

Trabajando con git

A good starting tutorial

man gitcore-tutorial

- Subir un directorio existente como repositorio al repositorio previamente creado en gitlab.com (Es importante no crear los archivos sugeridos *README* o *.gitignore*):

```
cd existing_folder
git init
git add .
git commit -m "Initial commit"
git remote add origin git@gitlab.com:pavelinux/repo_creado.git
git push -u origin master
git log # Revisa historial de commits hechos en el repo
```

En caso de que ya se haya hecho commit en el repo local y haya commits en la versión remota, hacer lo siguiente

```
git pull
git branch --set-upstream-to=origin/master master
git pull
git config pull.rebase true
git pull
git push
```

- Si ya existe el repositorio en el servidor remoto (en este caso en gitlab), por ejemplo:

```
git clone git@gitlab.com:pavelinux/documentacion_ometeotl.git
```

- Si se va a exportar un repositorio local al GitLab, por ejemplo:

Primero crear el repositorio en GitLab (o Bitbucket) y luego ejecutar el comando para importar el *repo*:

```
cd existing_repo
git remote rename origin old-origin
git remote add origin git@gitlab.com:pavelinux/documentacion_ometeotl.git
git push -u origin --all
git push -u origin --tags
```

Implementación Servidor local (casero) GIT

Pasos efectuados en servidor git.jupyter.theworkpc.com (Puerto SSH 7721)

Se instala git en servidor Debian, Ubuntu:

```
adduser git
```

```
cd
```

```
$ mkdir .ssh && chmod 700 .ssh
```

```
$ touch .ssh/authorized_keys && chmod 600 .ssh/authorized_keys
```

Usuario git

1. Se crea directorio de tesis.git (como usuario git) en la siguiente ruta:

```
mkdir /srv/git/tesis.git
```

```
cd /srv/git/tesis.git/
```

```
git init --bare
```

2. Se habilitan las llaves ssh para conectarse al servidor

```
cd /srv/git/tesis.git/
```

```
eval "$(ssh-agent -s)"
```

```
ssh-add ~/.ssh/bitbucket_ed25519
```

Desde la laptop cliente, modificar el fichero

```
tu_repo/.git/config
```

Agregar las siguientes lineas al fichero config

```
[remote "home"]
```

```
url = ssh://git@git.jupyter.theworkpc.com:7721/srv/git/tesis.git
```

```
fetch = +refs/heads/*:refs/remotes/home/*
```

Enseguida subir el código que se quiere versionar en el repositorio:

```
git push home master
```

Sesión en equipo cliente con un servidor GIT "casero"

Desde un equipo cliente

1. Copiar las llaves (previamente generadas con el comando `ssh-keygen -t ed25519`

`-C "pavel-bitbucket"`, por ejemplo) del usuario del repositorio al fichero

`authorized_keys` en el directorio:

```
/home/git/.ssh/authorized_keys
```

En este caso se copiaran las llaves que se encuentran en bitbucket en nuestro equipo cliente:

```
ssh-copy-id -p 7721 -i ~/.ssh/bitbucket.pub git@git.jupyter.theworkpc.com
```

2. Agregar al repositorio casero el código que se quiere versionar:

Desde la laptop cliente, modificar el fichero

```
tesis/.git/config
```

Agregar las siguientes lineas

```
[remote "home"]
```

```
url = ssh://pavel@git.jupyter.theworkpc.com:7721/srv/git/tesis.git
```

```
fetch = +refs/heads/*:refs/remotes/home/*
```

```
git remote -v
```

SALIDA:

```
origin ssh://pavel@git.jupyter.theworkpc.com:7721/srv/git/tesis.git (fetch)
```

```
origin ssh://pavel@git.jupyter.theworkpc.com:7721/srv/git/tesis.git (push)
```

4. Se hace una verificación, se algún fichero y luego se hace un commit al servidor casero:

```
git add .  
git config user.name "Pavel"  
git config user.email "pavel@atmosfera.unam.mx"  
git commit -m "Cambiando a servidor casero"
```

5. Si se desea subir las modificaciones al servidor bitbucket con los cambios que se hayan hecho en el servidor casero:

```
git pull home master  
git push origin master
```

GIT / Gitlab

0.1 Consejos para realizar *commits*:

- `git add specific_files_per_topic` Se agregan a la etapa (*stage*) los archivos o directorios relacionados al tema.

```
git add -p setup.sh
```

- Hacer *commits* por tema o cambio realizado. *Un commit por tema*
- El mensaje del *commit* deberá ser claro y conciso

```
git commit -m "SUBJECT" -A "FULL DESCRIPTION" # Por qué se hace el cambio,  
consideraciones importantes, versiones, etc
```

0.2 Eliminar un commit:

```
git reset HEAD-1
```

La bandera *HEAD-1* indica el ultimo cambio realizado en nuestro repositorio local.

