Taller de Python Científico

Introducción a 

Franco N. Bellomo - fnbellomo@gmail.com

¿Necesito control de versiones?

Comenzemos un nuevo proyecto

GenialProyecto

¡Todo bien! (por ahora)

¿Necesito control de versiones?

El proyecto va crecien y necesitamos hacer algunos "back-ups"

Copy
GenialProyecto

Copy (1)
GenialProyecto

GenialProyecto

Ya comienza a crecer y a complicarse los directorios

¿Necesito control de versiones?

Ya no nos acordamo que modificación realizamos en cada copia y les ponemos nombre

Copy
GenialProyecto

GenialProyecto
Optimizacion integracion

Copy (1)
GenialProyecto

GenialProyecto
Modificacion API

GenialProyecto

GenialProyecto
Arreglo bug

No parace algo muy simple de mantener...
Crece rapidamnente el tamaño de disco usado

¿Necesito control de versiones?

¿Que pasa si trabajo en la pc del trabajo y quiero sincronizar con mi notebook?

* Usamos un pendrive (copia completa de todo el proyecto)
* DropBox (no puede escalar en los integrantes)
* Trabajamos en una sola computadora

¿Necesito control de versiones?

Y ahora un colega quiere coloborar en el desarrollo de nuestro software

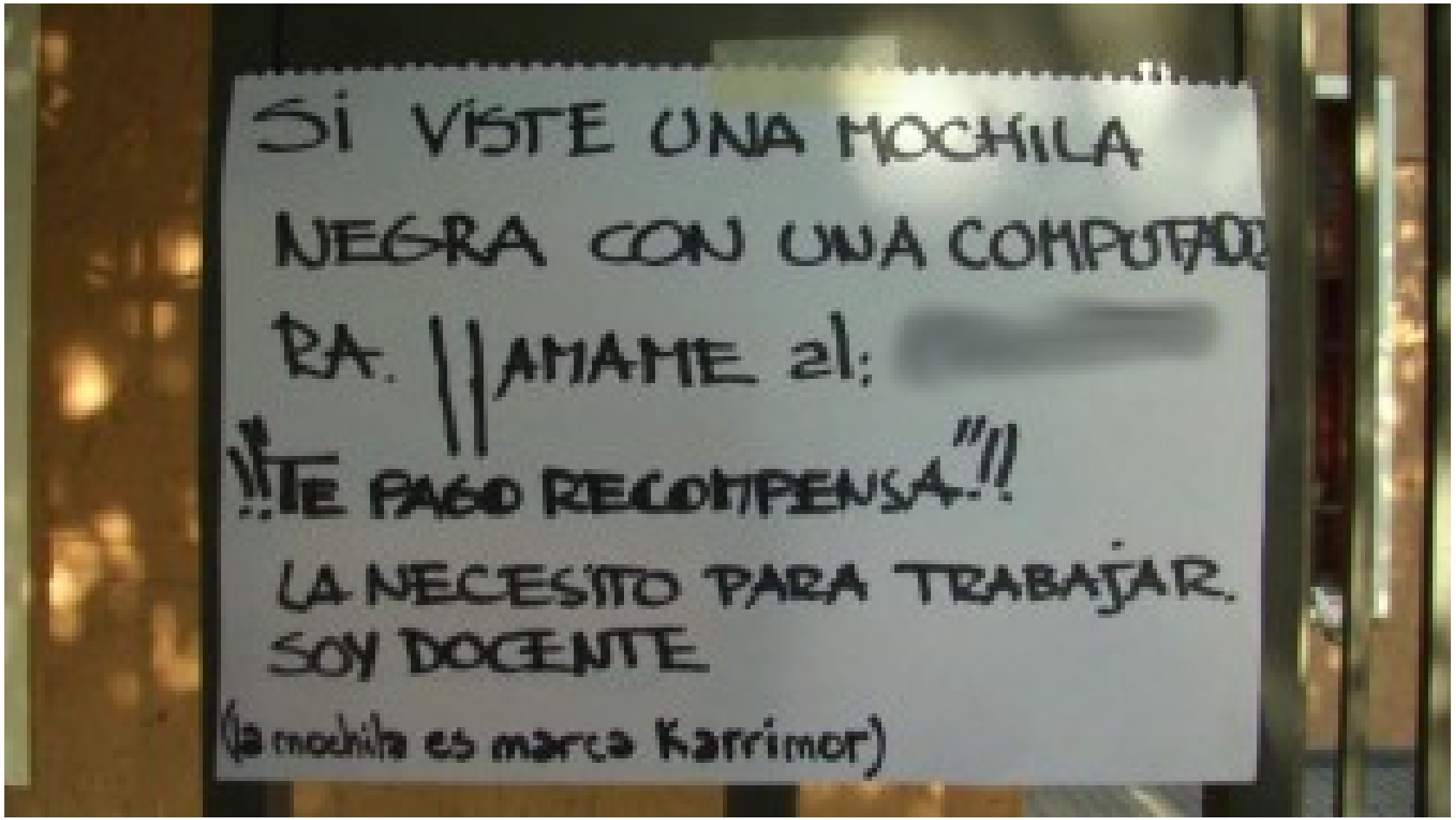
From: <JonSnow@todaviaNoSeGit.com>
To: Ned <NedStark@todaviaNoSeGit.com>
Subject: Trabajo

