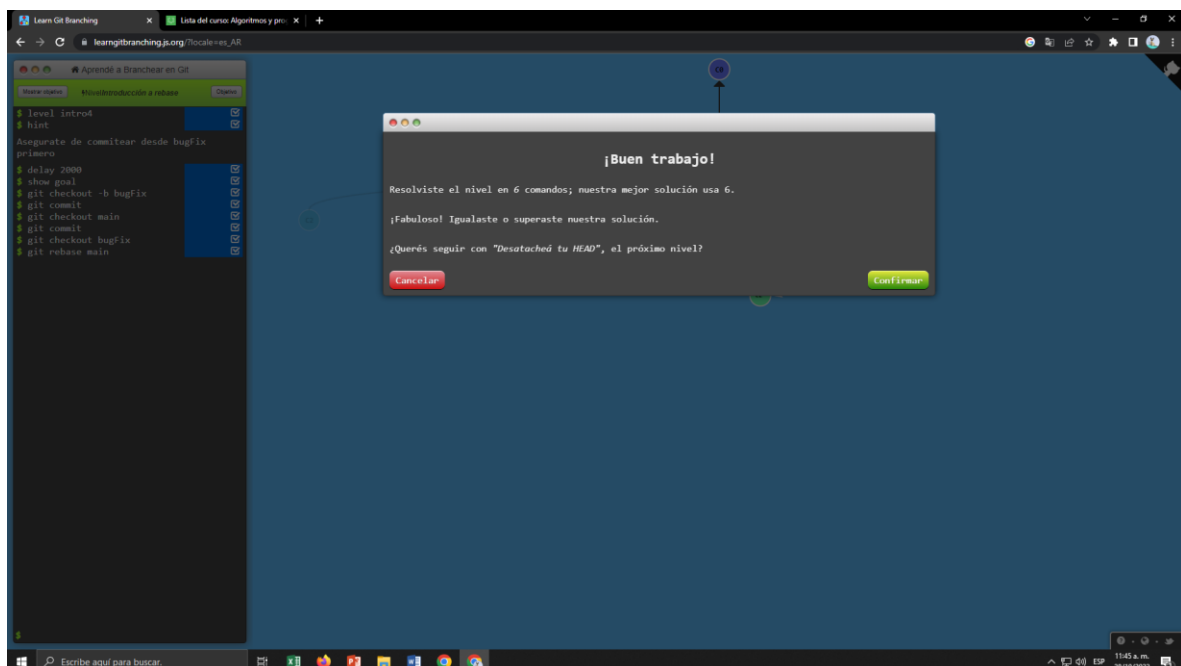
En este caso creamos una rama al primer repositorio denominada bugFix con el comando **git branch** y posteriormente nos posicionamos en ella a través del comando **git checkout**.

### 3) Mergeando ramas:



En este caso creamos una nueva rama con el comando **git branch** denominada bugFix y posteriormente creamos un nuevo repositorio con **git commit** que incluye esta rama, posteriormente nos posicionamos en la rama main y hacemos que esta misma también pase a un nuevo repositorio con **git commit** (Ambas tienen en común el repositorio inicial del cual se crearon como ancestro). Finalmente, con el comando **git merge** hacemos que ambas ramas se fusionen.

