



INTRODUCCIÓN A GIT Y GITHUB

Semana Mecatrónica FIUADY
Br. Alejandro Santoscoy Rivero



¿QUÉ ES GIT?

- Git es un sistema de control de versiones
- Open source
- Command line
- Permite el trabajo remoto con otros colaboradores



¿QUÉ ES GITHUB?

- **GitHub no es Git**
- Es una página web
- Aquí se pueden encontrar los repositorios que la gente sube

ERRORES COMUNES



new.**psd**



newfinal.**psd**



newfinalfinal.**psd**



newfinalestfinal.**psd**

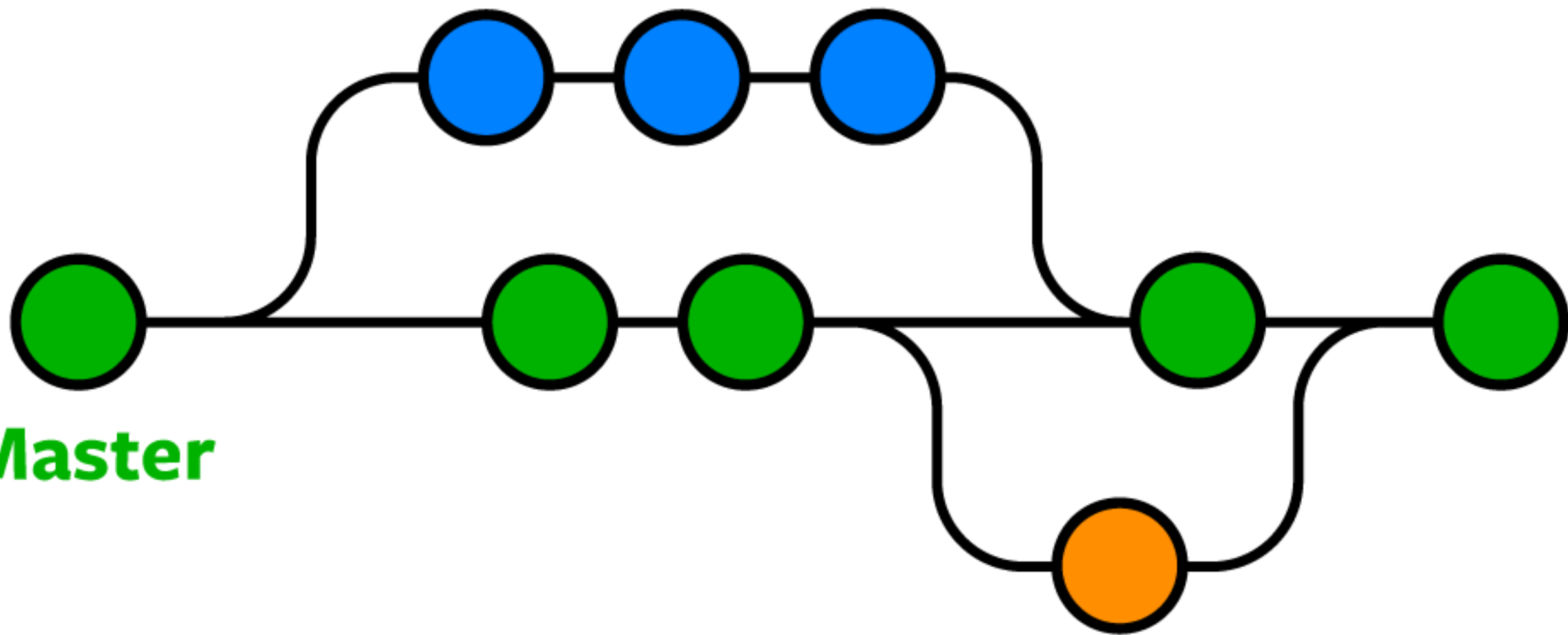


newfinalestfinal
forsure.**psd**



