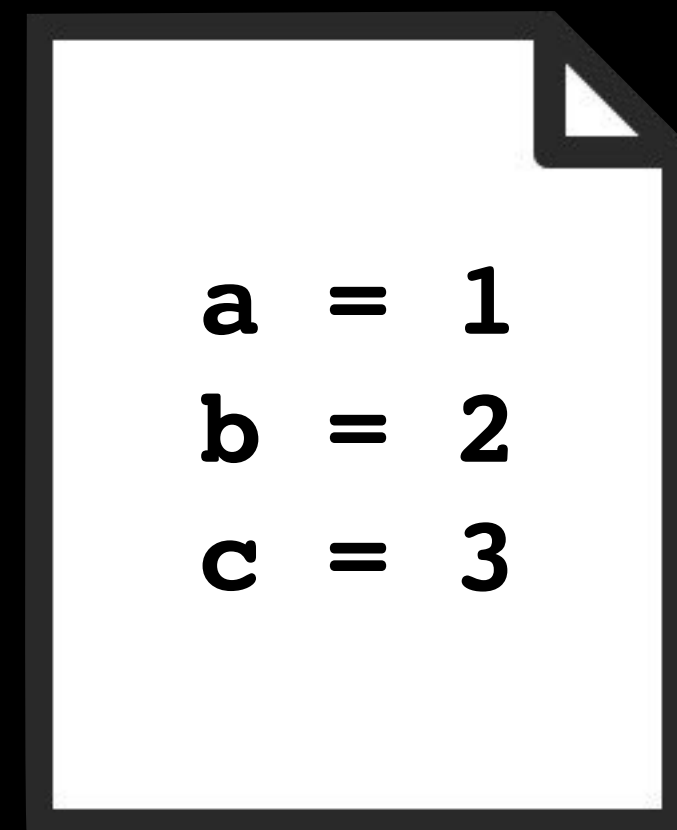


# *Taller Git*

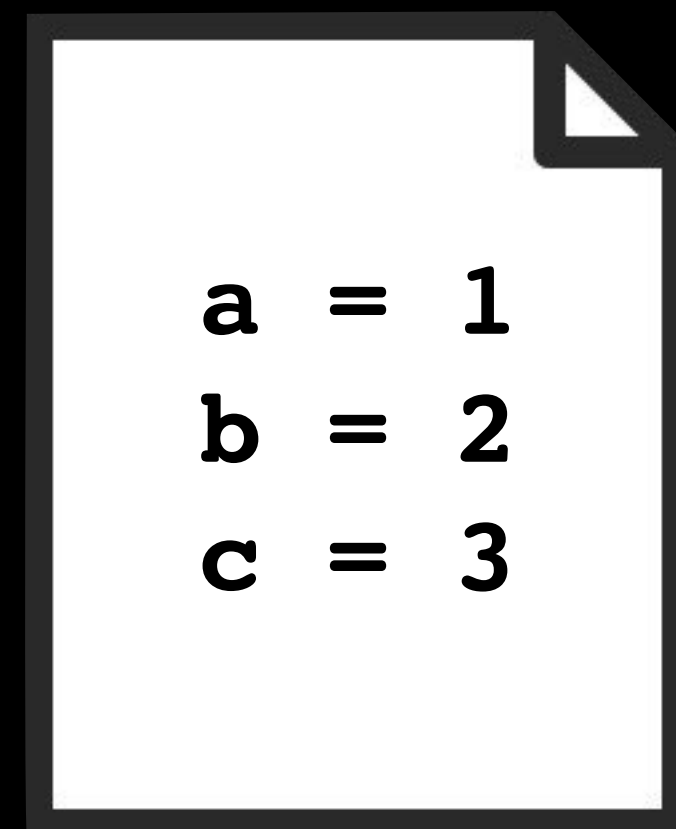
¿Qué es Git?

# Permite dar seguimiento a los cambios en el código



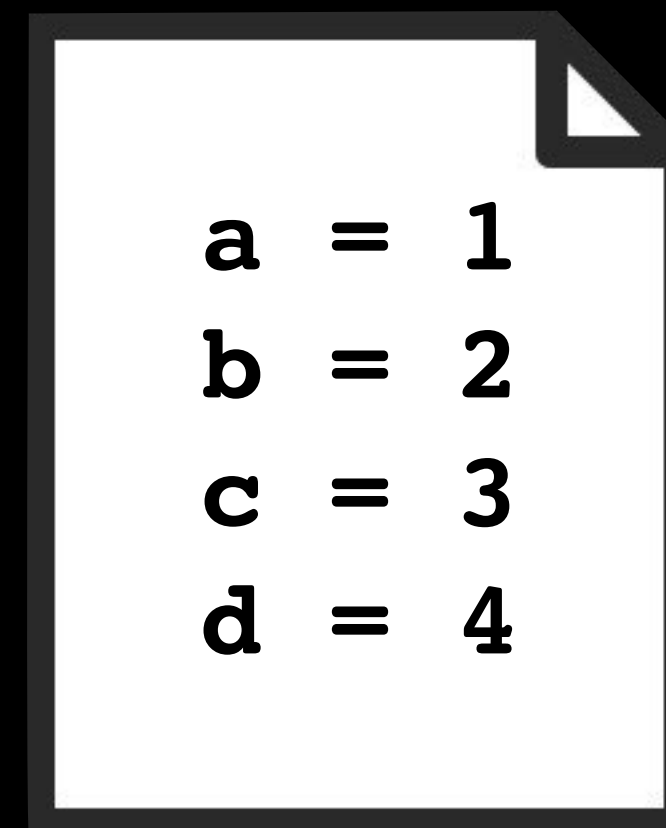
Crear un archivo

# Permite dar seguimiento a los cambios en el código



```
a = 1  
b = 2  
c = 3
```

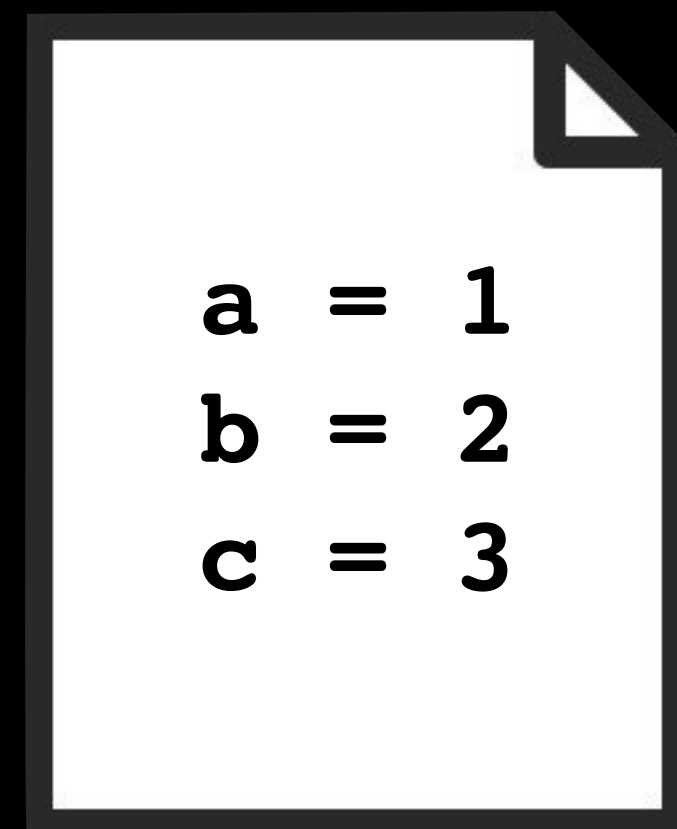
Crear un archivo



```
a = 1  
b = 2  
c = 3  
d = 4
```

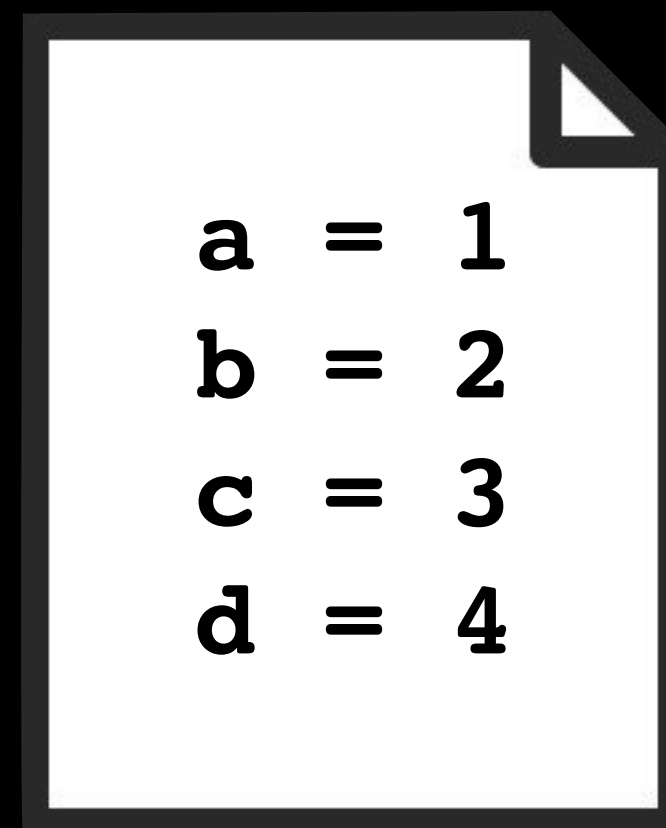
Agregar una línea

# Permite dar seguimiento a los cambios en el código

A white document icon with a folded top-right corner, containing the code:

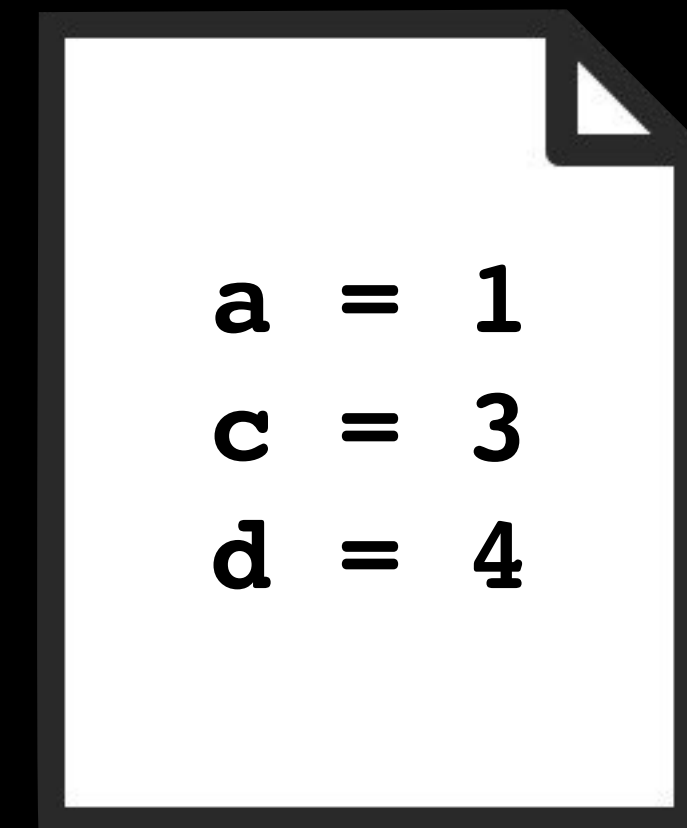
```
a = 1  
b = 2  
c = 3
```

Crear un archivo

A white document icon with a folded top-right corner, containing the code:

```
a = 1  
b = 2  
c = 3  
d = 4
```

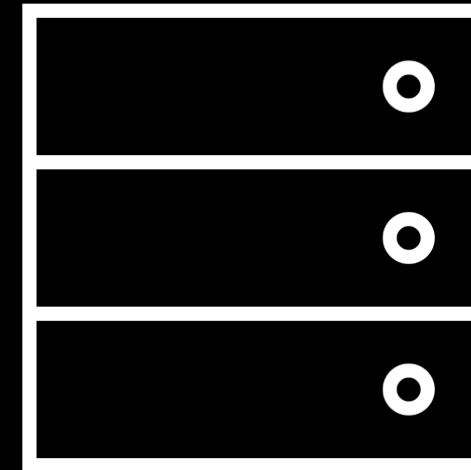
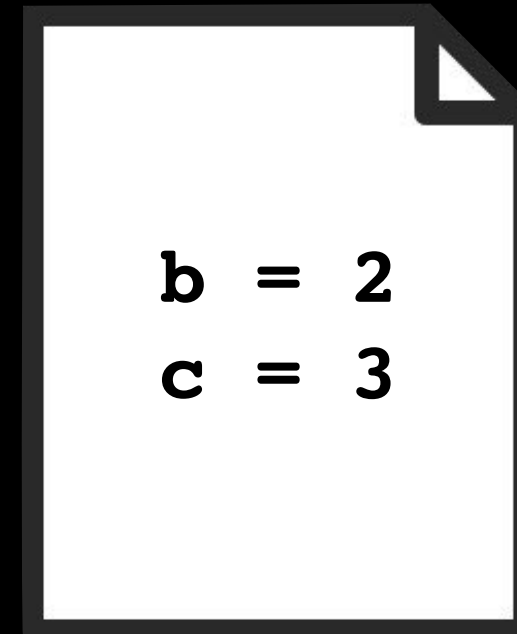
Agregar una línea

A white document icon with a folded top-right corner, containing the code:

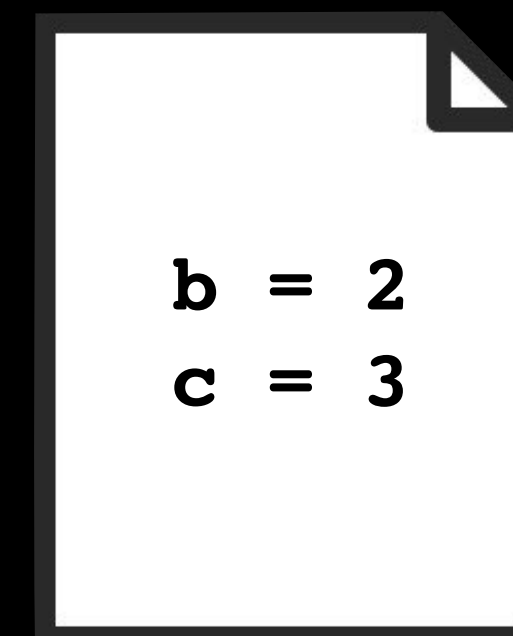
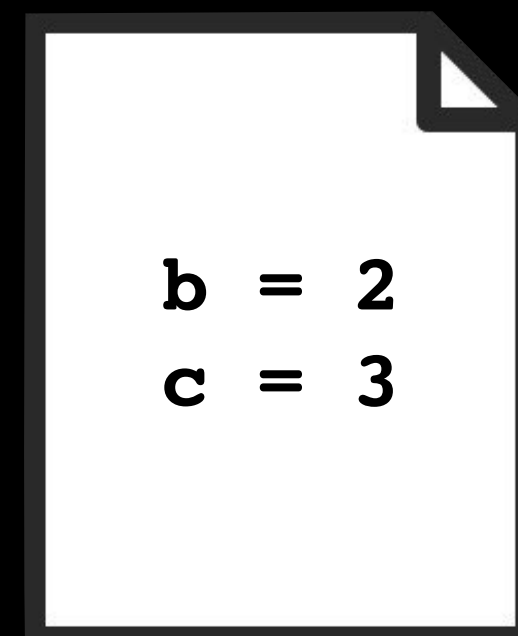
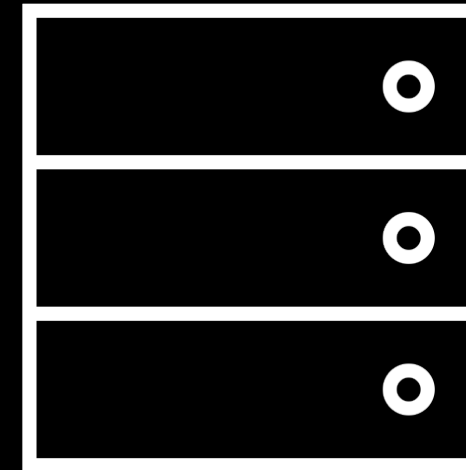
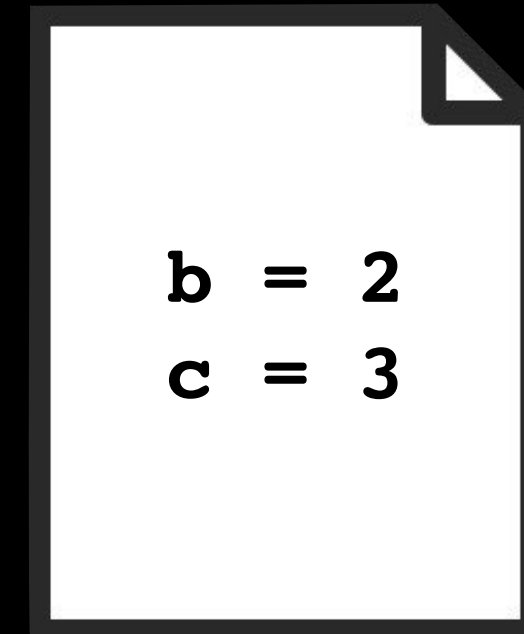
```
a = 1  
c = 3  
d = 4
```

Quitar una línea

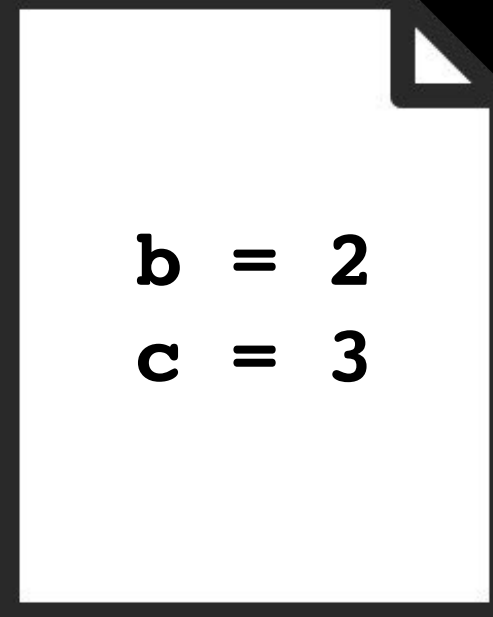
# Sincronizar código entre varias personas