### 4) Git Rebase:

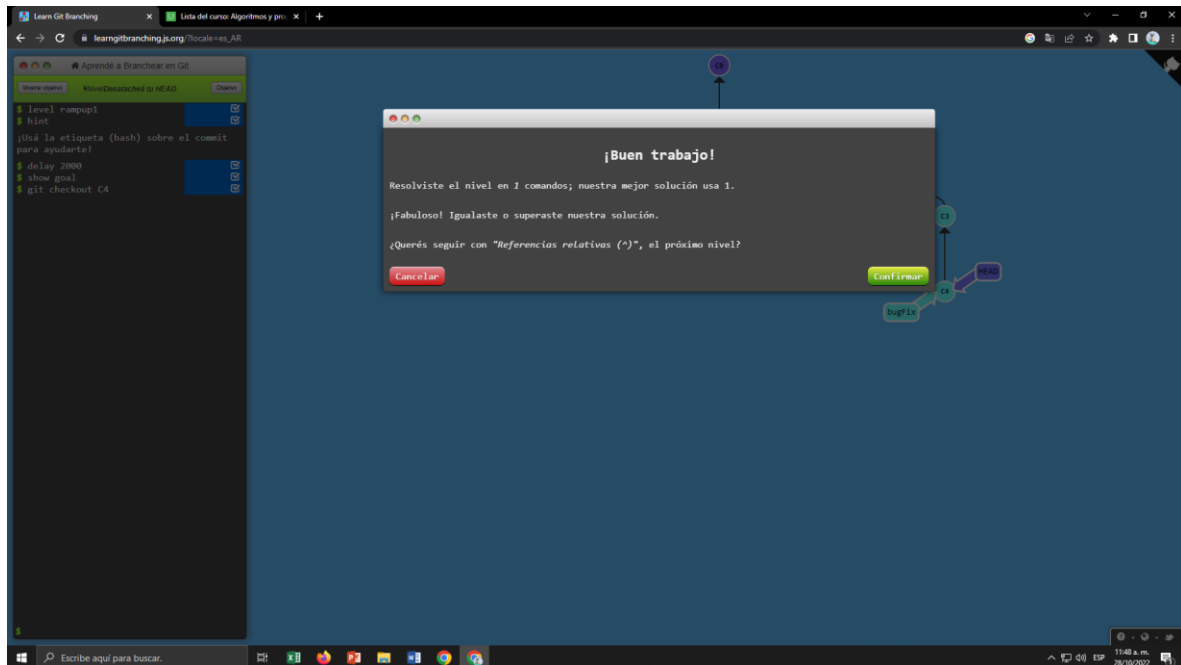


En este caso creamos una rama denominada bugFix y nos posicionamos en ella para trasladarla a un nuevo repositorio y repetimos el mismo procedimiento para la rama main utilizando los

comandos **git checkout** y **git commit**. Teniendo en cuenta que ambos repositorios quedan en paralelo, con el comando **git rebase** hacemos que se cree un repositorio complementario en serie con la rama main y se omita el anterior.