Ned, acá te mando el zip con la última versión (creo) con mi parte del trabajo.

From: <NedStark@todaviaNoSeGit.com>
To: Jon <JonSnow@todaviaNoSeGit.com>
Subject: Re: Trabajo

Jon, me olvidé de avisarte que yo ya había modificado el integrador numérico.

¿Necesito control de versiones?



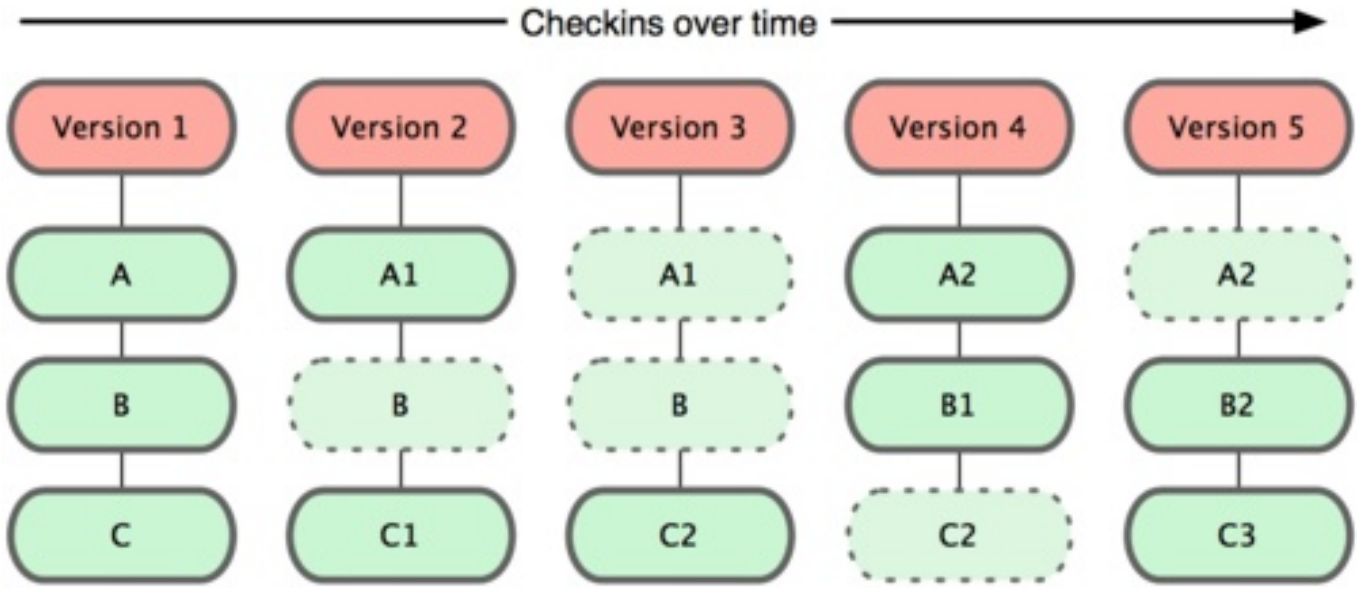
Ha perdido dos años de trabajo...

¿Qué es Git?

Es un software de control de versiones diseñado por Linus Torvalds, pensando en la eficiencia y la confiabilidad del mantenimiento de versiones de aplicaciones cuando estas tienen un gran número de archivos de código fuente.