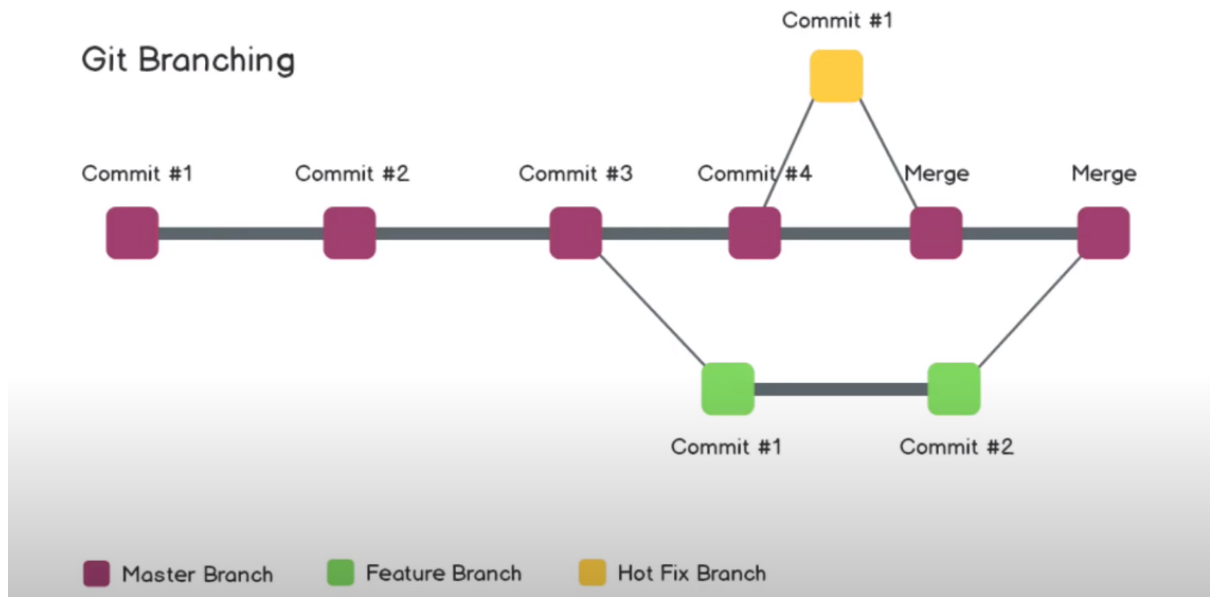
0.3 Eliminar un commit y los cambios realizados:

```
git reset --hard HEAD-1
```