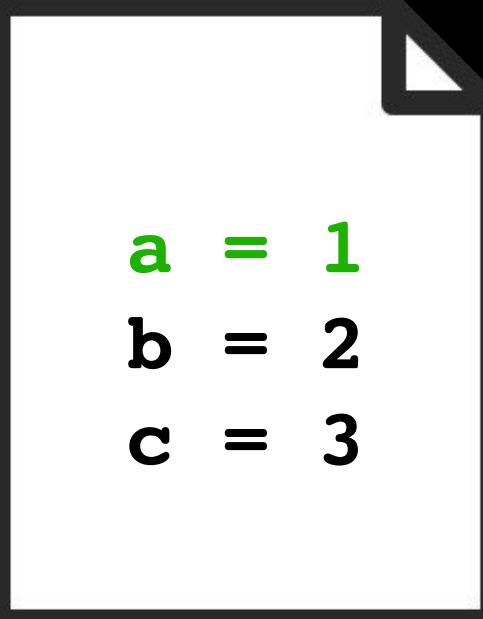
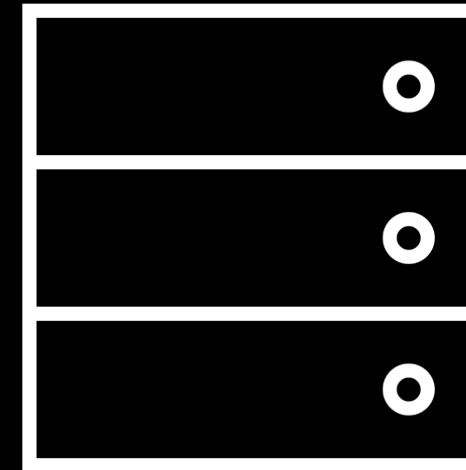
# Sincronizar código entre varias personas



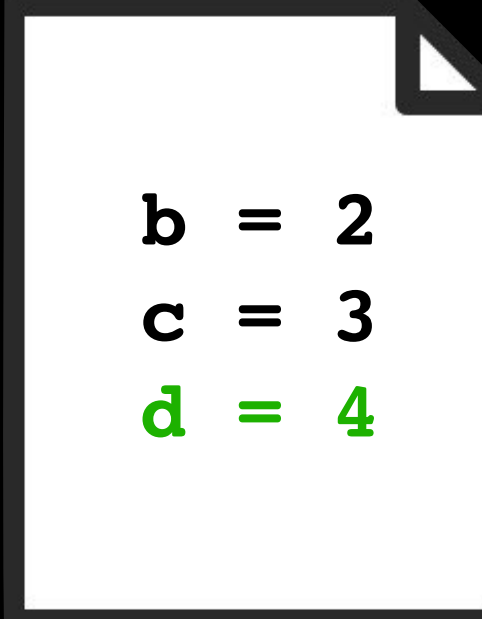
# Sincronizar código entre varias personas



```
b = 2  
c = 3
```



```
a = 1  
b = 2  
c = 3
```

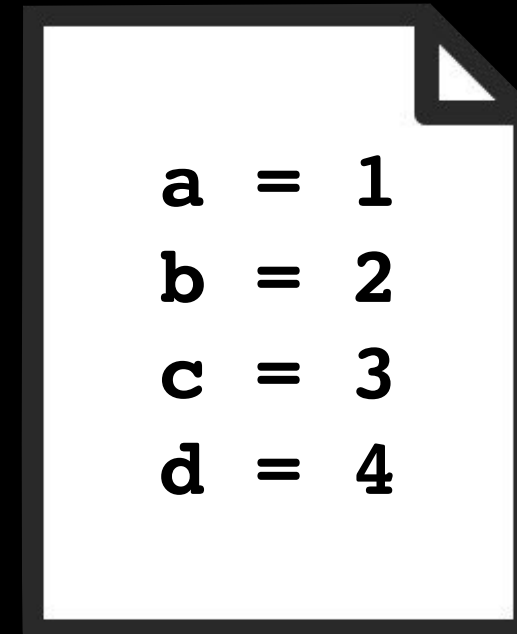


```
b = 2  
c = 3  
d = 4
```

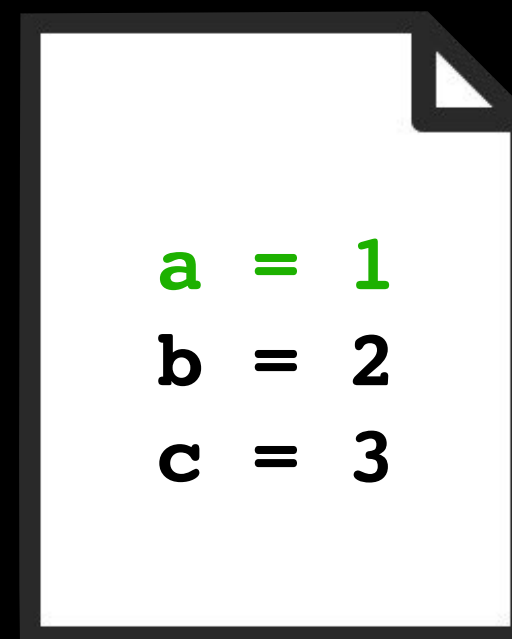
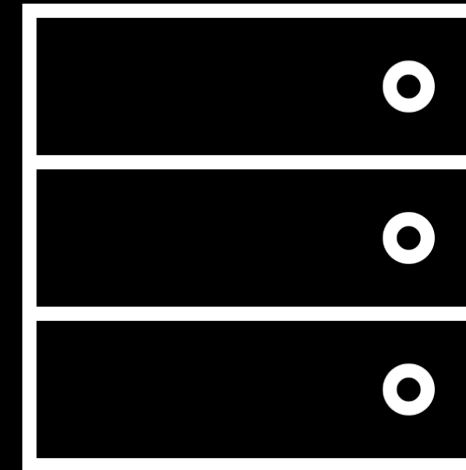




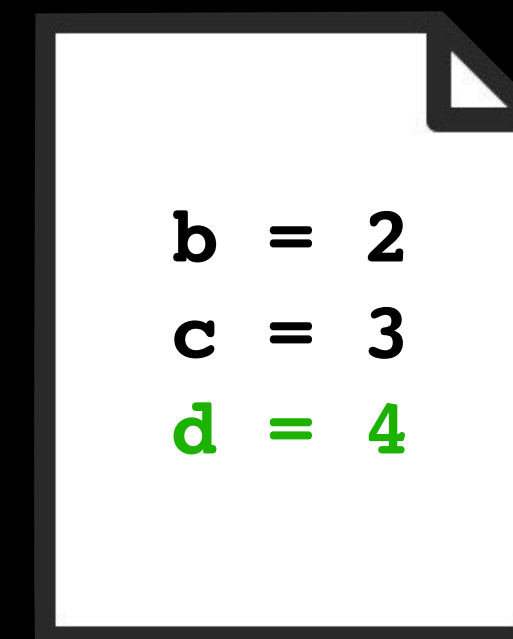
# Sincronizar código entre varias personas



```
a = 1  
b = 2  
c = 3  
d = 4
```



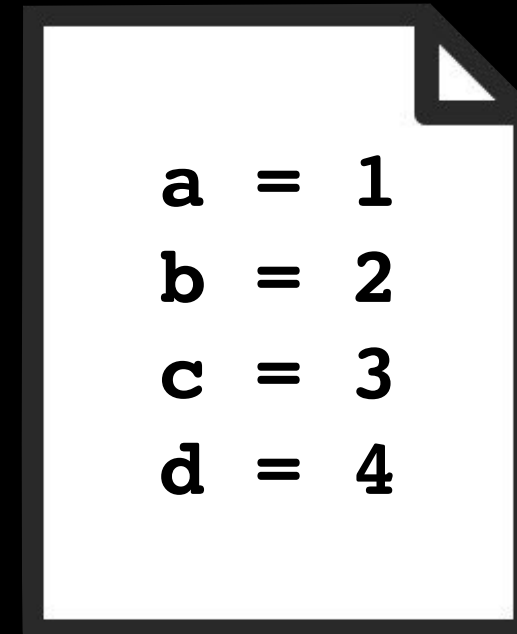
```
a = 1  
b = 2  
c = 3
```



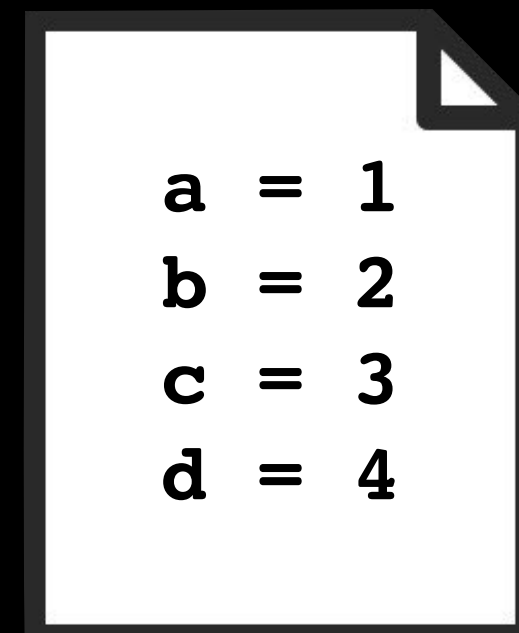
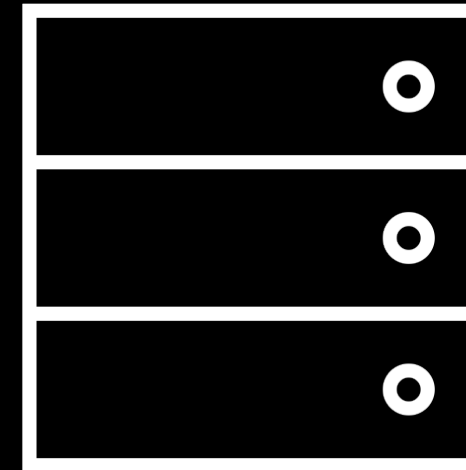
```
b = 2  
c = 3  
d = 4
```



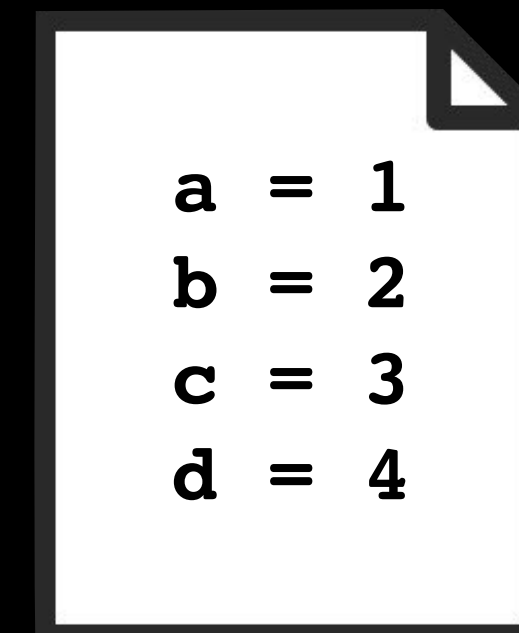
# Sincronizar código entre varias personas

A document icon with a folded top-right corner, containing code.

```
a = 1  
b = 2  
c = 3  
d = 4
```

A document icon with a folded top-right corner, containing code.

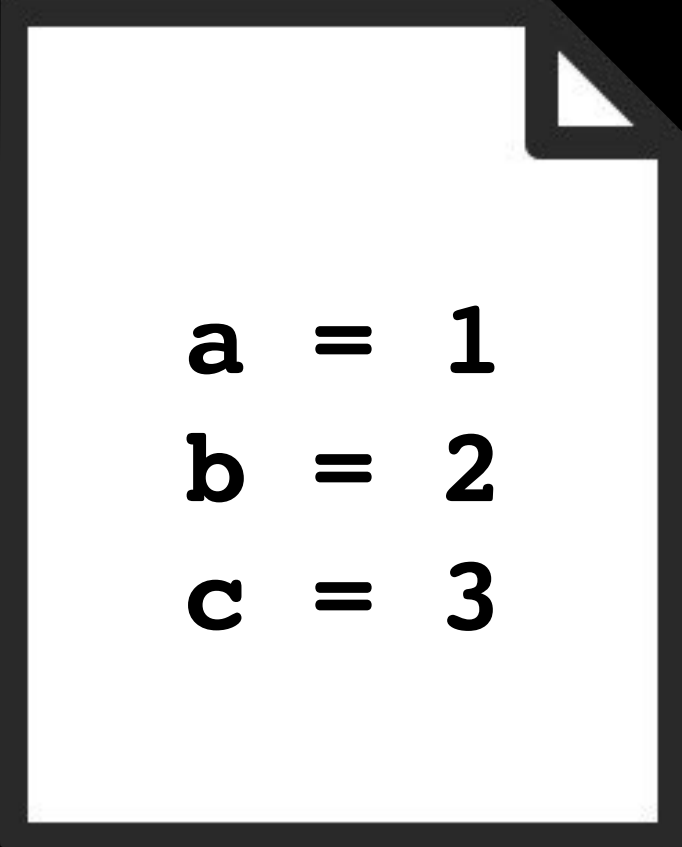
```
a = 1  
b = 2  
c = 3  
d = 4
```

A document icon with a folded top-right corner, containing code.

```
a = 1  
b = 2  
c = 3  
d = 4
```

