FRP Workshop:
Git

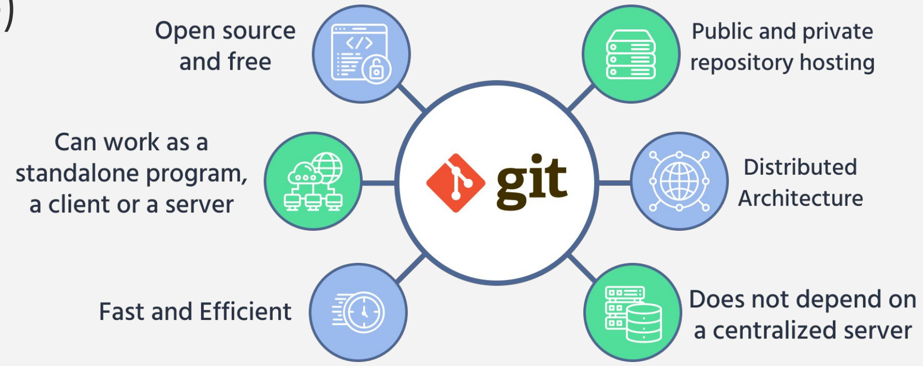
Ignacio Dassori W.
27th of May, 2025

Schedule

- What is git?
- Working with the terminal
- Creating a git repository
- Setting up ssh keys
- Using git
  - Pushing, pulling, branching, merging
  - More functionalities...
- Hands-on exercise

What is git? Why use it?

- Version control system
- Repository hosting (Github/Gitlab)
- History of changes
- Branching for experimentation
- Collaborate on a codebase
- Revert if needed
- Created by Linus Torvalds


Open source and free
Public and private repository hosting
Can work as a standalone program, a client or a server
git
Distributed Architecture
Fast and Efficient
Does not depend on a centralized server

Hosting Projects

GitHub

GitLab

Bitbucket

Working with the Terminal

CTRL + ALT + T : Open a new terminal

CTRL + SHIFT + W : Close terminal

$ cd <folder>: “Change Directory”, moves to specified directory

$ ls : “List”, lists the contents of the directory

$ touch <file> : Creates a file (doesn’t have to be text file)

$ mkdir <name> : Creates a folder

Working with the Terminal

$ history : Print history of commands

$ … | grep <text> : Only prints if contains <text>

$ cp <src> <dest> : Copies file to another dir

$ mv <src> <dest> : Moves file to another dir

$ rm <file> : Deletes a file

$ rm -r <dir> : Deletes a folder and its contents

Text editors

Nano: $ nano <file>

Vim: $ vim <file>

VSCode: $ code .

Creating a Git Repository

GitHub

DEMO TIME!

Setting up ssh keys

1. **Generate SSH Key**
   a. ssh-keygen -t ed25519 -C "your_email@example.com"
2. **Add key to SSH Agent**
   a. eval "$(ssh-agent -s)"
   b. ssh-add ~/.ssh/id_ed25519
3. **Copy public key to Github**

If originally cloned with HTTPS: git remote set-url origin git@github.com:IgnacioDassori/test_repository.git

Basic Git Commands

$ git status : Shows the current state of working directory

$ git add : Stages changes for commit

$ git commit -m “<message>” : Creates snapshot

$ git push : Pushes commit to remote repository

$ git diff : Shows differences between your code and remote

$ git restore : Revert local changes or unstage files

Use .gitignore

This file allows you to specify stuff you don’t want pushed!

Some examples are:

- .vscode
- .venv
- _pycache/
- outputs

Creating a Branch

Make use of branches to experiment

git branch : Shows current branch

git checkout -b <new_branch> : Creates new branch & switches to it

Undoing a commit

Different types depending on what you want to do:

Soft: Undo last commit, keep changes staged.

```
$ git reset --soft HEAD~1
```

Mixed: Undo last commit, keep changes locally.

```
$ git reset --mixed HEAD~1
```

Hard: Undo last commit, drop changes

```
$ git reset --hard HEAD~1
```

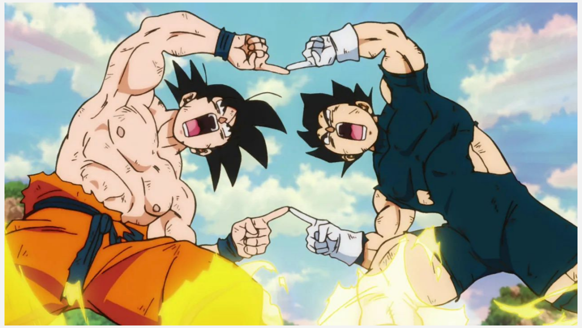
Merging & Pull(Merge) Requests

Combine two branches into one!

Locally (first checkout to the branch receiving changes):

```
$ git merge <feature_branch>
```

This will create a commit with the merged code.