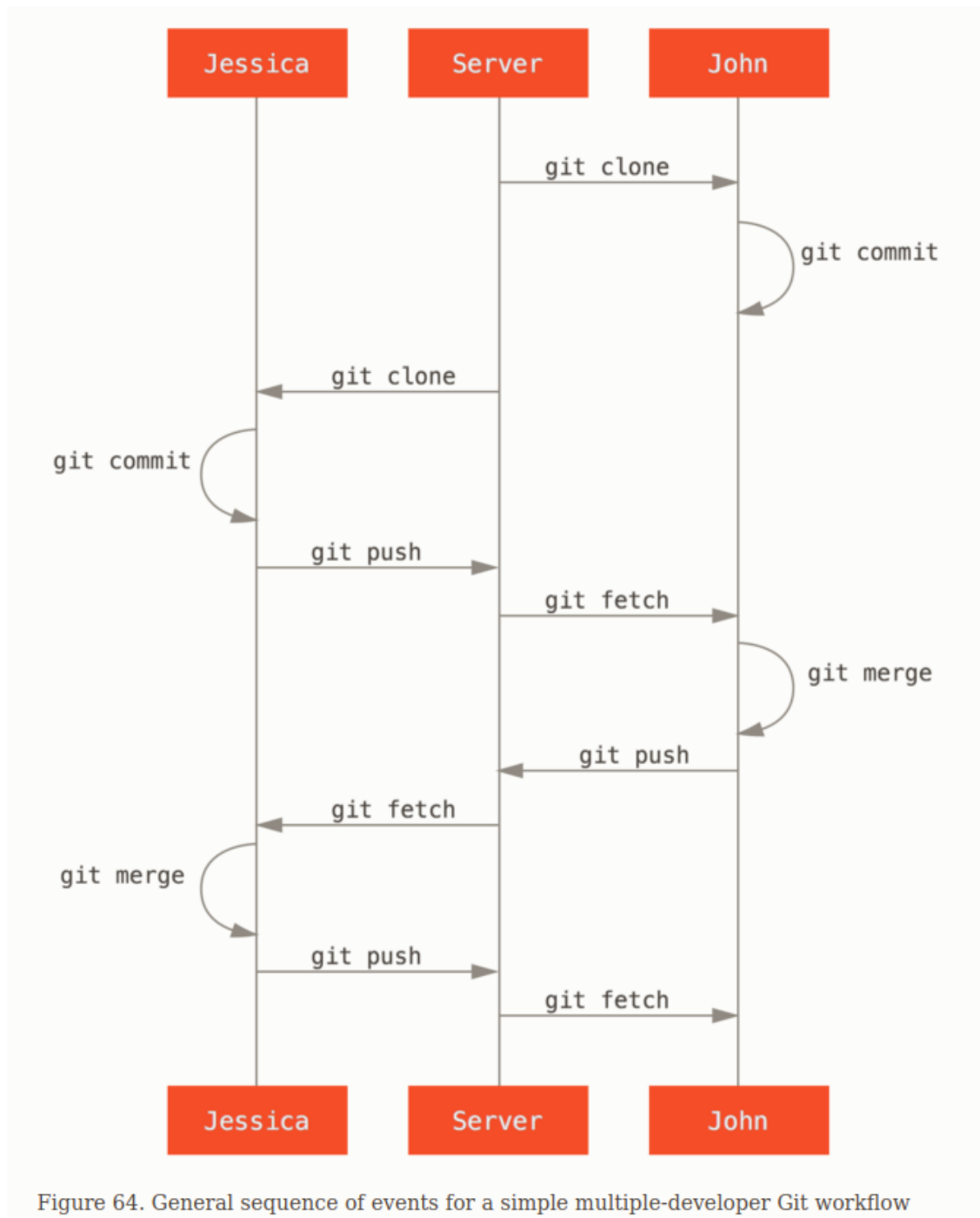
0.4 Flujos de trabajo y ramas de trabajos (*branching*):

0.4.1 Basic *branching*

Git Branching

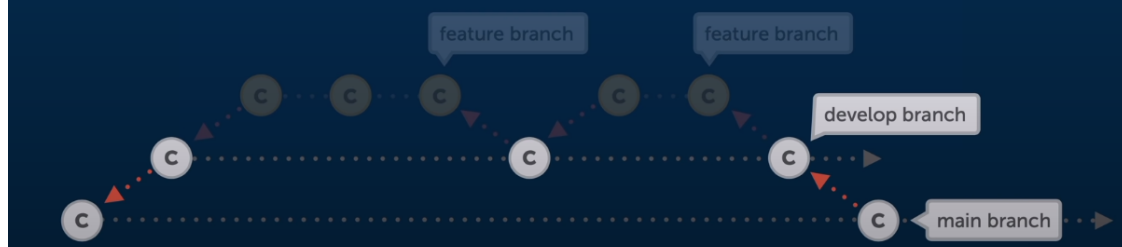


1. Private Small Team:



Long-Running & Short-Lived Branches

- exist through the complete lifetime of the project
- often, they mirror “stages” in your dev life cycle



2. Forks & Pull requests

```
# Primero hacer el fork del repo deseado
# Luego clonar el repo desde nuestra cuenta
# Crear una rama:
git branch -l
git branch my_tests
git checkout my_tests
... some changes on filex
git add filex
git commit -am "Added some changes"
git push --set-upstream origin master

# Para agregar una rama al repositorio:
git push origin dev
```

```
soporte@tlaloc -> [63]
(~/repos/courses) 20:42:08 -> git commit -am "Agrego docs de git para revisar. Para taller de git"
[dev 402eba7] Agrego docs de git para revisar. Para taller de git
2 files changed, 161 insertions(+)
create mode 100644 docs/md/git.md
create mode 100644 docs/md/git_enable_ssh_publickey.md
soporte@tlaloc -> [64]
(~/repos/courses) 20:43:18 -> git status
On branch dev
nothing to commit, working tree clean
soporte@tlaloc -> [65]
6 (~/repos/courses) 20:43:20 -> git push origin dev
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 4 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (6/6), 2.47 KiB | 2.47 MiB/s, done.
Total 6 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'dev' on GitHub by visiting:
remote:   https://github.com/doncafe/courses/pull/new/dev
remote:
To github.com:doncafe/courses.git
* [new branch]      dev -> dev
soporte@tlaloc -> [66]
(~/repos/courses) 20:48:10 -> git
```