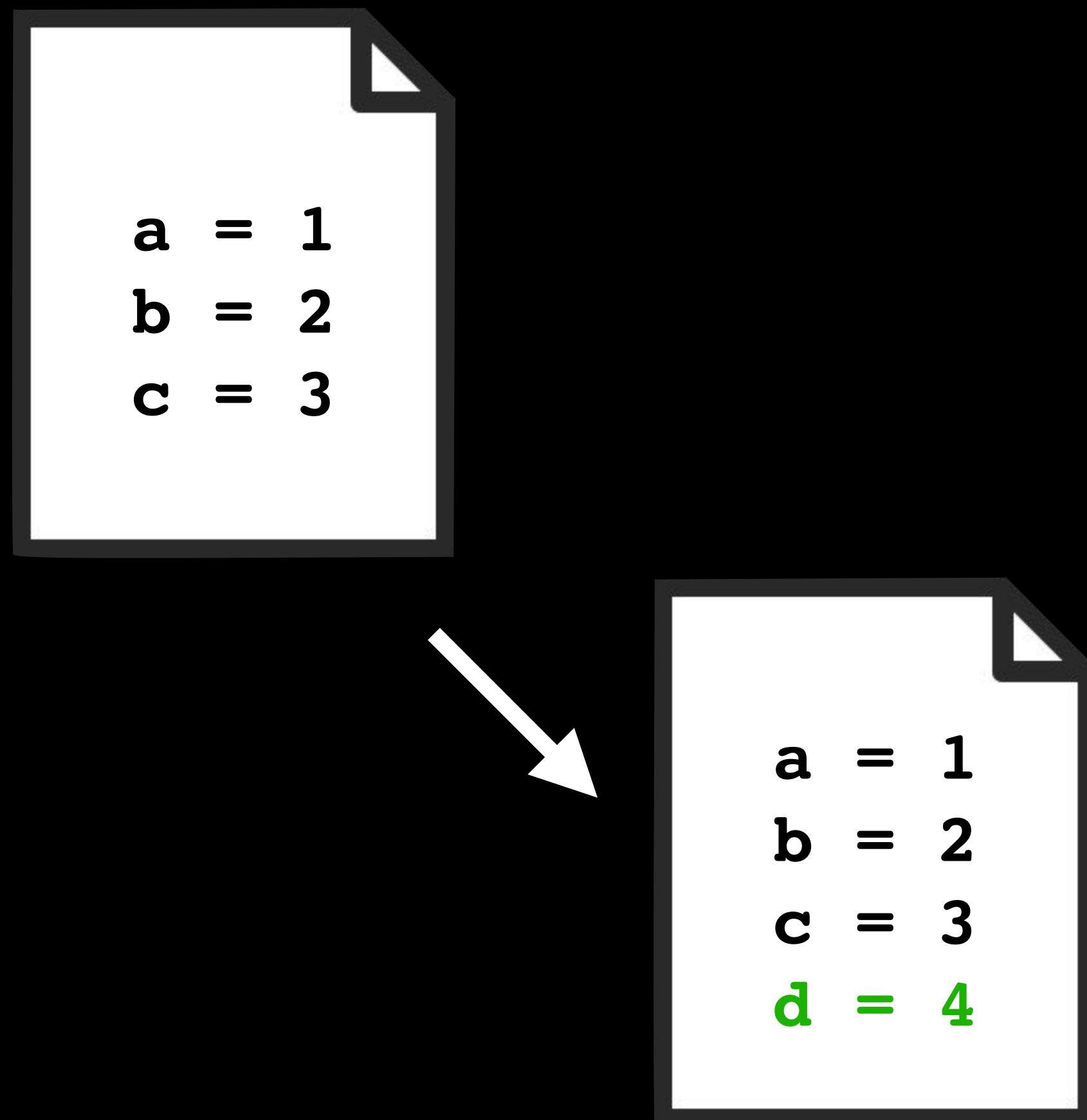


# Hacer pruebas al código sin perder el original

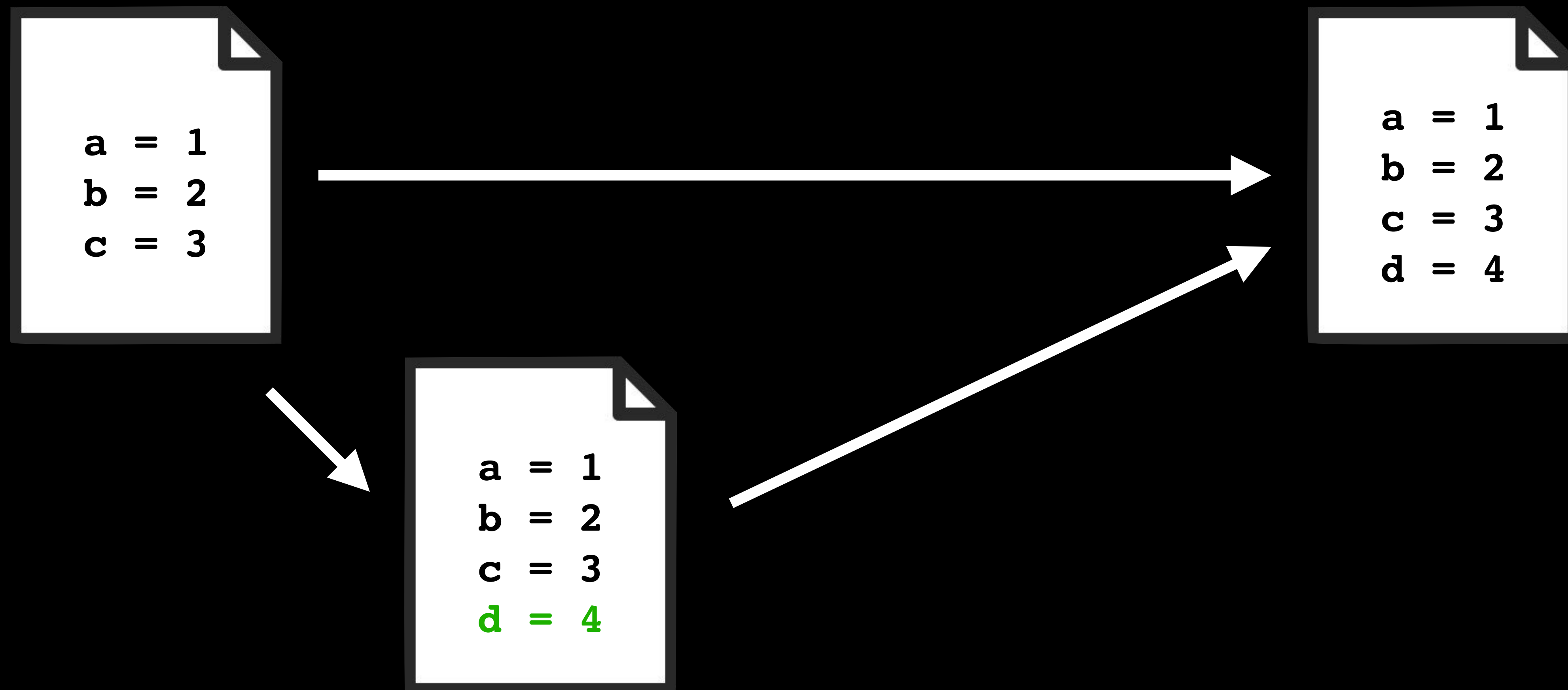


```
a = 1  
b = 2  
c = 3
```

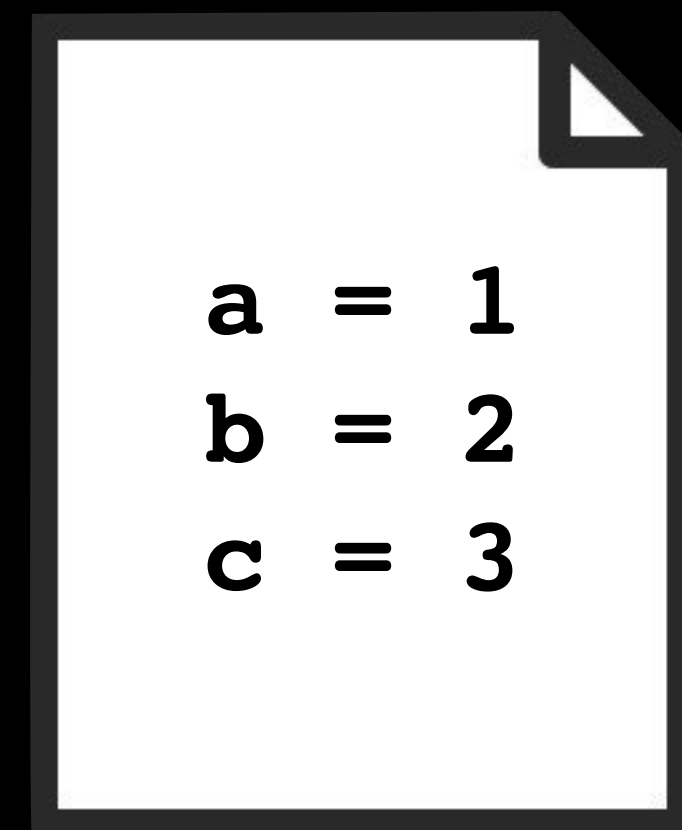
# Hacer pruebas al código sin perder el original



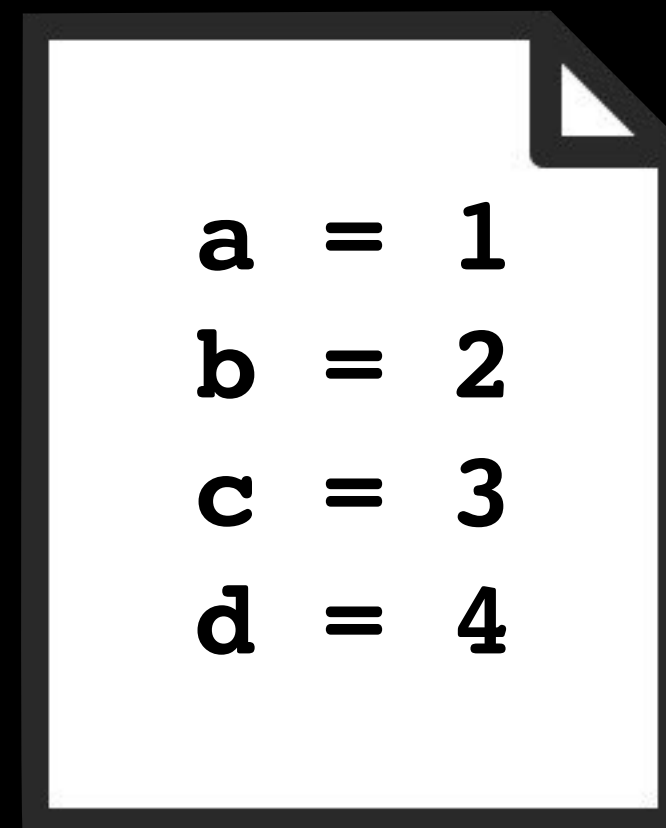
# Hacer pruebas al código sin perder el original



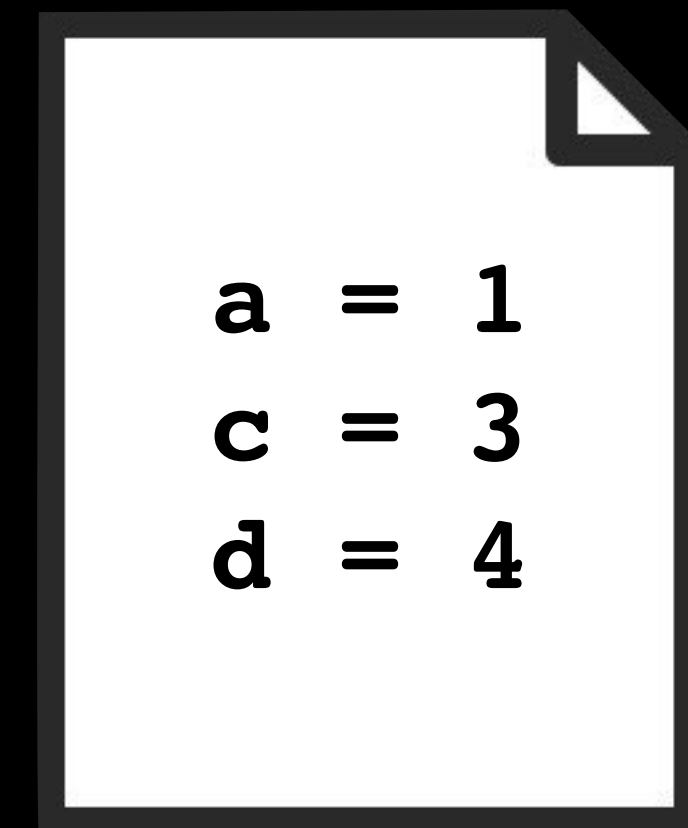
# Regresar a versiones anteriores del código



Crear un archivo



Agregar una línea



Quitar una línea

# ¿Qué es Git?

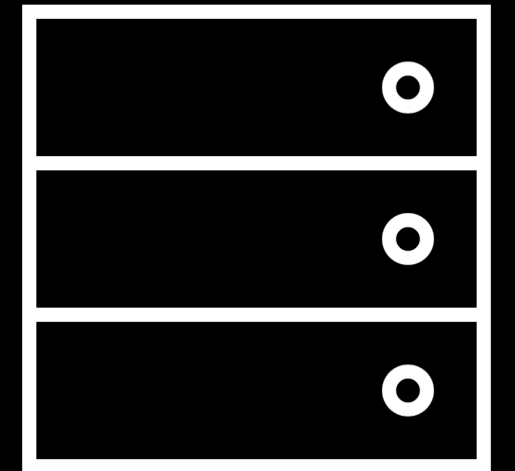
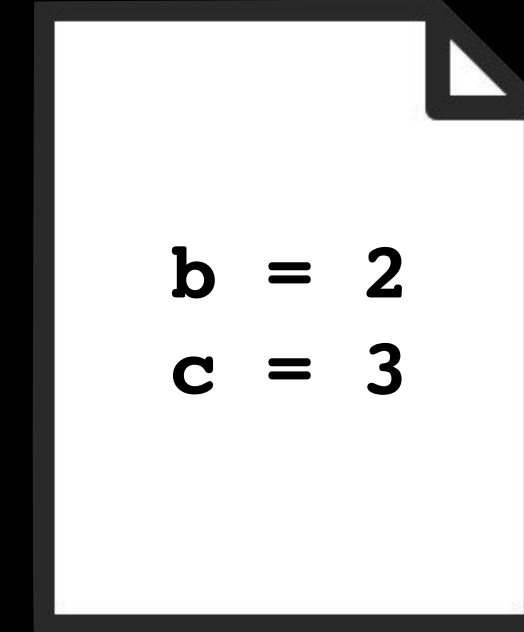
- Permite dar seguimiento a los cambios en el código
- Sincroniza el código entre varias personas
- Realizar pruebas al código sin perder el original
- Regresar a las versiones anteriores del código

git clone



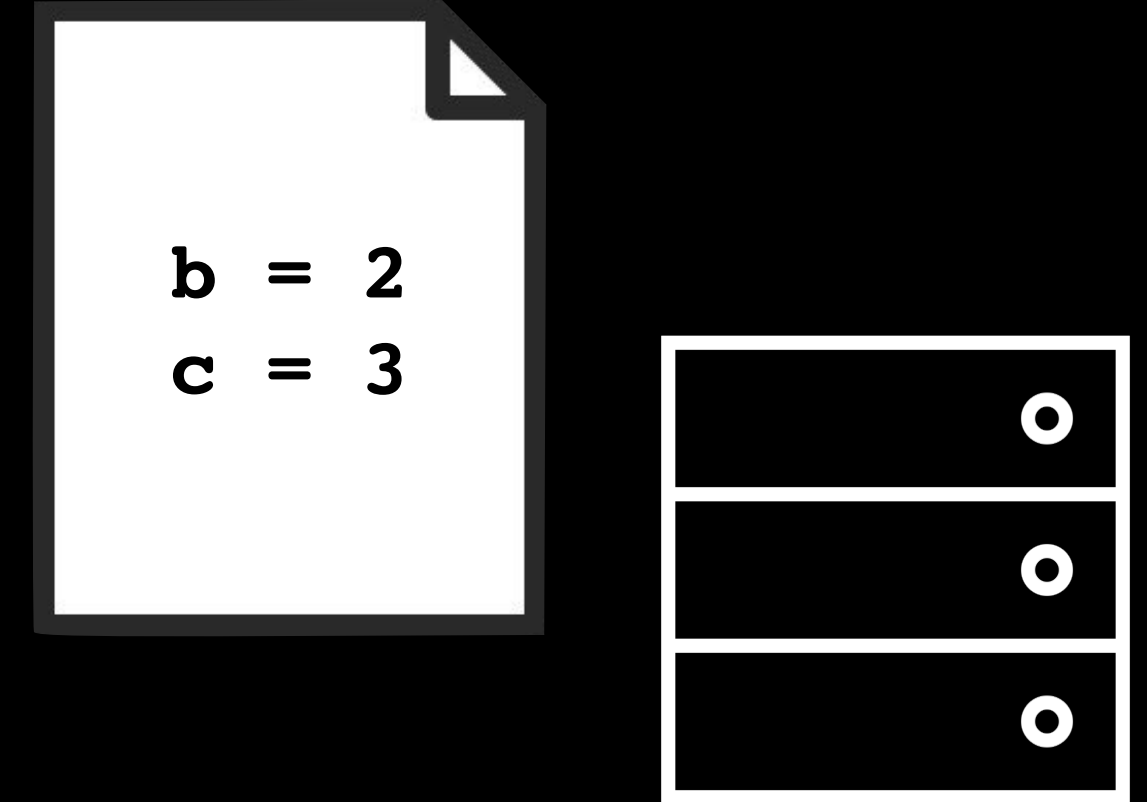
# git clone <url>

- Hace una copia del repositorio
- Guarda el repositorio localmente
- un “fork” crea tu propia copia del repositorio de alguien mas.



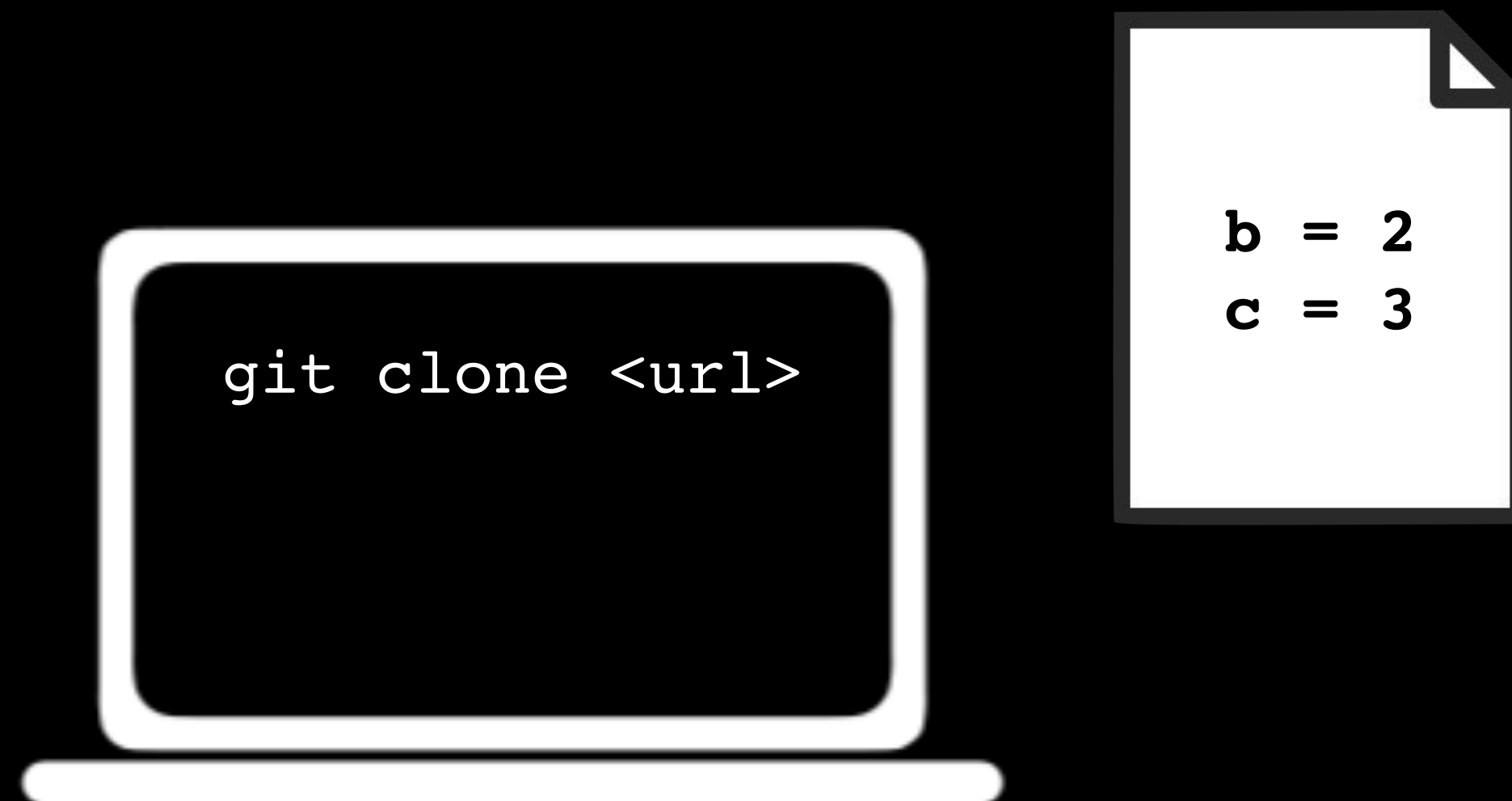
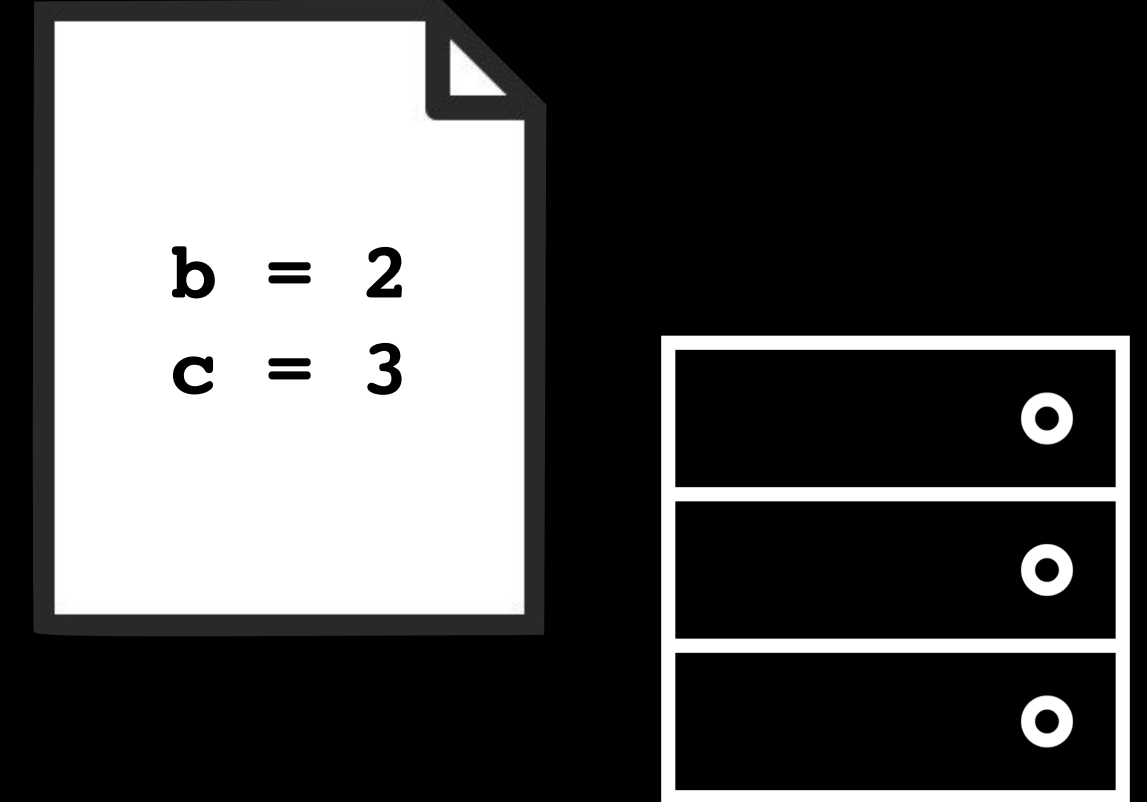
# git clone <url>

- Hace una copia del repositorio
- Guarda el repositorio localmente
- un “fork” crea tu propia copia del repositorio de alguien mas.