newfinalestf...kthis
s...itfinal.**psd**

Your Work



Master

Someone Else's Work

GITHUB

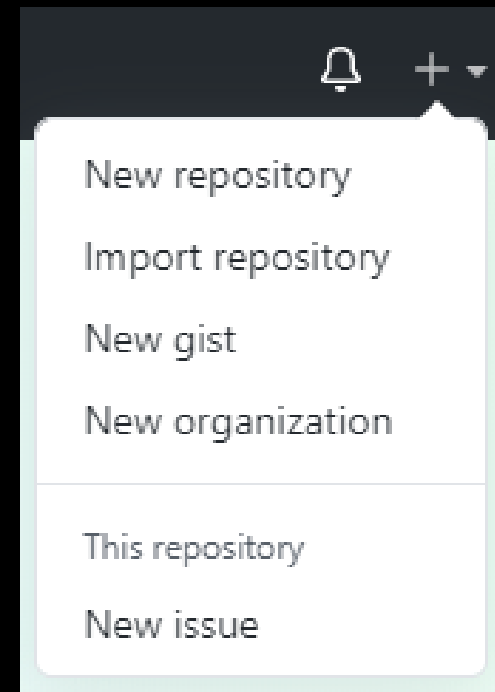
- Primero crear una cuenta (de preferencia institucional, trae beneficios)

Join GitHub

Create your account

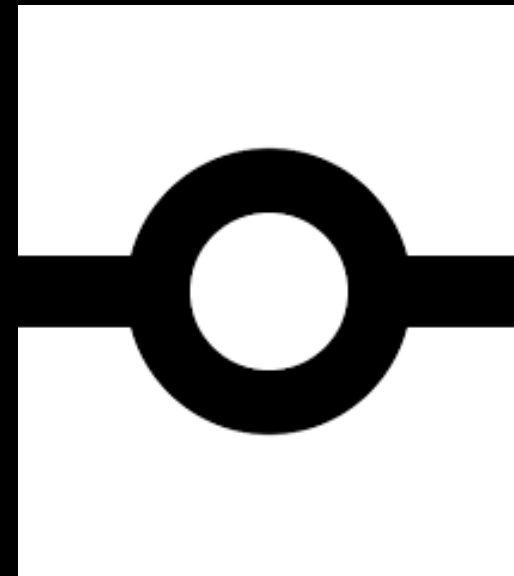
CREAR REPOSITORIO

- Un repositorio “repo” es como un proyecto, se pueden alojar archivos y carpetas



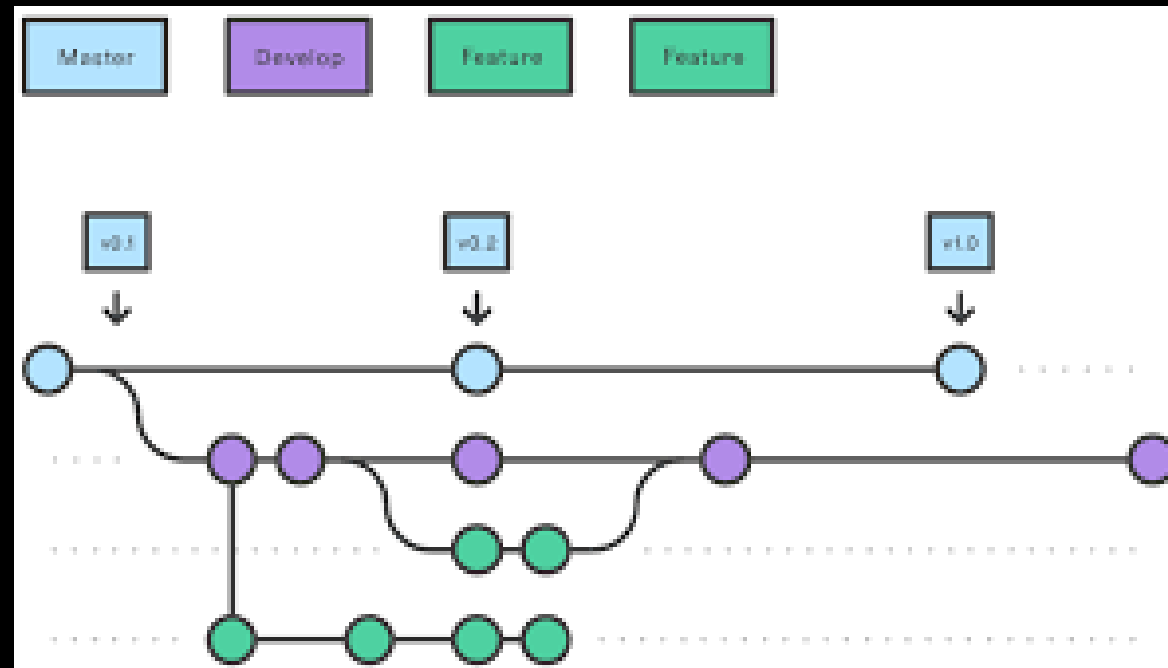
COMMIT

- Un commit es como una aportación al proyecto
- Tiene un ID y es el registro de cada punto de cambio