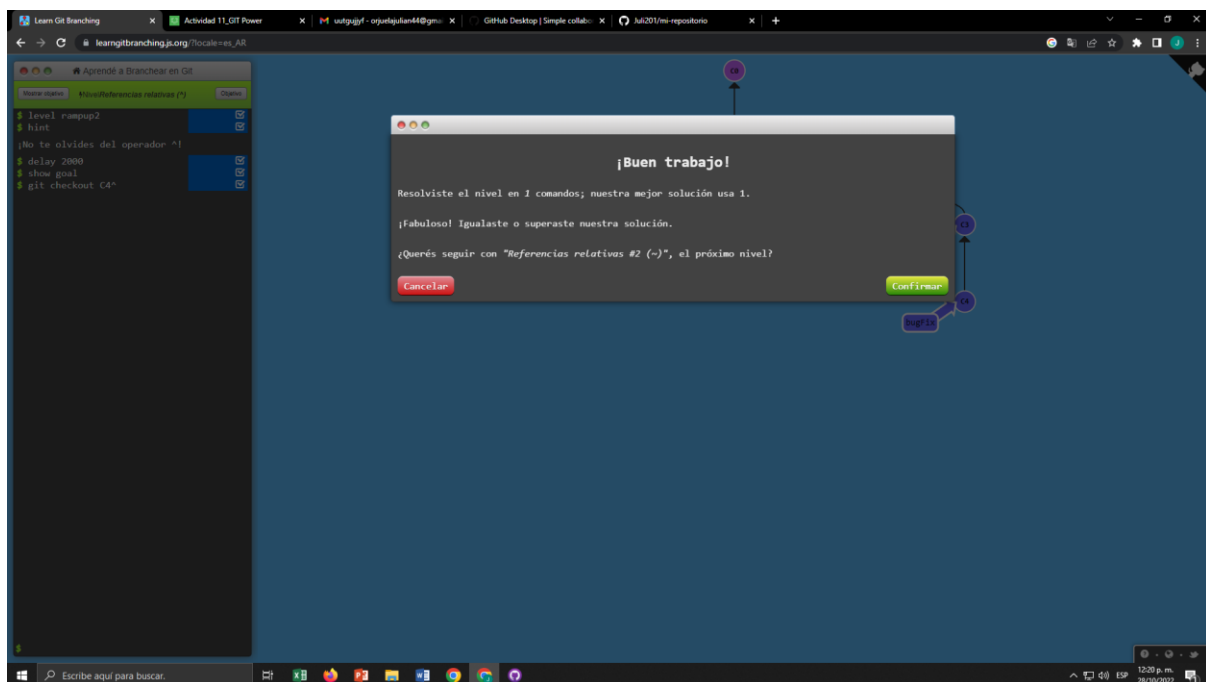
- **Acelerando:**

### 1) Desatachea tu HEAD:



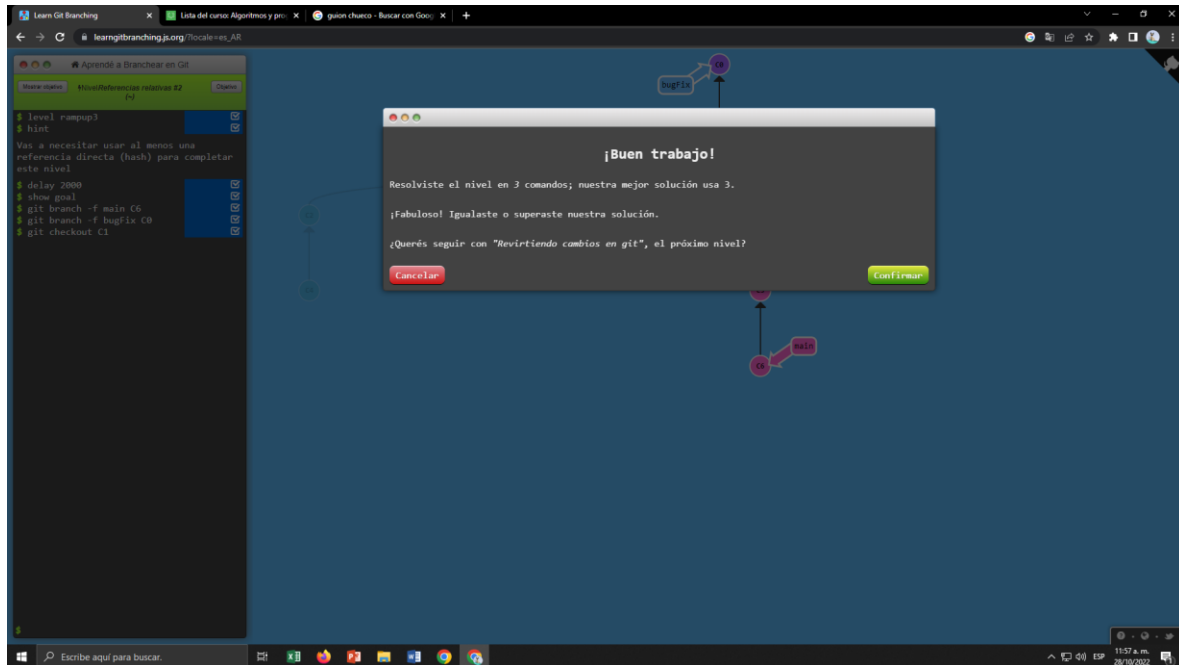
En este caso nos posicionamos en el repositorio C3 y con el comando **git checkout** extraemos el HEAD de este repositorio para que podamos trabajar con el posteriormente.