Remotely through Github's Pull Request:

Create a pull request and review it online!

After finishing, delete merged branch! `$ git branch -d <feature_branch>`

Merge Conflicts

A merge doesn’t always go smoothly

-> If changes were made to the origin after branching

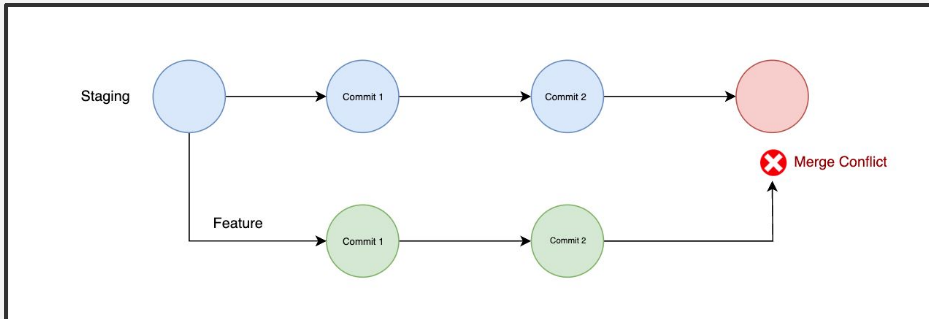
-> Merging will result on a conflict that must be resolved

-> Which changes to keep?

2 recommended ways to solve merge conflicts:

-> **VSCode** Git Lense extension

-> **Github** interface



FRP Workshop:

Git

Activity!

Merge Conflicts

This activity is done in pairs! Instructions are on the workshop repository!

1. Choose person A & B
2. Follow the instructions
3. Implement basic code
4. Take turns in solving merge conflicts
5.

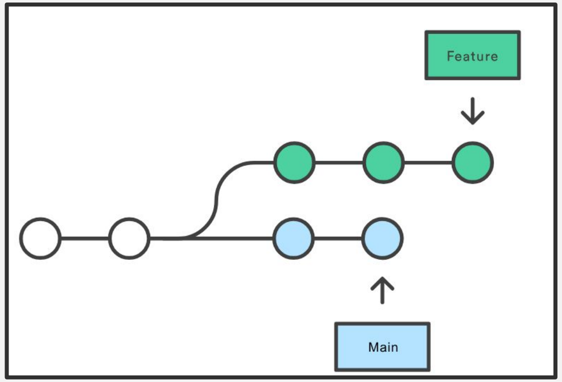
More advanced git

It can always go deeper

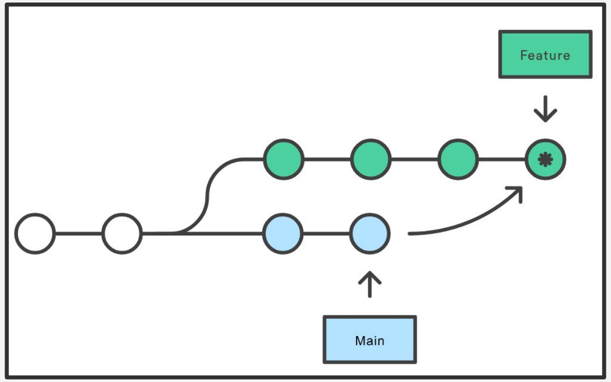
- Rebase vs Merge
- Cherry Picking
- Squash Commits
- Git LFS
- Pruning
- Submodules
- ....

Git Rebase

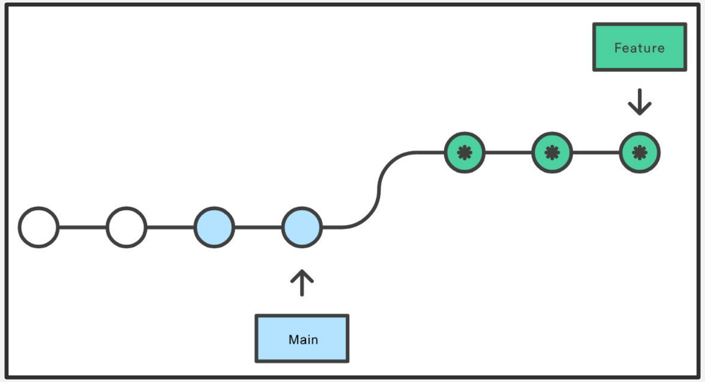
Rewrites commit history: Rebase changes the “base” of your branch



Divergence



Merge



Rebase

From feature branch: $ git rebase main

Git Stash

Save a snapshot without commiting

Sometimes you need to change branch, but you don’t want to loose the changes on your feature branch. -> Commit? Not always wanted.

$ git stash	: Saves a stash of current state
$ git stash pop	: Applies last stash and deletes it
$ git stash apply	: Applies last stash and keeps it
$ git stash list	: Shows all stashes

FRP Workshop: **Git**

Ignacio Dassori W.
27th of May, 2025