# git clone <url>

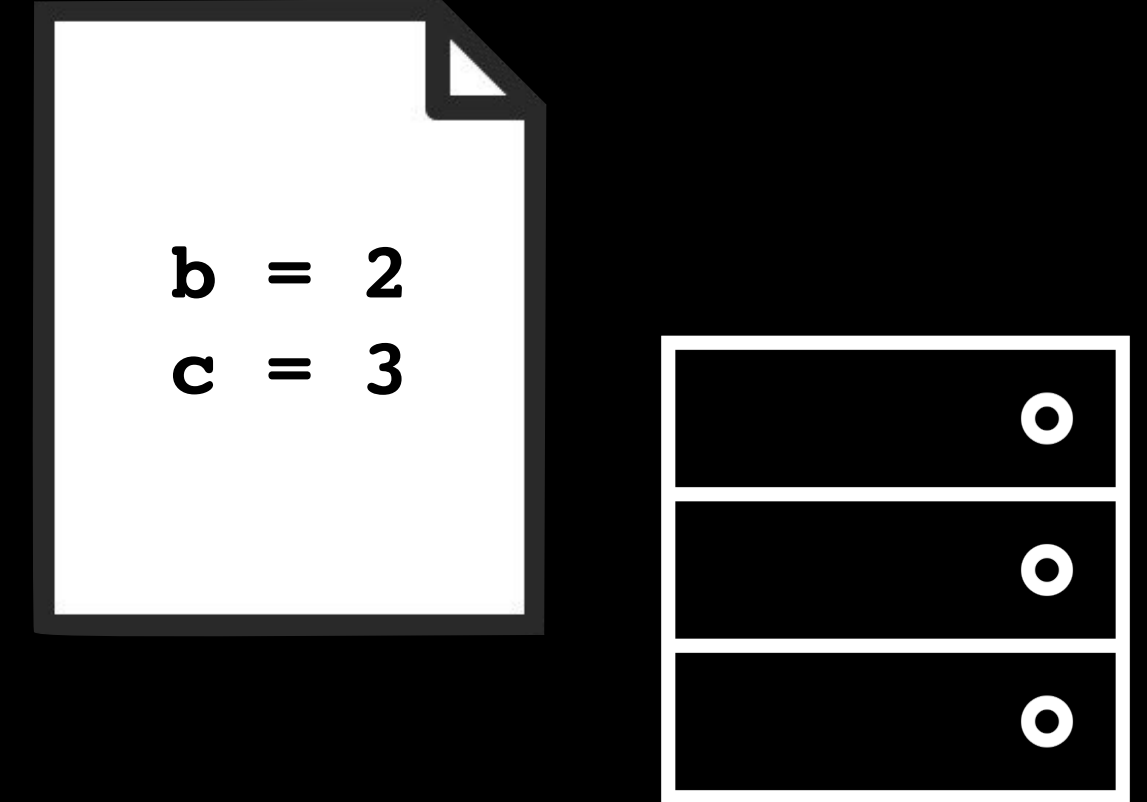
- Hace una copia del repositorio
- Guarda el repositorio localmente
- un “fork” crea tu propia copia del repositorio de alguien mas.



```
git add
```

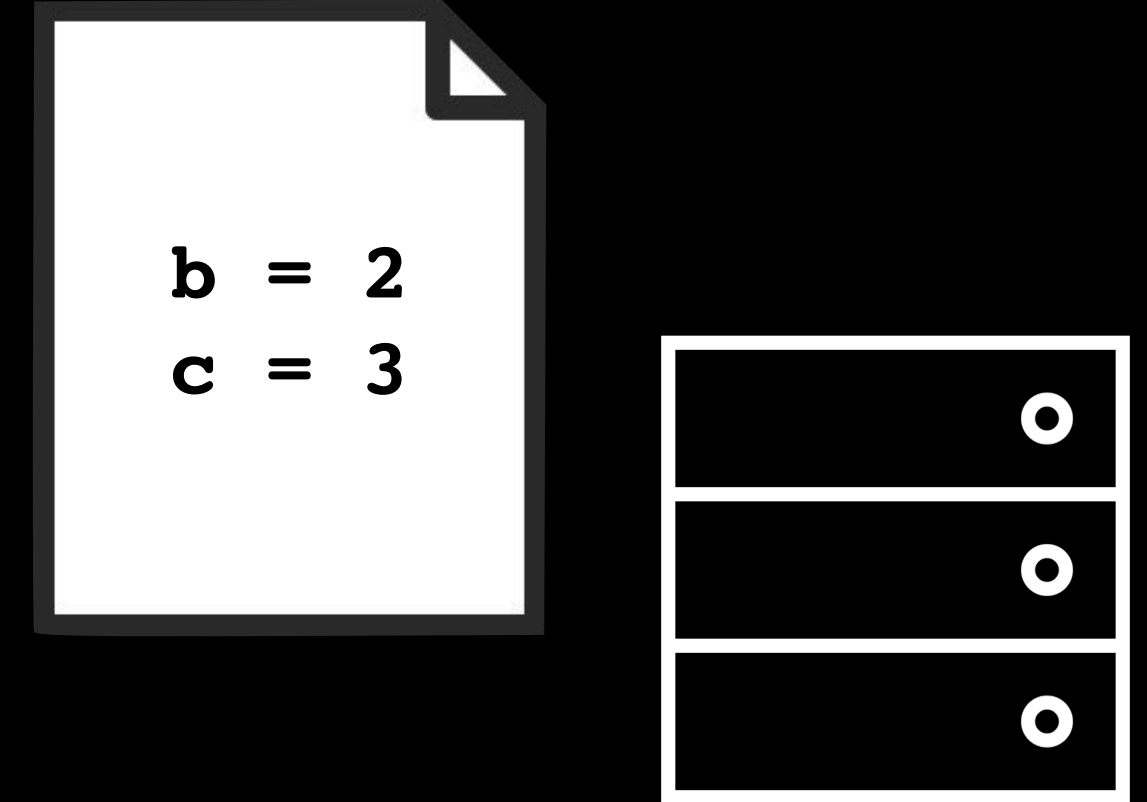
# git add <Archivo>

- Agrega el archivo al área de “staging”
- Incluye el archivo en la próxima revisión del repositorio
- **git add \*** incluye todos los archivos modificados



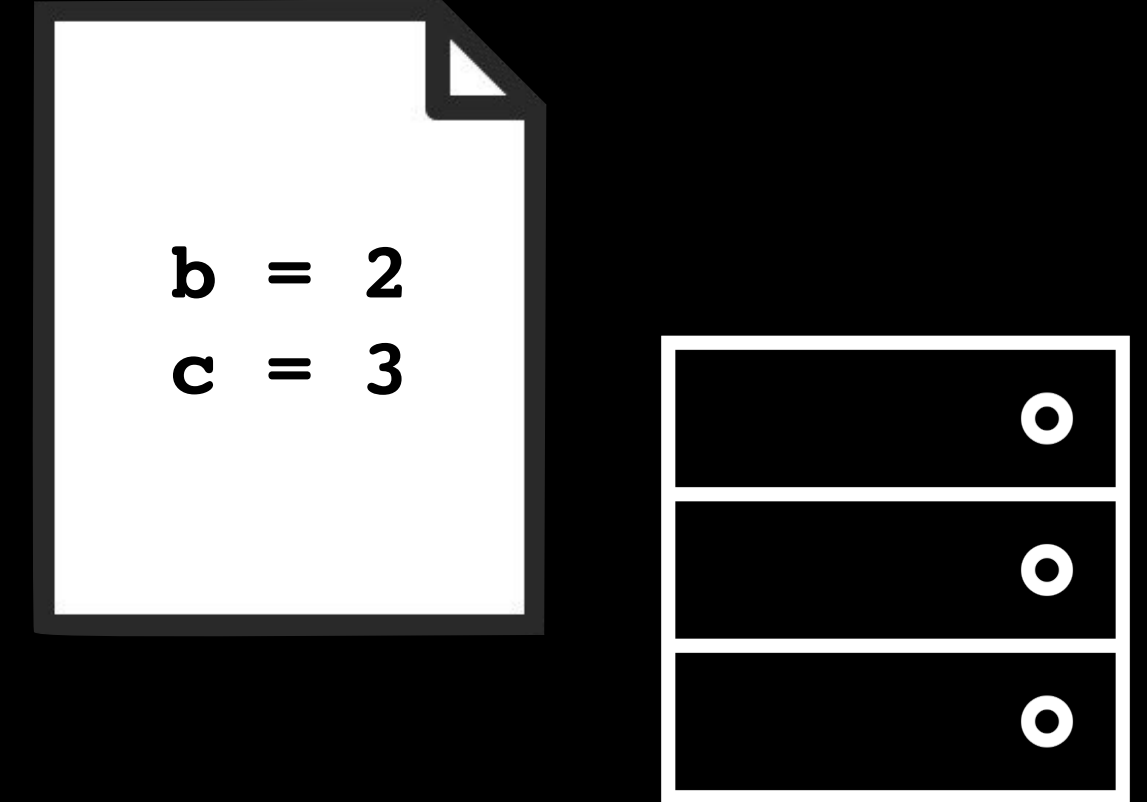
# git add <Archivo>

- Agrega el archivo al área de “staging”
- Incluye el archivo en la próxima revisión del repositorio
- **git add \*** incluye todos los archivos modificados



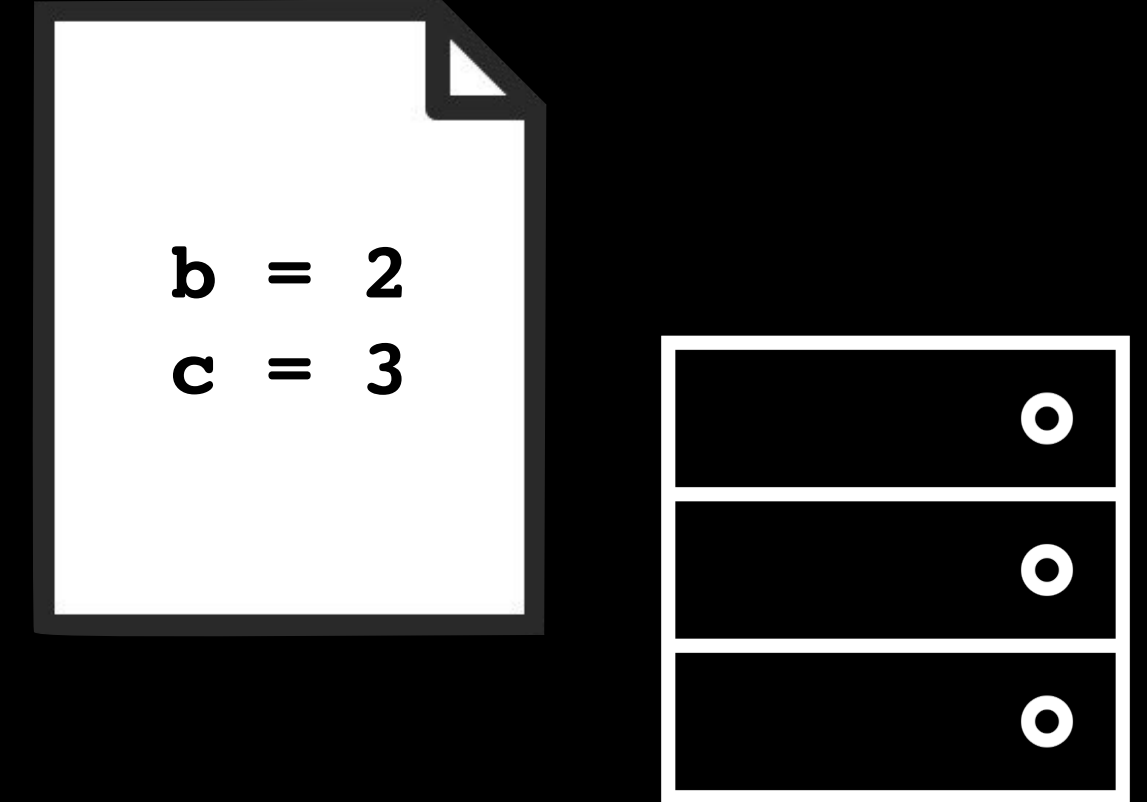
# git add <Archivo>

- Agrega el archivo al área de “staging”
- Incluye el archivo en la próxima revisión del repositorio
- **git add \*** incluye todos los archivos modificados



# git add <Archivo>

- Agrega el archivo al área de “staging”
- Incluye el archivo en la próxima revisión del repositorio
- **git add \*** incluye todos los archivos modificados



**Changes to be committed:**

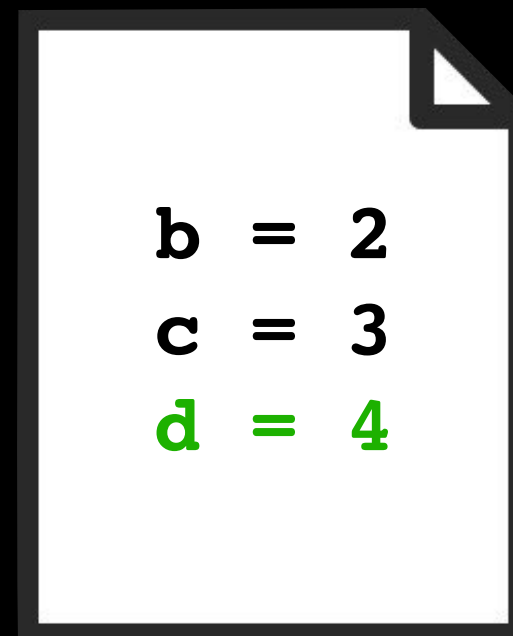
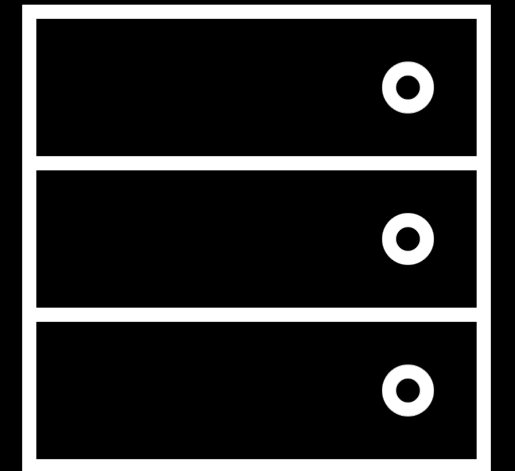
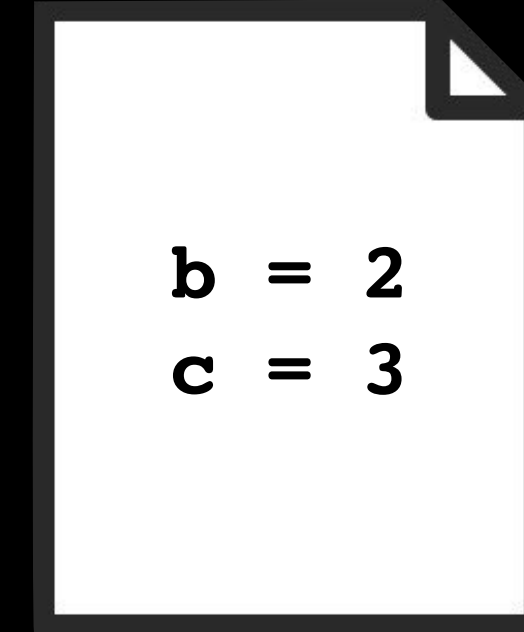
**modified: vars.py**



git commit

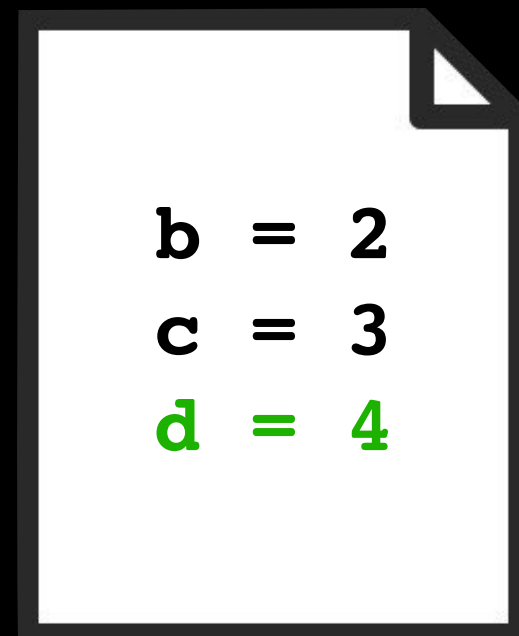
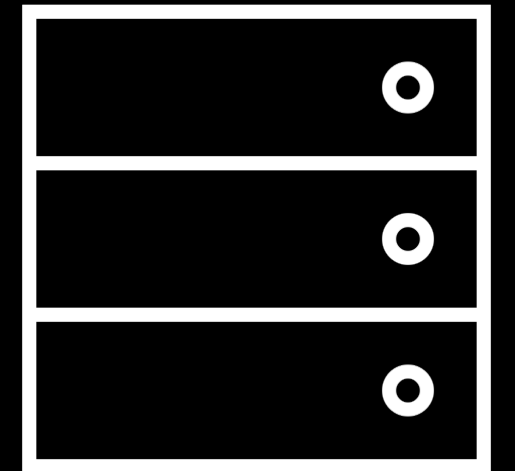
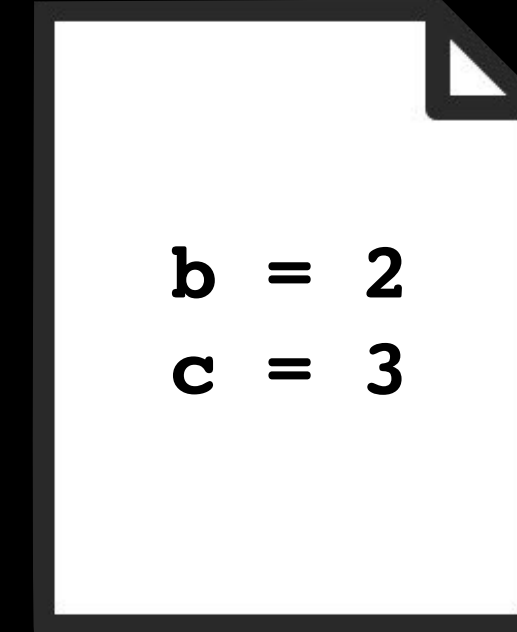
# git commit -m "mensaje"

- Guarda los cambios realizados al repositorio como una nueva versión
- Guarda un mensaje
- **git commit -am "mensaje"** agrega y guarda al mismo tiempo