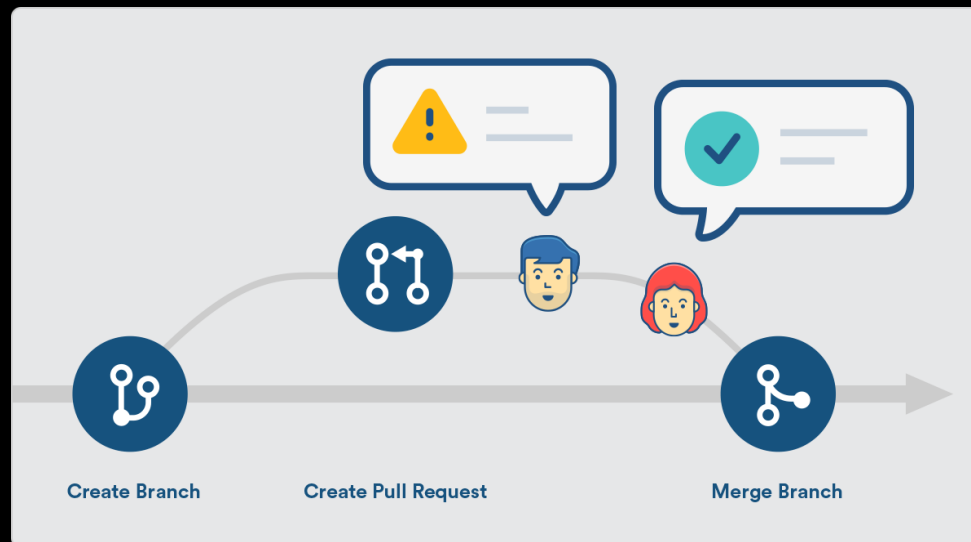
BRANCH

- Trabajar paralelamente



PULL REQUEST

- Combinar el avance de una rama a la original
- Lo que hace en realidad es un merge (después hablamos mejor de esto)



FORK

- Permite tener un repositorio idéntico de otra persona y hacer tus propios cambios
- Se pueden unir con un Pull Request
- NO LO CLONA (bueno, si pero no)

LINUX

- Git utiliza arquitectura Linux
- Es conveniente saber los comandos básicos de Linux como:
- **ls** (enlistar todos los archivos de un directorio – “**list**”)
- **mkdir** (crear directorio – “**make directory**”)
- **cd** (cambiar directorio – “**change directory**”)
- **touch** (crear un archivo)
- **rm** (remover o borrar un archivo – “**remove**”)
- Iremos viendo otros comandos básicos



NOTA DE CONFIGURACIÓN

- `git config --global user.name "TuNombre"`
- `git config --global user.email you@example.com`
- `git config --list`

COMANDOS DE GIT

- **git init** – “Initialize” Inicializar un control de versiones (git)
- **git status** – Comprobar estado del sistema
- **git add** – Empezar a seguir un archivo (“track”) **git add *.txt**
git add -A
- **git commit -m “commitear”** contribuir al sistema
archivos trackeados



IGNORAR
ARCHIVOS

.gitignore

HISTORIAL DE VERSIONES

- **git log** – Revisar los commits del proyecto
- **git checkout** – Regresar a una versión anterior del proyecto
- **git --no-commit 1234abcde1df2...HEAD** – Hacer que regresar sea un cambio

BRANCHES (RAMAS)

- git branch
- git checkout -b newBranch
- git checkout master
- git checkout -b newBranch existingBranch
- git branch -d existingBranch
- git merge

RESOLVER CONFLICTOS

```
$ git merge ramaError
CONFLICT (add/add): Merge conflict in errorFile.txt
Auto-merging errorFile.txt
Automatic merge failed; fix conflicts and then commit the result.
```

If you have questions, please

<<<<<< HEAD

open an issue

=====

ask your question in IRC.

>>>>>> branch-a

- Resolverlo con cualquier editor de texto decente
- Solo hay que borrar
- <<<<<<<, =====, >>>>>>>

GIT Y GITHUB REMOTO

- `git remote add origin SSH_HTTP_url`
- `git clone SSH_HTTP_url`
- `git pull origin master`
- `git push origin master`



EXTRAS

GITHUB PAGES

- Sitios web gratis en GitHub!!!
- Crear branch **gh-pages**
- Hacerla default
- Eliminar **master**
- Crear **index.html**