Escuela de Programación para Investigación Científica EPIC-II

Grupo de Investigadores Latitud Cero & Departamento de Física de la Escuela Politécnica Nacional

Luis Manzanillas

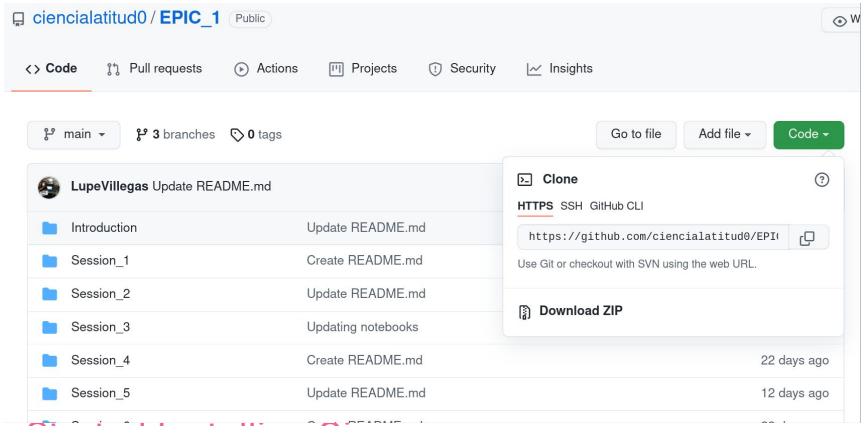
Introducción a git y github

Git is an Open Source Distributed Version Control System for tracking changes in source code or any other set of files

- **Control System:** Git is a content tracker.
- **Version Control System:** The code which is stored in Git keeps changing as more code is added. Many developers can add code in parallel. Keeps history of what changes have happened. Also, Git provides features like branches and merges.
- **Distributed Version Control System:** Git has a **remote repository** which is stored in a server and a **local repository** which is stored in the computer of each developer.

GitHub: *company that provides hosting for software development version control using Git*

Using git



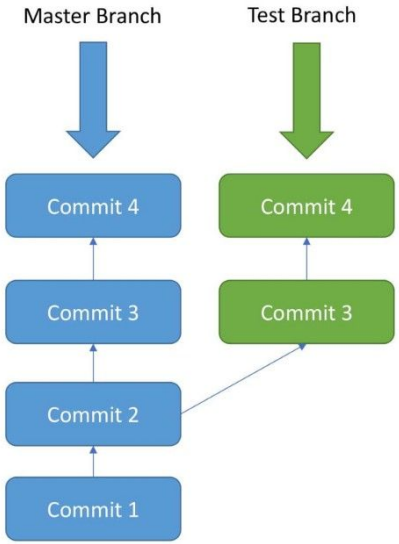
- Install: Ubuntu

  #sudo apt-get install git

  others: https://git-scm.com/book/en/v2/Getting-Started-Installing-Gi

- Download EPIC-1 material
  - Go to https://github.com/ciencialatitud0/EPIC_2
  - Open a terminal, create a directory, then inside the directory:
    - git clone https://github.com/ciencialatitud0/EPIC_2.git ./
  - Now you have all the material in your own computer
  - Updating the repository:
    - git pull
  - Adding/modifying:
    - git add .
  - Commit:
    - git commit -m “my clear message indicating the changes”
  - Pushing the changes to the remote repository:
    - git push
  - All this commands should work with your own repository or in any repository in which you are contributor

Using git: branches




```
$ git status                          <-- Check status of repository
On branch master
Your branch is up-to-date with 'origin/master'.

nothing to commit, working directory clean
$ git branch                          <-- Check available branches
* master
$ git branch test                     <-- Create new branch named 'test'
$ git branch
* master
  test
$ git checkout test                   <-- Switch branch
Switched to branch 'test'
$ git branch
  master
* test                                <-- Active branch changed
```

To merge master branch with test branch:

git checkout master

git merge test