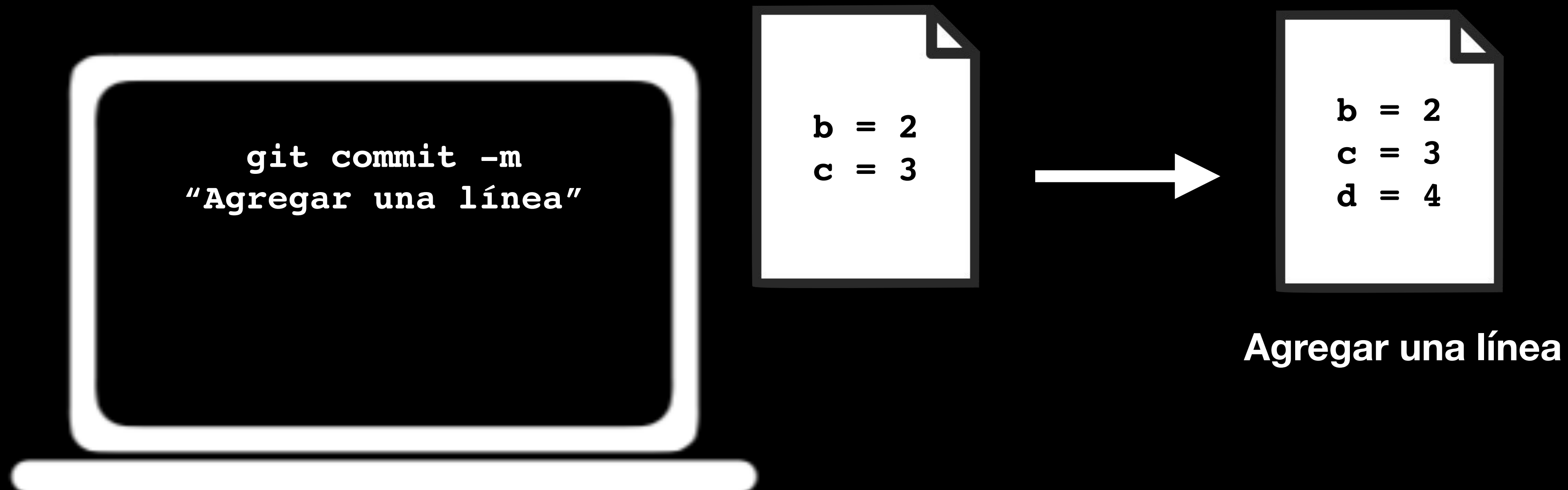
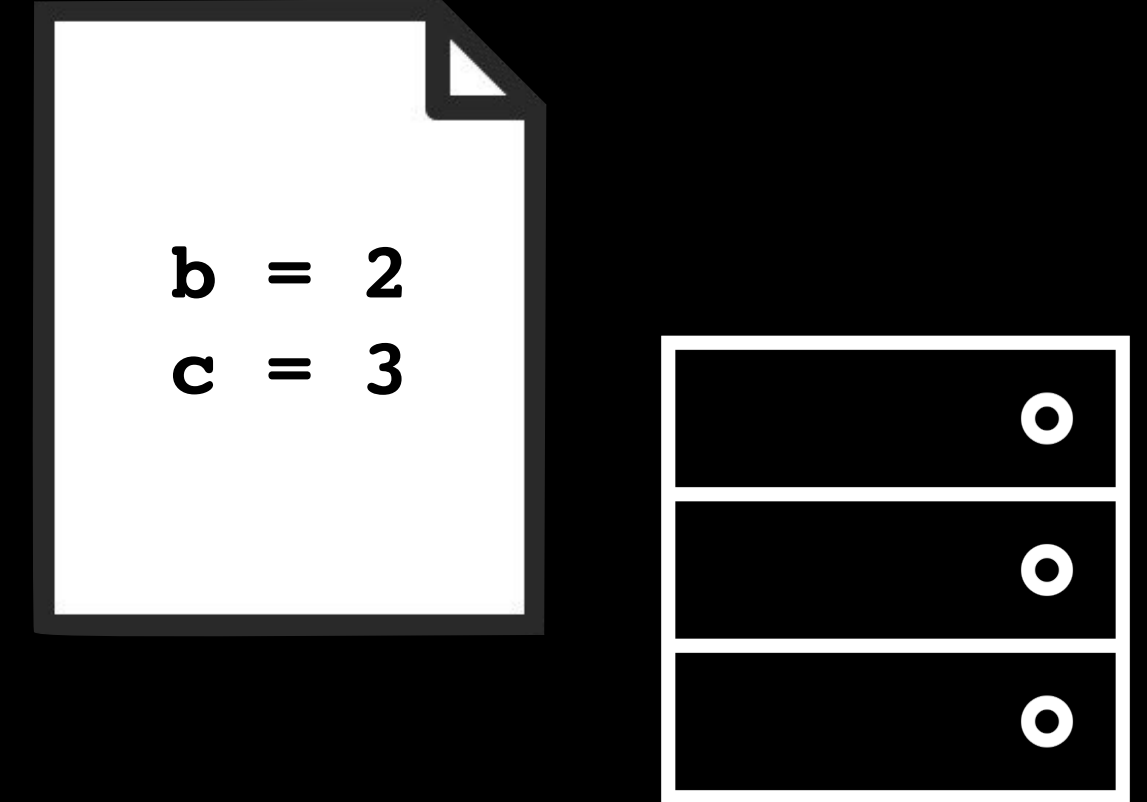
# git commit -m "mensaje"

- Guarda los cambios realizados al repositorio como una nueva versión
- Guarda un mensaje
- **git commit -am "mensaje"** agrega y guarda al mismo tiempo



# git commit -m "mensaje"

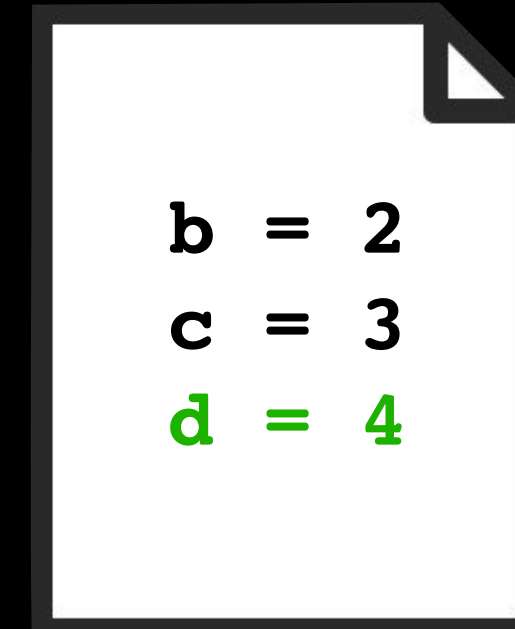
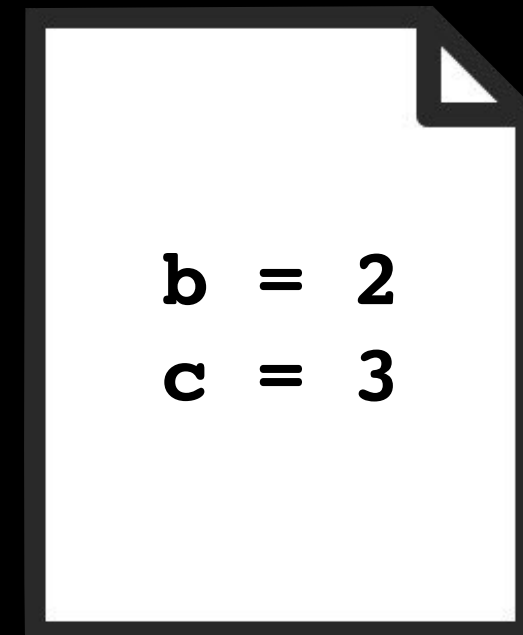
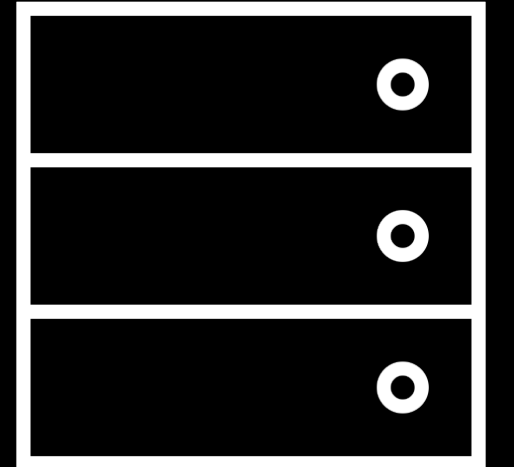
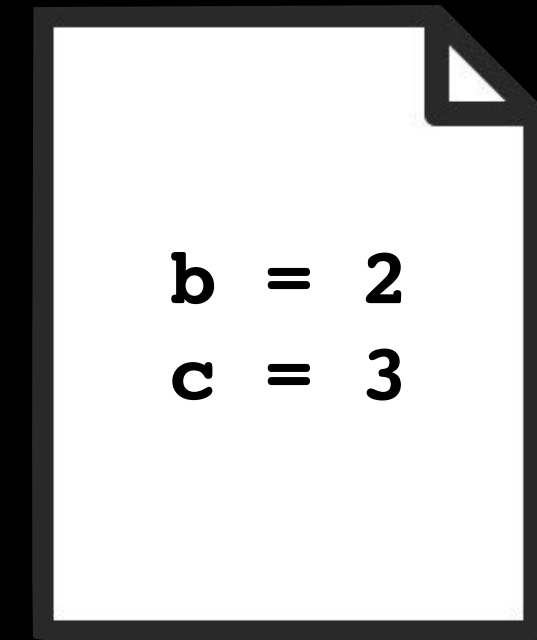
- Guarda los cambios realizados al repositorio como una nueva versión
- Guarda un mensaje
- **git commit -am "mensaje"** agrega y guarda al mismo tiempo



```
git status
```

# git status

- Muestra el estado actual del repositorio



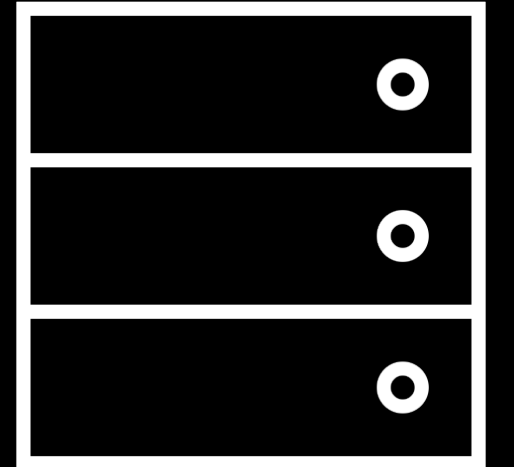
Agregar una línea



# git status

- Muestra el estado actual del repositorio

```
b = 2  
c = 3
```



```
b = 2  
c = 3
```



```
b = 2  
c = 3  
d = 4
```

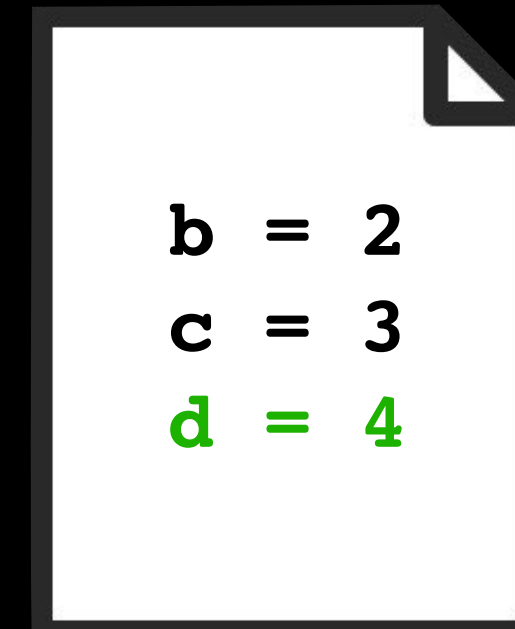
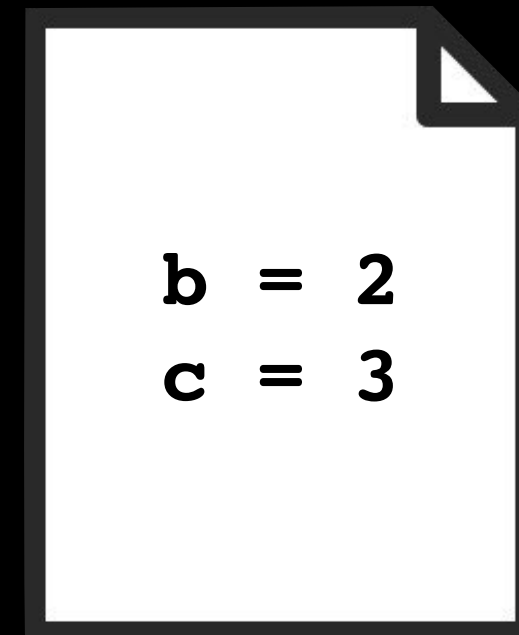
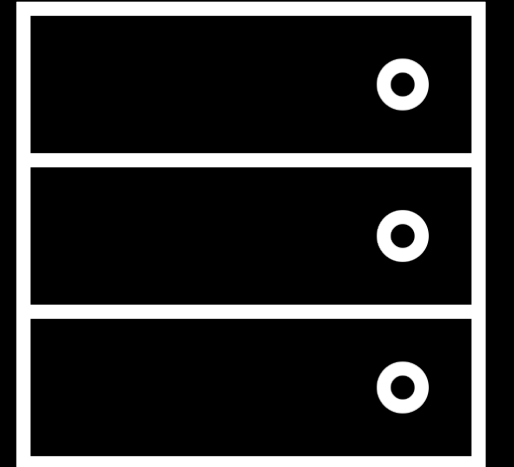
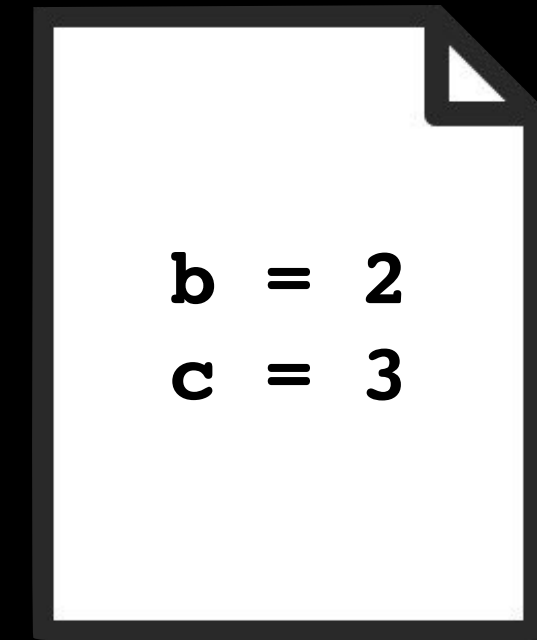
Agregar una línea



```
git status
```

# git status

- Muestra el estado actual del repositorio



Agregar una línea



On branch master

Your branch is ahead of 'origin/master' by 1 commit.

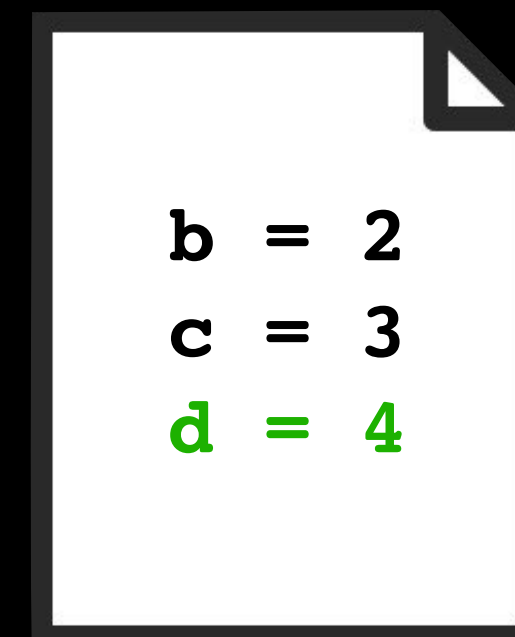
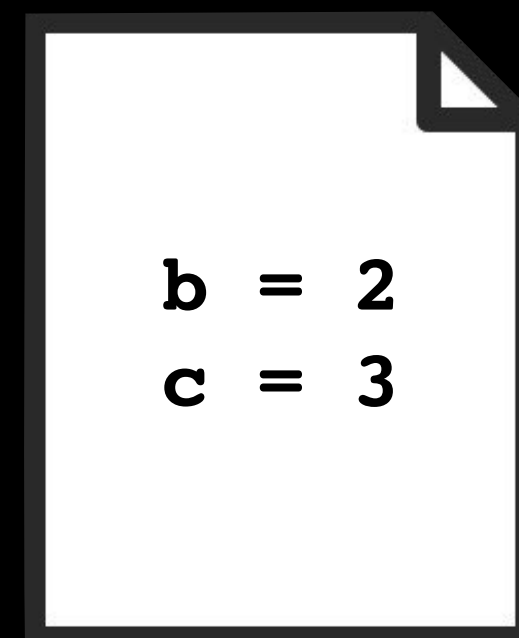
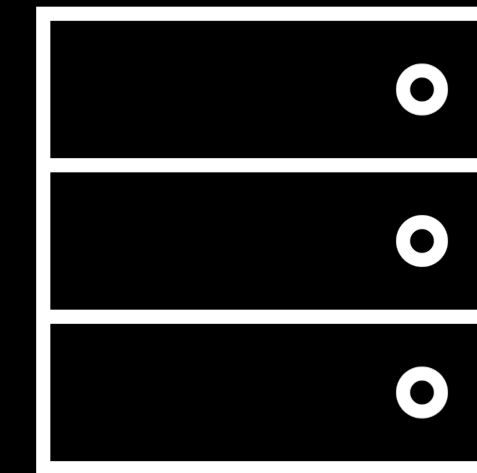
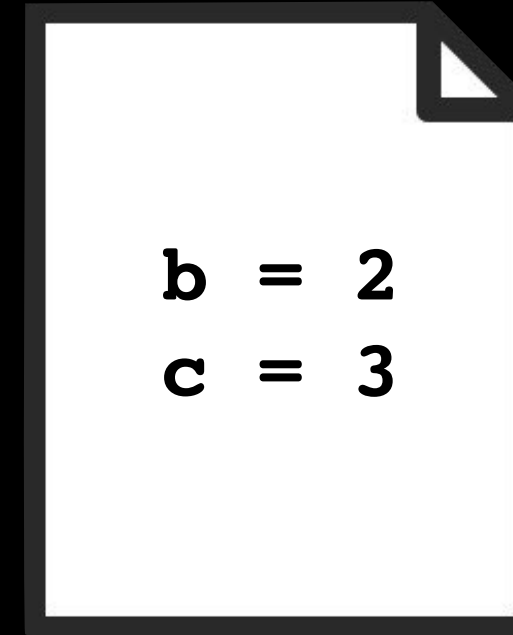
(use "git push" to publish your local commits)



```
git push
```