### 2) Referencias relativas:



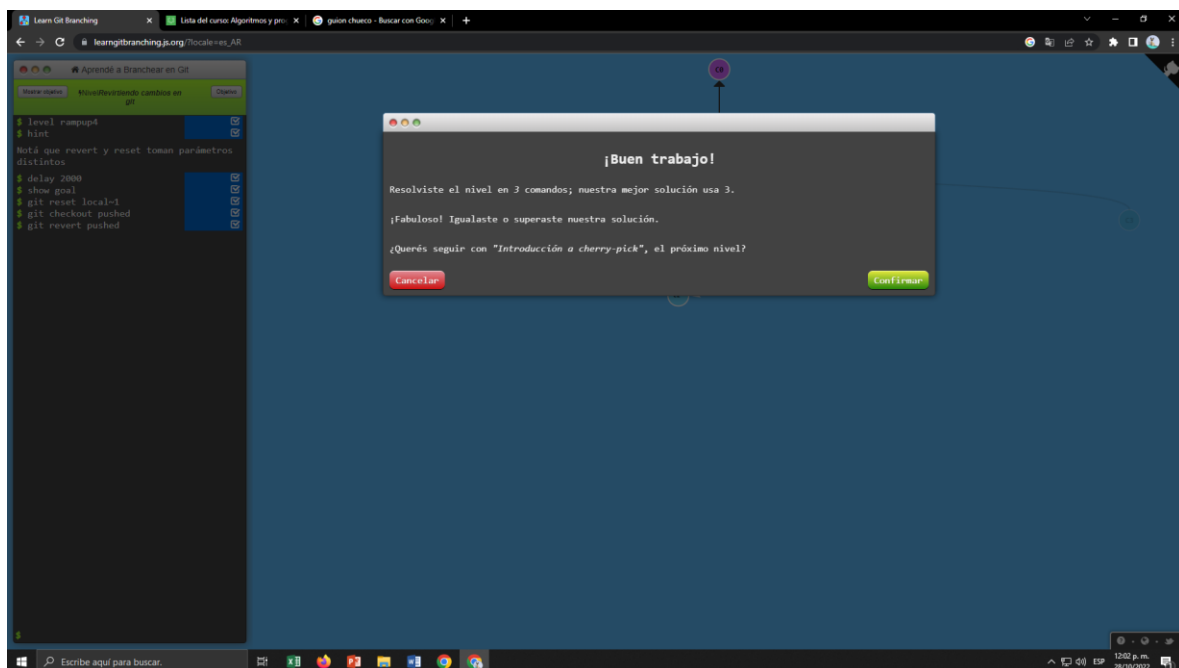
En este caso realizamos el mismo procedimiento anterior, solo que en esta ocasión hacemos que el repositorio C4 utilice como referencia principal al repositorio anterior ubicado por encima de este mismo.

### 3) El operador (~):



En este caso se referencian dos ramas (bugFix y main) en dos repositorios diferentes (C6 y C0) con el comando **git branch** y forzamos la acción con el comando **-f** para que al final el HEAD se posicione en el repositorio C1.