Agrego docs de git para revisar. Para taller de git #1

Open

doncafe wants to merge 1 commit into `main` from `dev`

Conversation 0

Commits 1

Checks 0

Files changed 2

+161 -0

doncafe commented now

Falta detallar cómo agregar *ramas* al repo y cómo fusionarlas en la rama principal.
Este paso es importante al momento de incorporar código de otros colaboradores

Agrego docs de git para revisar. Para taller de git482eba7

Reviewers

No reviews

Assignees

No one—assign yourself

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

Successfully merging this pull request may close these issues.

None yet

Notifications

Customize

Unsubscribe

You're receiving notifications because you're watching this repository.

Add more commits by pushing to the `dev` branch on `doncafe/courses`.

Continuous integration has not been set up

GitHub Actions and several other apps can be used to automatically catch bugs and enforce style.

This branch has no conflicts with the base branch

Merging can be performed automatically.

Merge pull request

or view command line instructions.

Write

Preview

H B I

Leave a comment

Attach files by dragging & dropping, selecting or pasting them.

Add more commits by pushing to the `dev` branch on `doncafe/courses`.

Continuous integration has not been set up

GitHub Actions and several other apps can be used to automatically catch bugs and enforce style.

This branch has no conflicts with the base branch

Merging can be performed automatically.

Merge pull request

or view command line instructions.

Merging via command line

If you do not want to use the merge button or an automatic merge cannot be performed, you can perform a manual merge on the command line. However, the following steps are not applicable if the base branch is protected.

HTTPS

SSH

Patch

git@github.com:doncafe/courses.git

Step 1: Clone the repository or update your local repository with the latest changes.

```
git pull origin main
```

Step 2: Switch to the base branch of the pull request.

```
git checkout main
```

Step 3: Merge the head branch into the base branch.

```
git merge dev
```

Step 4: Push the changes.

```
git push -u origin main
```

Projects

None yet

Milestone

No milestone

Development

Successfully merging this pull request may close these issues.

None yet

Notifications

Unsubscribe

You're receiving notifications because you're watching this repository.

1 participant

Lock conversation

```
sopo@tlaloc ~$ git branch -l
dev
* main
sopo@tlaloc ~$ git merge dev
Merge made by the 'ort' strategy.
 docs/md/git_enable_ssh_publickey.md | 143 +-----
 docs/md/git.md | 18 +-----
 2 files changed, 161 insertions(+)
 create mode 100644 docs/md/git.md
 create mode 100644 docs/md/git_enable_ssh_publickey.md
sopo@tlaloc ~$ git st
stage      status
sopo@tlaloc ~$ git status
On branch main
Your branch is ahead of 'origin/main' by 2 commits.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean
sopo@tlaloc ~$ git push -u origin main
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (2/2), 347 bytes | 347.00 KiB/s, done.
Total 2 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:doncafe/courses.git
 d38ba41..7d3e8a3 main -> main
branch 'main' set up to track 'origin/main'.
sopo@tlaloc ~$
```


0.5 Repaso de comandos principales:

1. `git status`
- 1.1 `git add .` Si se agregaron cambios o ficheros al repositorio local
- 1.2 `git commit -m`
2. `git branch` En cual rama del repositorio se está trabajando. Listar las ramas creadas: `git branch -l`
3. `git checkout -b axel` Creamos la rama correspondiente a la funcionalidad del script, o bien `git branch axel`
4. `git checkout master` Cambiamos a la rama master (si nos encontramos en otra rama.)
5. `git push --set-upstream origin axel` Subimos la rama *axel* al GitLab, o bien: `git push origin axel`
6. `git pull` Bajamos todos los cambios del repositorio remoto al repositorio local.
7. `git fetch origin` Traer los cambios del repositorio remoto.
8. `git merge origin/master`
9. `git push origin master`
10. `git branch -m axel axel2` Cambiarle el nombre de la rama de *axel* a *axel2*.
11. Eliminar una rama: `git branch -d axel2`
12. Eliminar una rama en repositorio remoto: `git push origin --delete axel2`

0.6 Ligas y docs para aprender Git

- <https://git-scm.com/book/en/v2/>
- <https://git-scm.com/docs>
- <http://rogerdudler.github.io/git-guide/>