# git push

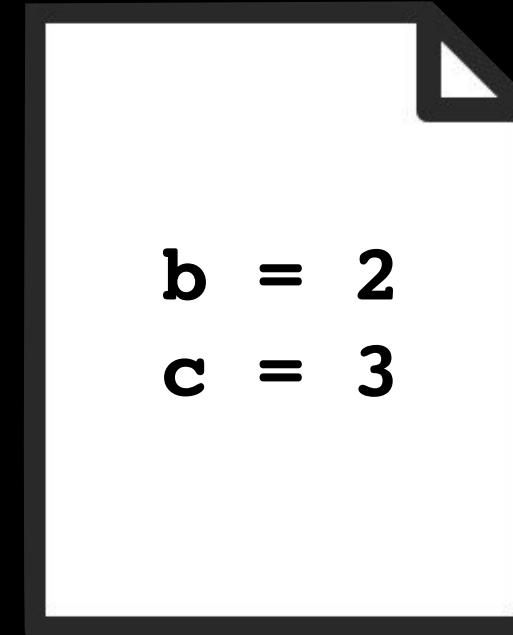
- Envía los cambios comprometidos al repositorio remoto
- Explícitamente se puede escribir como **git push origin master**



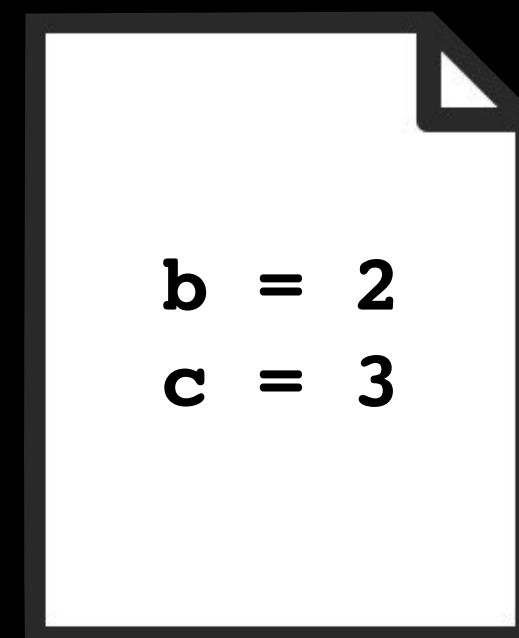
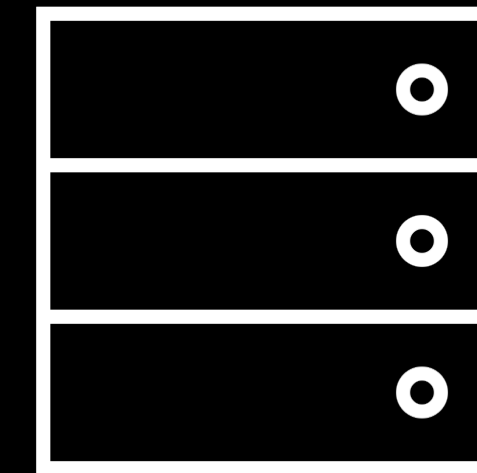
Agregar una línea

# git push

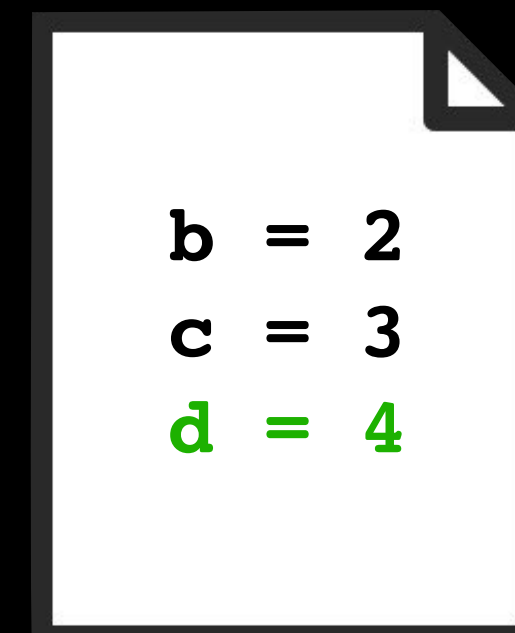
- Envía los cambios comprometidos al repositorio remoto
- Explícitamente se puede escribir como **git push origin master**



```
b = 2  
c = 3
```



```
b = 2  
c = 3
```

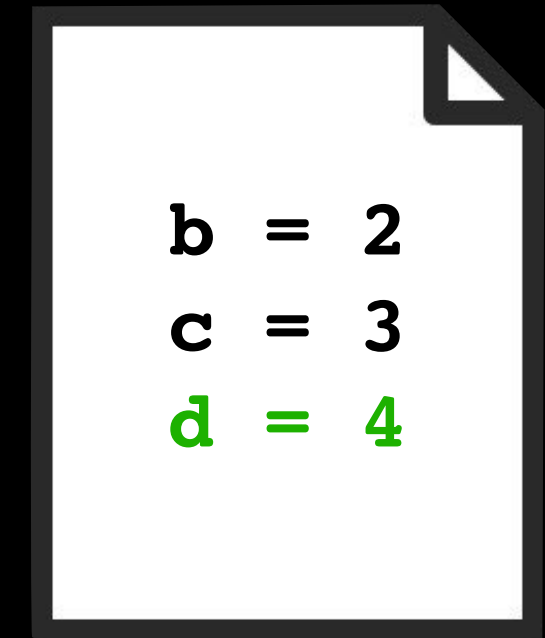
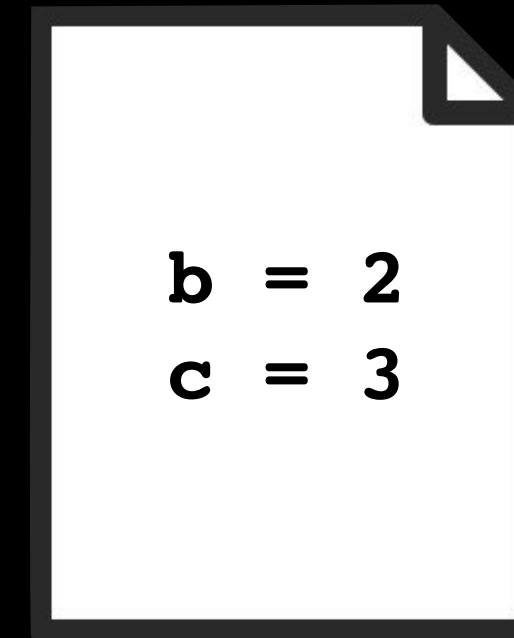


```
b = 2  
c = 3  
d = 4
```

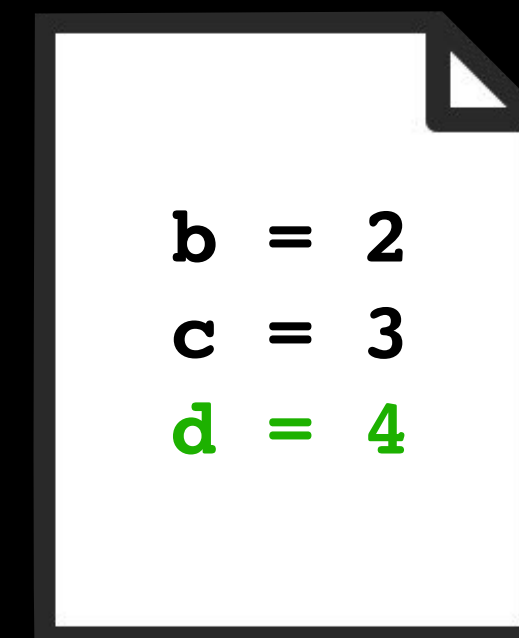
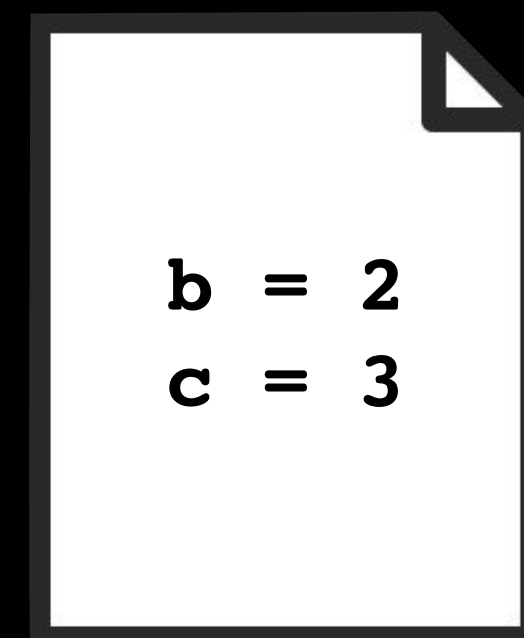
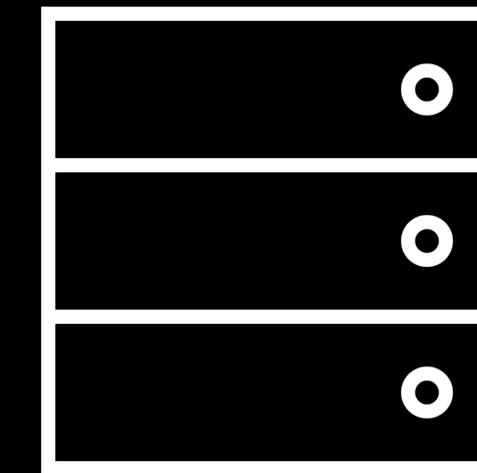
Agregar una línea

# git push

- Envía los cambios comprometidos al repositorio remoto
- Explícitamente se puede escribir como **git push origin master**



Agregar una línea

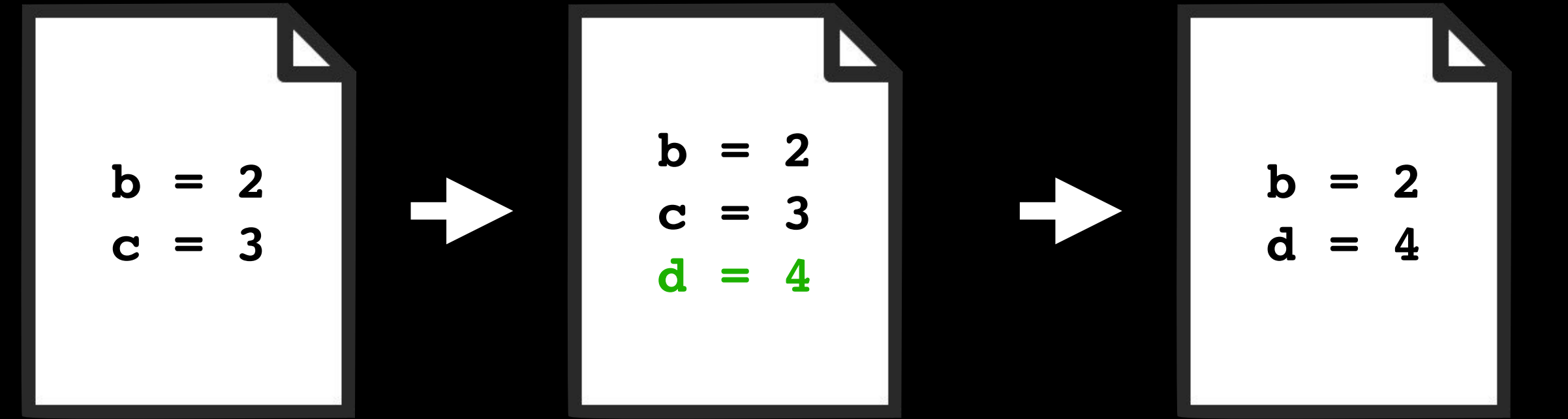


Agregar una línea

```
git pull
```

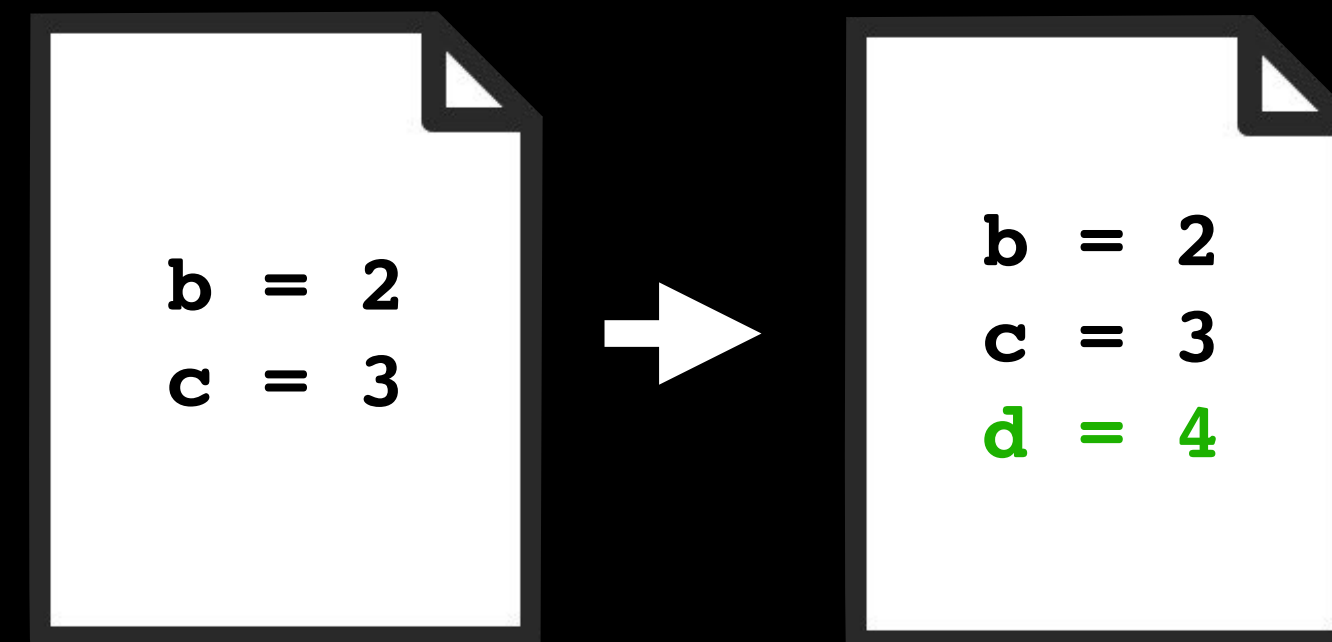
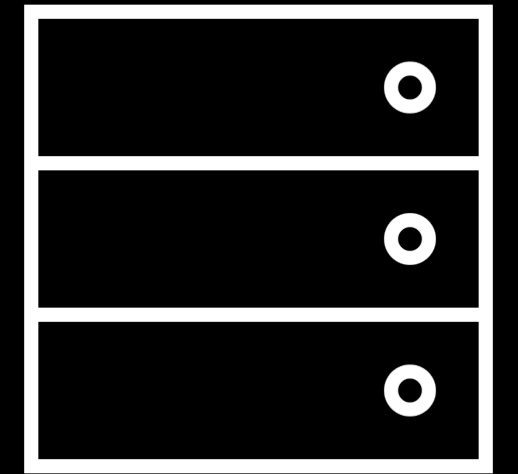
# git pull

- Descarga los cambios desde un repositorio remoto



Agregar una línea

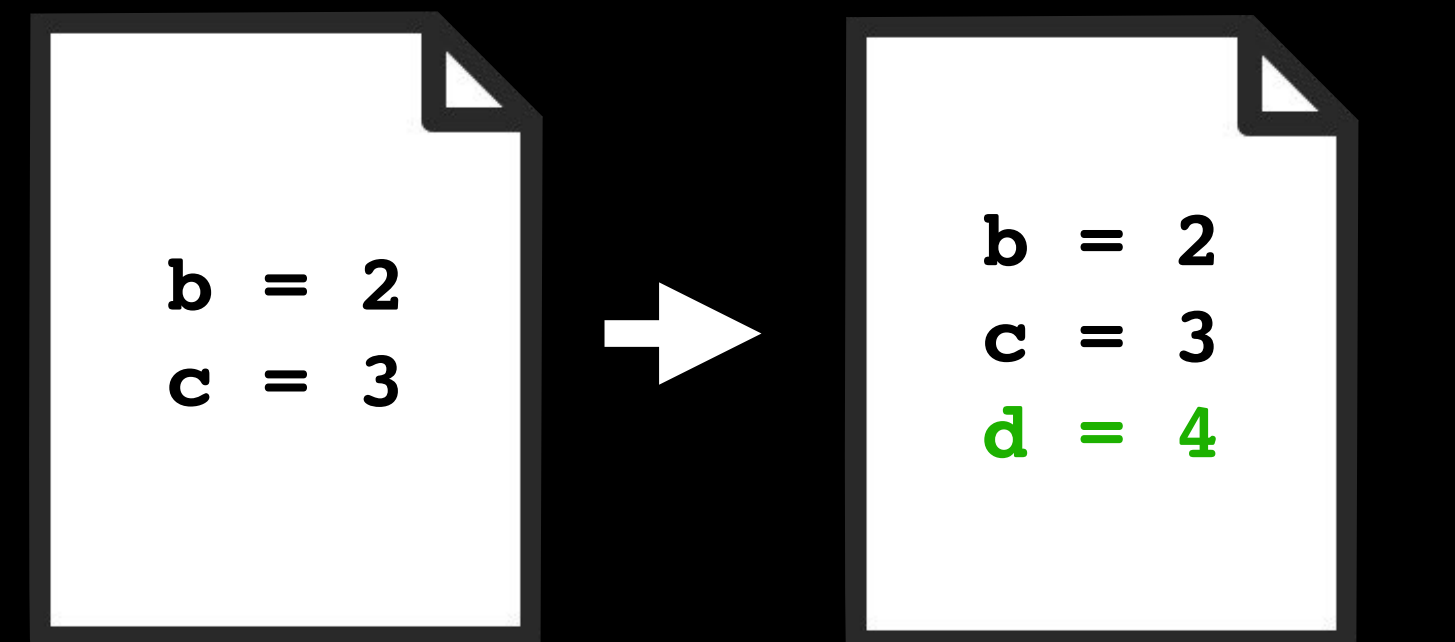
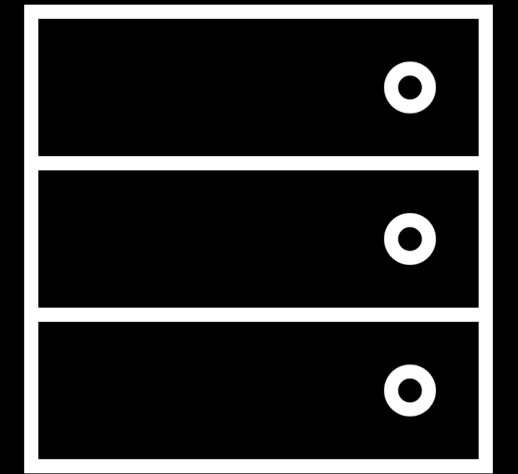
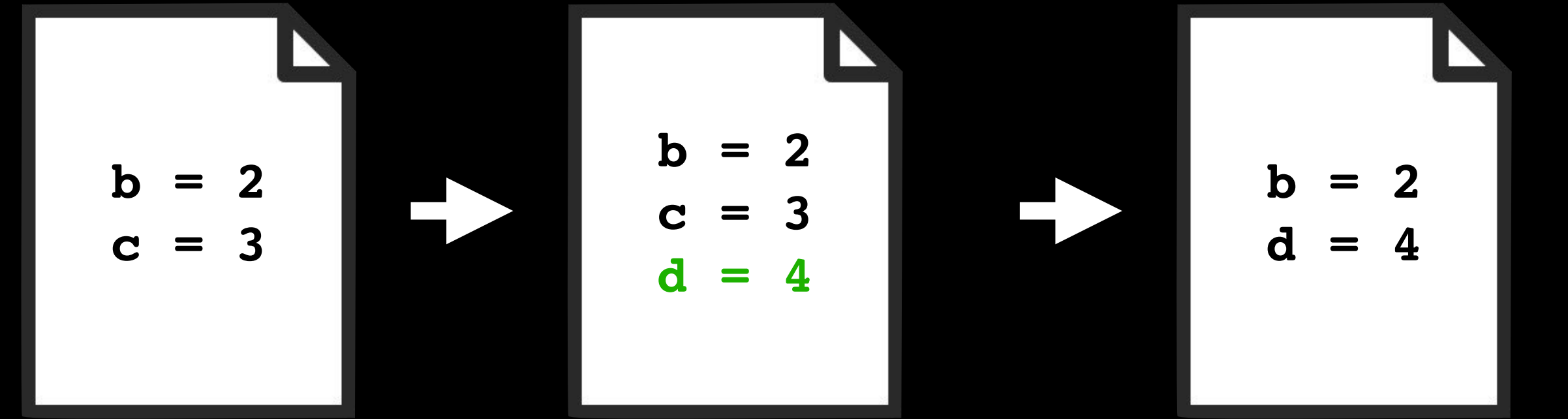
Quitar una línea



