

1. El **coordinador** crea un repositorio y da acceso a su compañero
 - a. Crear repositorio
 - b. Settings
 - c. Collaborators
 - d. Add people
 - e. Enviar invitación al nombre de usuario o email que tenga en GitHub
2. El **otro**, busca la invitación en su email y la acepta
3. Mientras, el **coordinador** sube los archivos iniciales (si el repositorio está vacío)
 - a. Clonar repositorio
 - i. `git clone *url ssh*`
 - b. Añadir archivos iniciales
 - c. Hacer push
 - i. `git add .`
 - ii. `git commit -m "mensaje"`
 - iii. `git push`
4. El **otro** clona el repositorio, que ya tiene los archivos.
 - a. `git clone *url ssh*`
5. **Ambos** crean una rama nueva (cada uno la suya)
 - a. Dentro del repositorio: `git checkout -b *nombre*`
6. Cada uno trabaja en su parte
 - a. Para abrir directamente la carpeta en visual: `code .`
 - b. Cada uno trabaja en el archivo que le corresponde, sin modificar el que le toca al compañero.
7. Cuando **alguien** haya acabado una parte, hace commit
 - a. `git add .`
 - b. `git commit -m "mensaje"`
 - c. `git push`
 - d. (La primera vez saldrá un error y un código que empieza por git push, solo hay que copiar el código entero y ejecutarlo)
8. Si el commit es definitivo (o la otra persona lo necesita) se mergea a main/master
 - a. Es probable que aparezca directamente un botón nuevo de `compare & pull request`. Click ahí.
 - b. Si no, click en Pull requests -> `New pull request`. Asegurarse de que en base pone main/master y en compare la rama a fusionar. -> `Create pull request`
 - c. `Create pull request`
9. La persona que **no** haya hecho pull request,

- a. Va a Pull requests
- b. Selecciona la pull request
- c. Click en `Merge pull request` -> `Confirm merge`
- d. Saldrá la opción de borrar la rama. Hacerle clic si no se va a seguir trabajando en ella.
- e. Si tiene trabajo sin commitear, ~~desde terminal hace git stash para guardarlo temporalmente~~
 - i. Para simplificar, de momento simplemente hará un `commit`
- f. Cambia a la rama main y la actualiza
 - i. `git checkout main`
 - ii. `git pull`
- g. Vuelve a su rama y la actualiza
 - i. `git checkout *nombre*`
 - ii. `git rebase main`
- ~~h. Recupera el trabajo que había guardado con git stash pop~~

10. Quien no haya acabado, sigue trabajando. Cuando acabe vuelve al paso 7.

11. Para borrar las ramas en local

- a. Cambiar a main `git checkout main`
- b. Borrar la rama `git branch -d *nombre*`
- c. Borrado forzado `git branch -D *nombre*`

Si en algún momento hay algún conflicto...

Seguramente os daréis cuenta al intentar hacer push, que pide que se haga pull primero. Hacemos pull **desde visual**, que nos aportará herramientas para solucionarlo.

Accept current change: nos quedamos con la version local

Accept incoming change: nos quedamos con la versión remota

Accept both changes: Nos quedamos con las dos versiones, añadiendo lineas

Una vez hayamos seleccionado la opción correspondiente, volvemos a hacer `git add .`

En caso de que surgiera el conflicto durante un rebase, continuamos con `git rebase --continue`

Si hacemos algún cambio más, volvemos a hacer `git add .`, hacemos `git commit -m "mensaje"` y `git push`, y ya no debería dar problemas.