### 4) Revirtiendo cambios en Git:

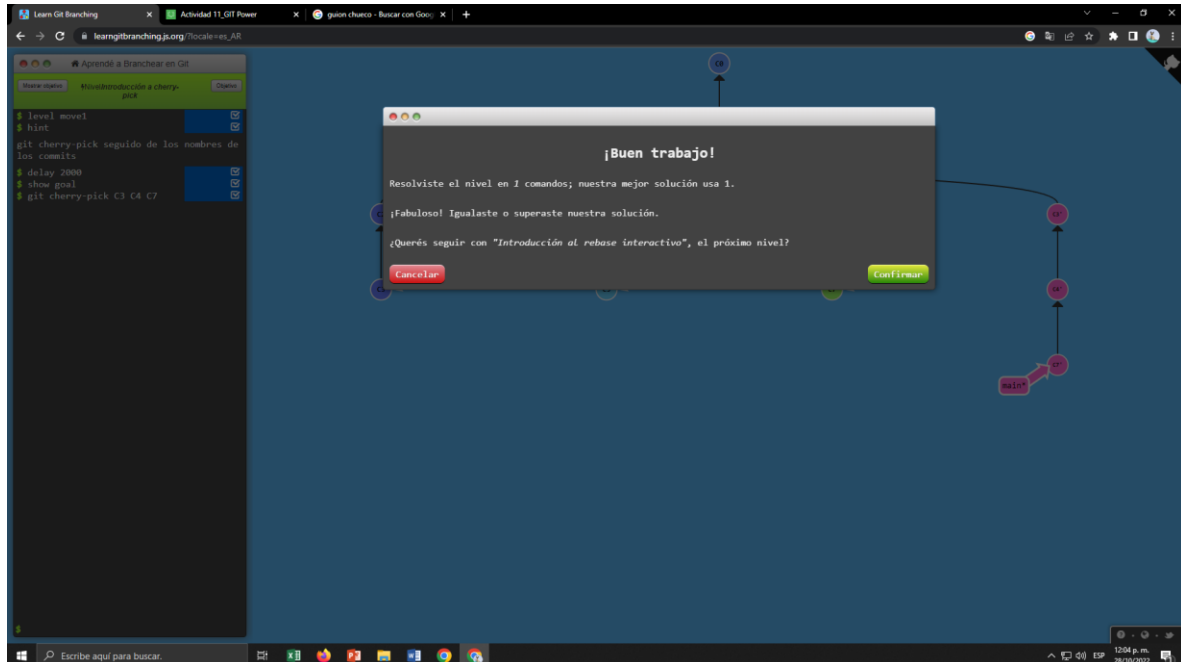


En este caso se revierten los cambios de la rama local con el comando **git reset** una posición hacia arriba en los repositorios, después nos posicionamos en la rama pushed y con el comando

**git revert** hacemos que esta misma haga una actualización general de sus cambios realizados, conservando al mismo tiempo el repositorio general.

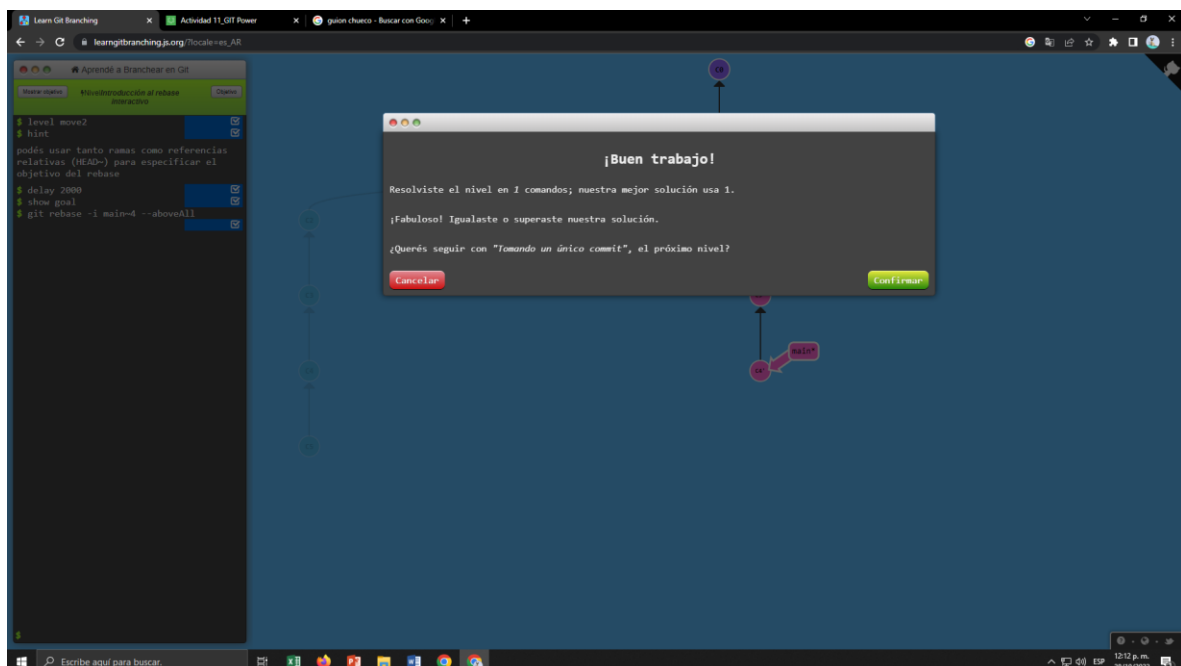
- **Moviendo el trabajo por ahí:**

### 1) Git Cherry-Pick:



En este caso con el comando **git cherry-pick** copiamos los repositorios C3, C4 y C7 en el HEAD de la rama main, de manera que su contenido se refleja ahora desde esta nueva rama al mismo tiempo.

### 2) Git rebase interactivo:



En este caso vamos a interactuar con el comando **git rebase** creando repositorios complementarios a partir de la rama main y alterando su posición (C4 ↔ C5) y omitiendo algunos de estos mismos (C2).