Agregar una línea

# git pull

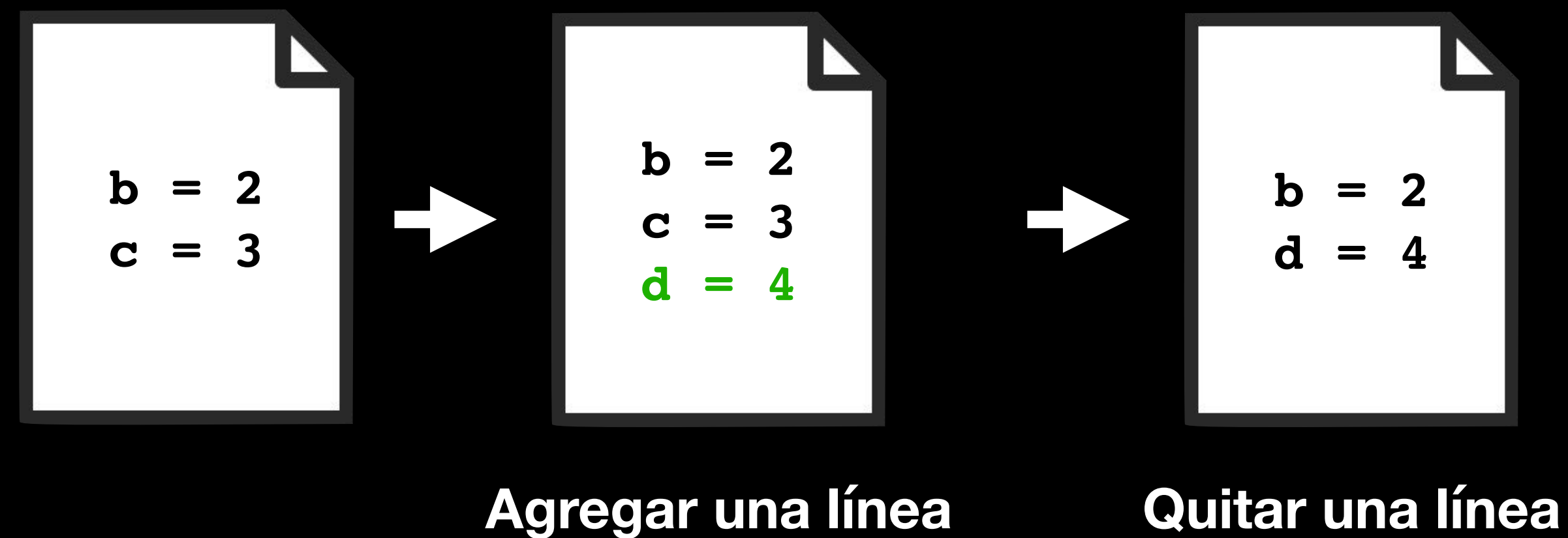
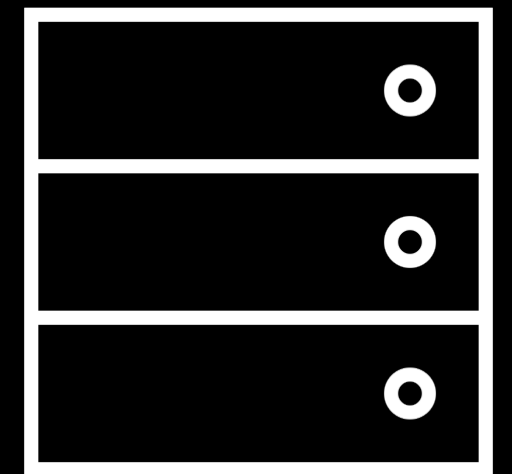
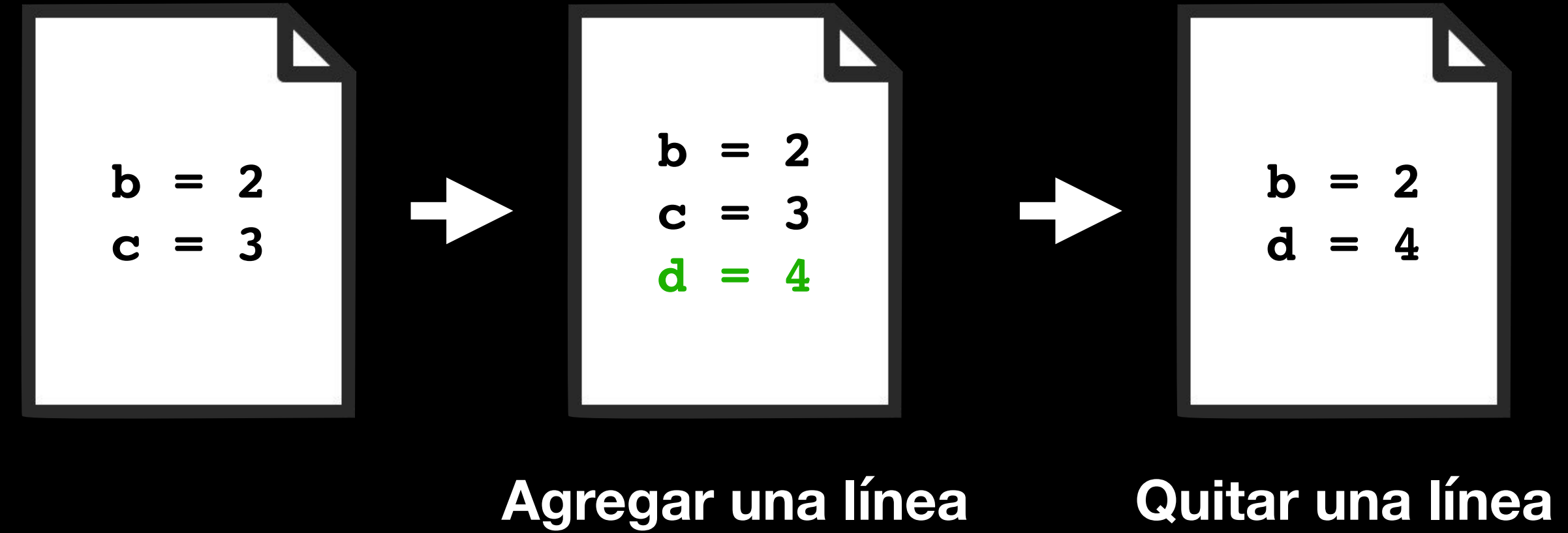
- Descarga los cambios desde un repositorio remoto



Agregar una línea

# git pull

- Descarga los cambios desde un repositorio remoto





# Merge Conflicts

# Merge Conflicts

- Cuando dos versiones diferentes del código no pueden ser fusionadas “Merge” automáticamente, se requiere resolver los conflictos



# Merge Conflicts

- Cuando dos versiones diferentes del código no pueden ser fusionadas “Merge” automáticamente, se requiere resolver los conflictos

A simple line-art icon of a laptop or terminal window. It consists of a rectangular frame with a slightly rounded top, and a horizontal line at the bottom representing the base. Inside the frame, the text 'git pull' is displayed in a monospaced font.

```
git pull
```

**CONFLICT (content): Merge conflict in vars.py**

**Automatic merge failed; fix conflicts and then commit the result**

# Merge Conflicts

- Cuando dos versiones diferentes del código no pueden ser fusionadas “Merge” automáticamente, se requiere resolver los conflictos



```
git pull
```

Tus cambios

Cambios remotos

```
a = 1
<<<<<<< HEAD
b = 2
=====
b = 0
>>>>>>> 65876587658765875e
c = 3
d = 4
e = 5
```

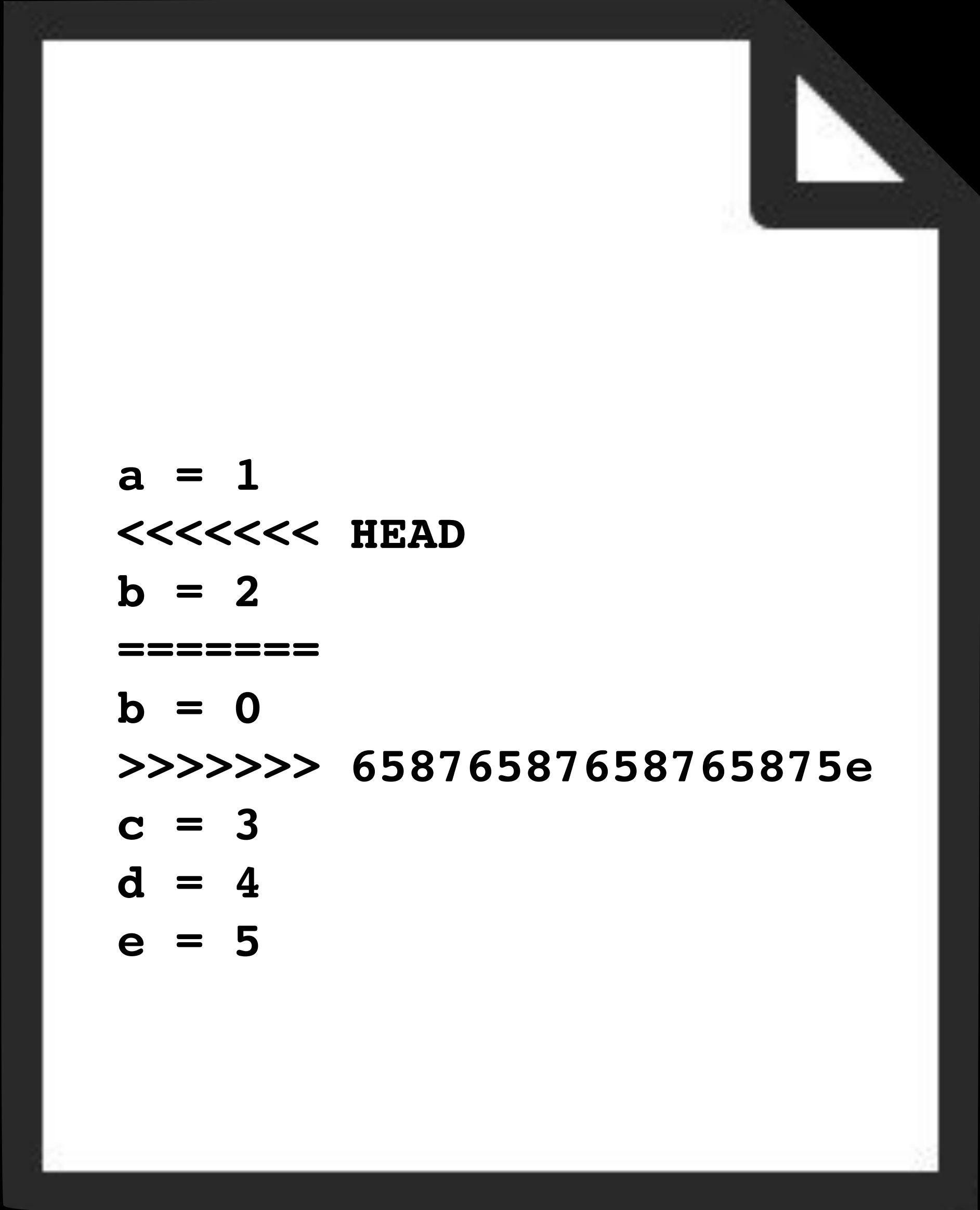
identificador del conflicto

# Merge Conflicts

- Cuando dos versiones diferentes del código no pueden ser fusionadas “Merge” automáticamente, se requiere resolver los conflictos



```
git pull
```



```
a = 1
<<<<<<< HEAD
b = 2
=====
b = 0
>>>>>>> 65876587658765875e
c = 3
d = 4
e = 5
```

# Merge Conflicts

- Cuando dos versiones diferentes del código no pueden ser fusionadas “Merge” automáticamente, se requiere resolver los conflictos



```
git pull
```

```
a = 1
```

```
b = 2
```

```
b = 0
```

```
c = 3
```

```
d = 4
```

```
e = 5
```

# Merge Conflicts

- Cuando dos versiones diferentes del código no pueden ser fusionadas “Merge” automáticamente, se requiere resolver los conflictos

A white-outlined icon of a laptop screen.

```
git pull
```

```
a = 1
```

```
b = 2
```

```
c = 3
```

```
d = 4
```

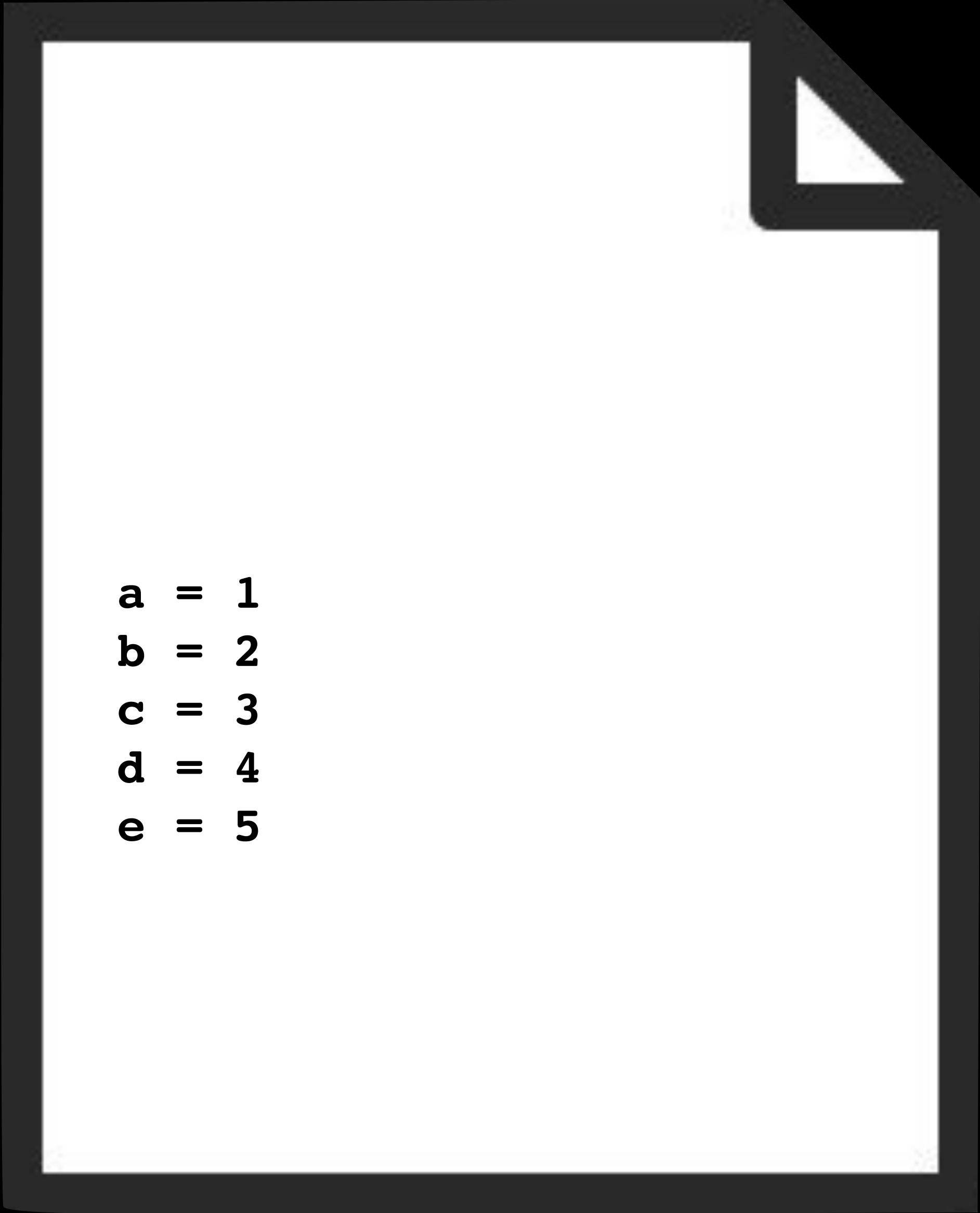
```
e = 5
```

# Merge Conflicts

- Cuando dos versiones diferentes del código no pueden ser fusionadas “Merge” automáticamente, se requiere resolver los conflictos

A white-outlined icon of a laptop screen.

```
git pull
```

A white-outlined icon of a document with a folded corner.

```
a = 1  
b = 2  
c = 3  
d = 4  
e = 5
```



git log


# git log

- Muestra el historial de cambios y mensajes



# git log

- Muestra el historial de cambios y mensajes



**git log**

# git log


- Muestra el historial de cambios y mensajes

```
commit 678698769876987698769876987e
Author: Jorge Mauricio <jorge.mauricio@me.com>
Date: Tue Oct 25 20:55:23 2017 -0600
```

Quitar una línea

```
commit 12342341234234523452345e
Author: Jorge Mauricio <jorge.mauricio@me.com>
Date: Tue Oct 25 20:50:00 2017 -0600
```

Agregar una línea

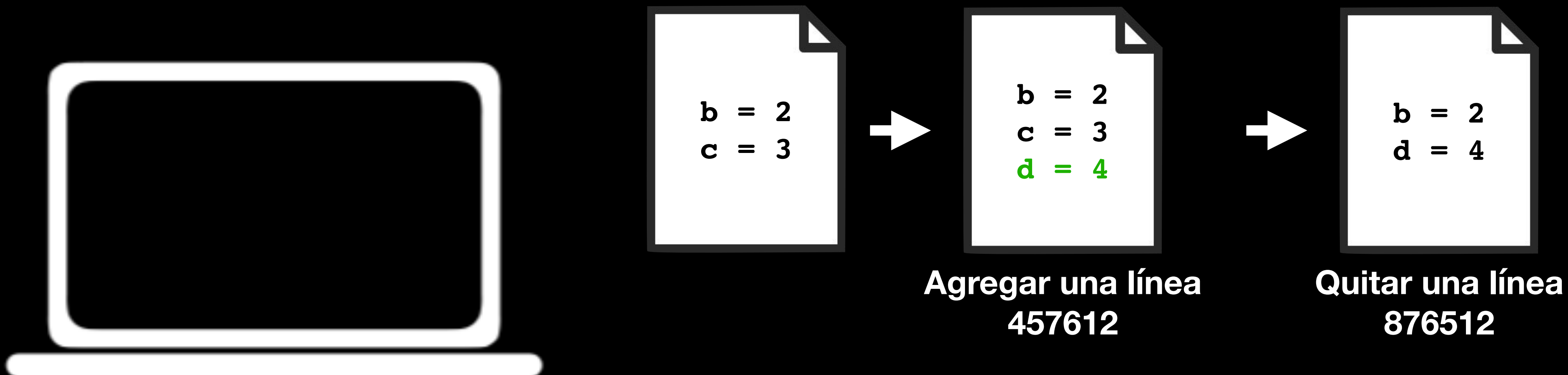


git log

```
git reset
```

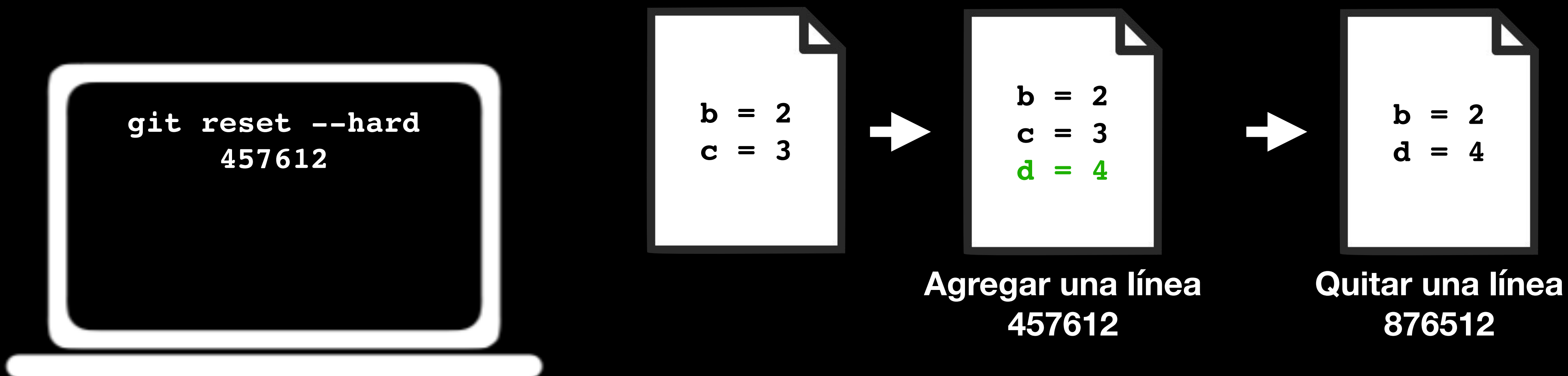
# git reset

- **git reset --hard <commit>**  
regresa el código a su versión anterior
- **git reset --hard origin/master**  
regresa el código a la versión del repositorio remoto



# git reset

- **git reset --hard <commit>**  
regresa el código a su versión anterior
- **git reset --hard origin/master**  
regresa el código a la versión del repositorio remoto



# git reset

- **git reset --hard <commit>**  
regresa el código a su versión anterior
- **git reset --hard origin/master**  
regresa el código a la versión del repositorio remoto



```
git reset --hard  
457612
```

```
b = 2  
c = 3
```

```
b = 2  
c = 3  
d = 4
```

Agregar una línea  
457612



branching

# Branching

- Branch es una versión del repositorio
- Cada una de las ramas “branch” tiene su propio historial de versiones y la versión actual

git branch

# git branch

- Despliega todas las branch del código
- Crea una nueva branch con **git branch <nombre\_branch>**
- Cambia de branch con **git checkout <nombre\_branch>**

master

```
b = 2  
c = 3  
d = 4
```



# git branch

- Despliega todas las branch del código
- Crea una nueva branch con **git branch <nombre\_branch>**
- Cambia de branch con **git checkout <nombre\_branch>**

master

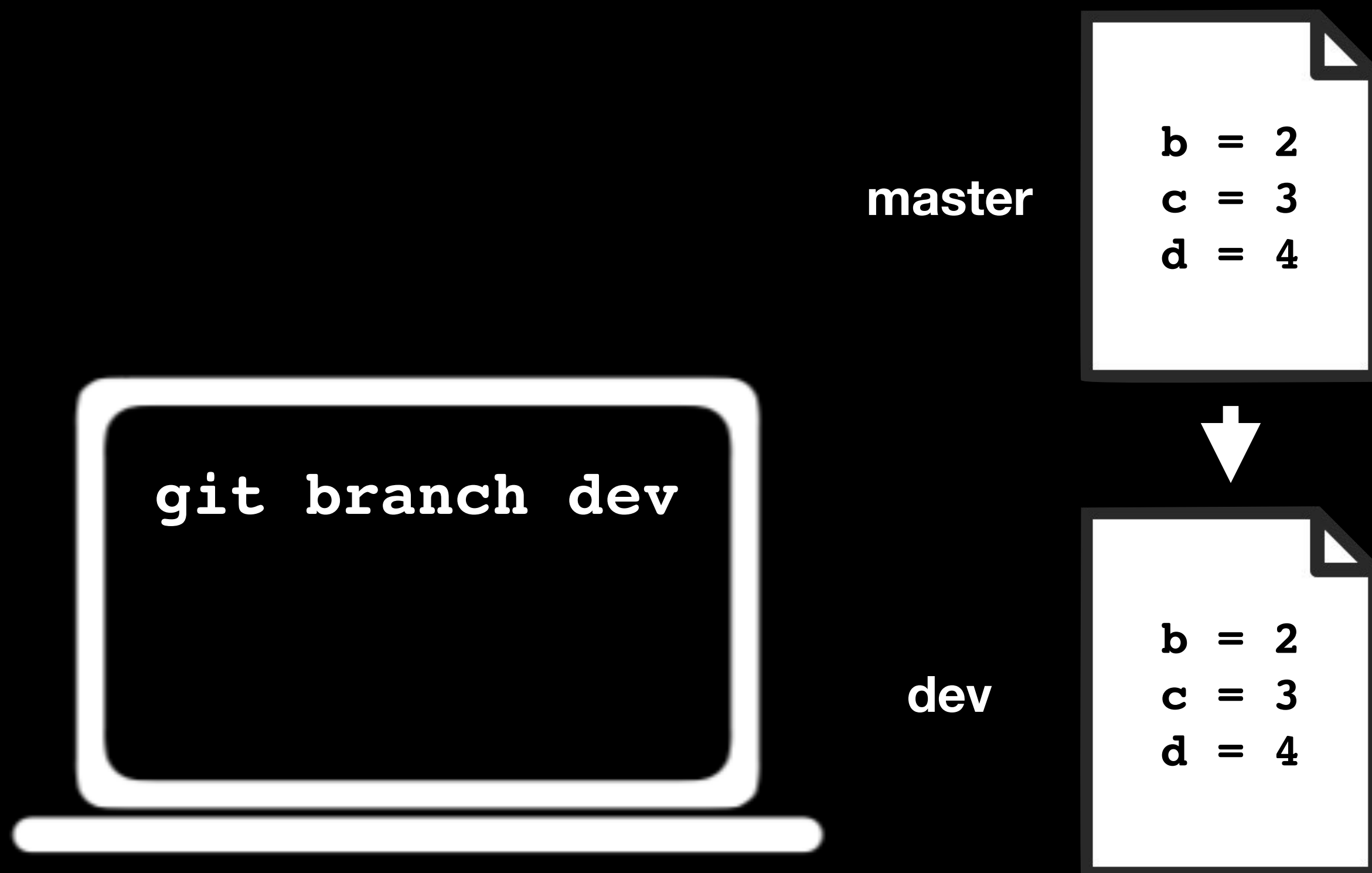
```
b = 2  
c = 3  
d = 4
```



```
git branch dev
```

# git branch

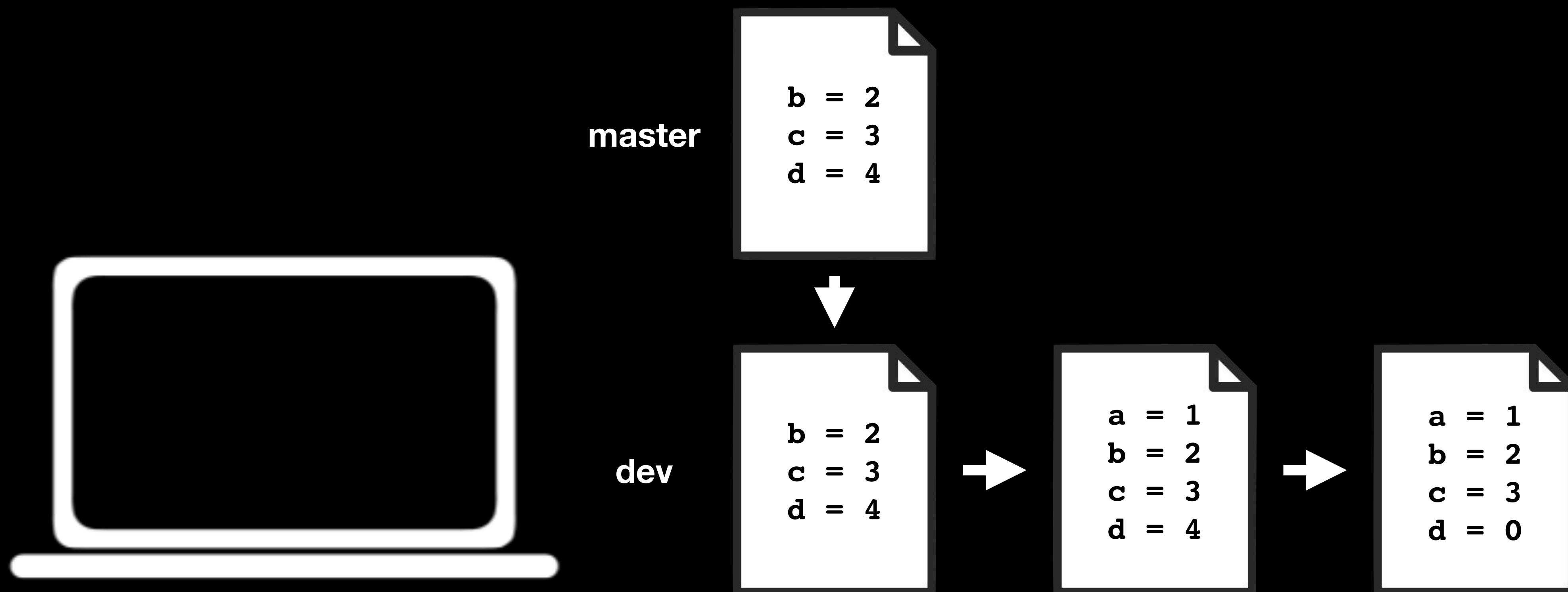
- Despliega todas las branch del código
- Crea una nueva branch con **git branch <nombre\_branch>**
- Cambia de branch con **git checkout <nombre\_branch>**



git merge

# git merge

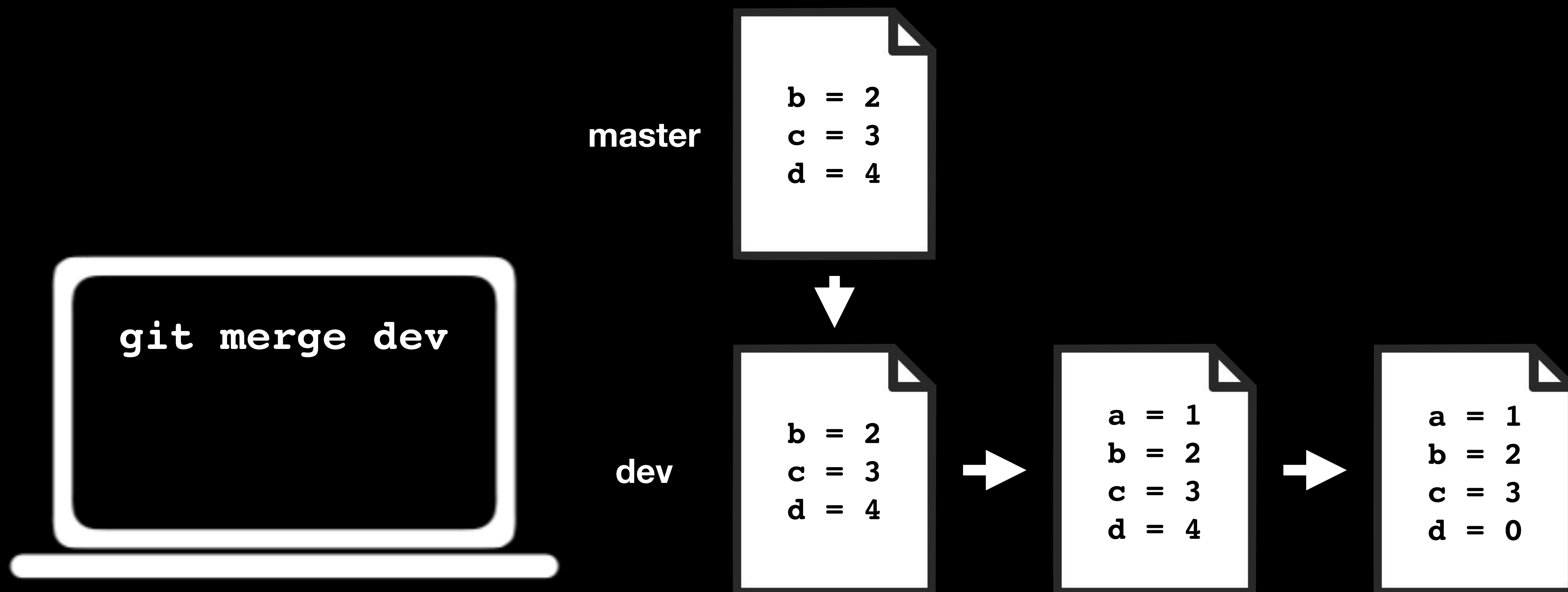
- **git merge <nombre\_branch>**  
fusiona el branch **nombre\_branch** con el actual branch





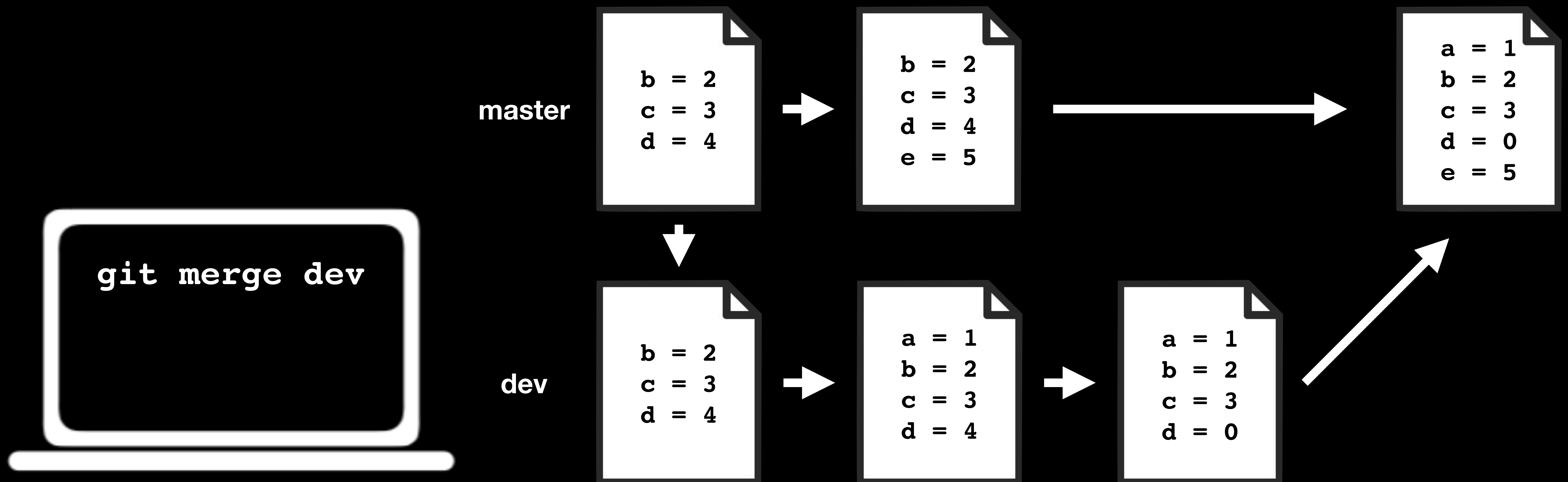
# git merge

- **git merge <nombre\_branch>**  
fusiona el branch **nombre\_branch** con el actual branch



# git merge

- `git merge <nombre_branch>`  
fusiona el branch `nombre_branch` con el actual branch



# Configuración inicial

```
git config --global user.email "usuario@mail.com"
```

```
git config --global user.name "Nombre Apellido"
```

# Agregar un repositorio

```
git remote add <alias> <url del repositorio>
```

```
git remote add laboratorio pi@172.16.1.229:/home/pi/repository/agentes_cambio.git
```

# Consultar repositorios

```
git remote -v
```

```
pi@172.16.1.229:/home/pi/repository/agentes_cambio.git(pull)
```

```
pi@172.16.1.229:/home/pi/repository/agentes_cambio.gi(fetch)
```

# Remover un repositorio

```
git remote rm <alias>
```

# Contacto

**IIS Jorge Ernesto Mauricio Ruvalcaba**  
[jorge.ernesto.mauricio@gmail.com](mailto:jorge.ernesto.mauricio@gmail.com)

**Muchas gracias!**