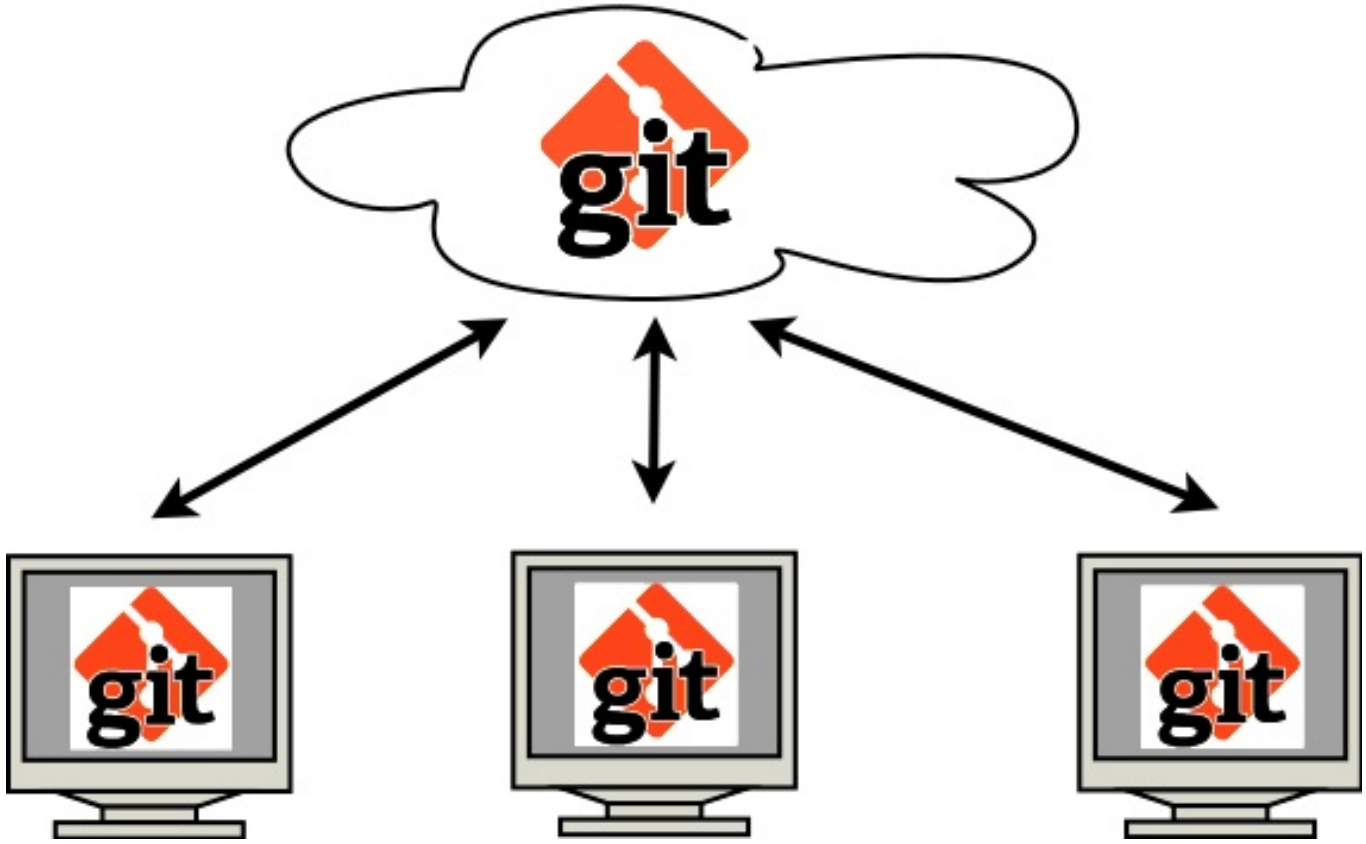
Algunas caracteristicas de Git

* Backup de snapshots del proyecto (de cualquier tipo de archivo).
* Mensaje de las modificaciones.
* Multiples autores.
* Multiples ramas de trabajo.



Algunas caracteristicas de Git

* Distribuido
  * Todos los usuarios tienen la historia completa.
  * Multiples servidores



Algunas caracteristicas de Git

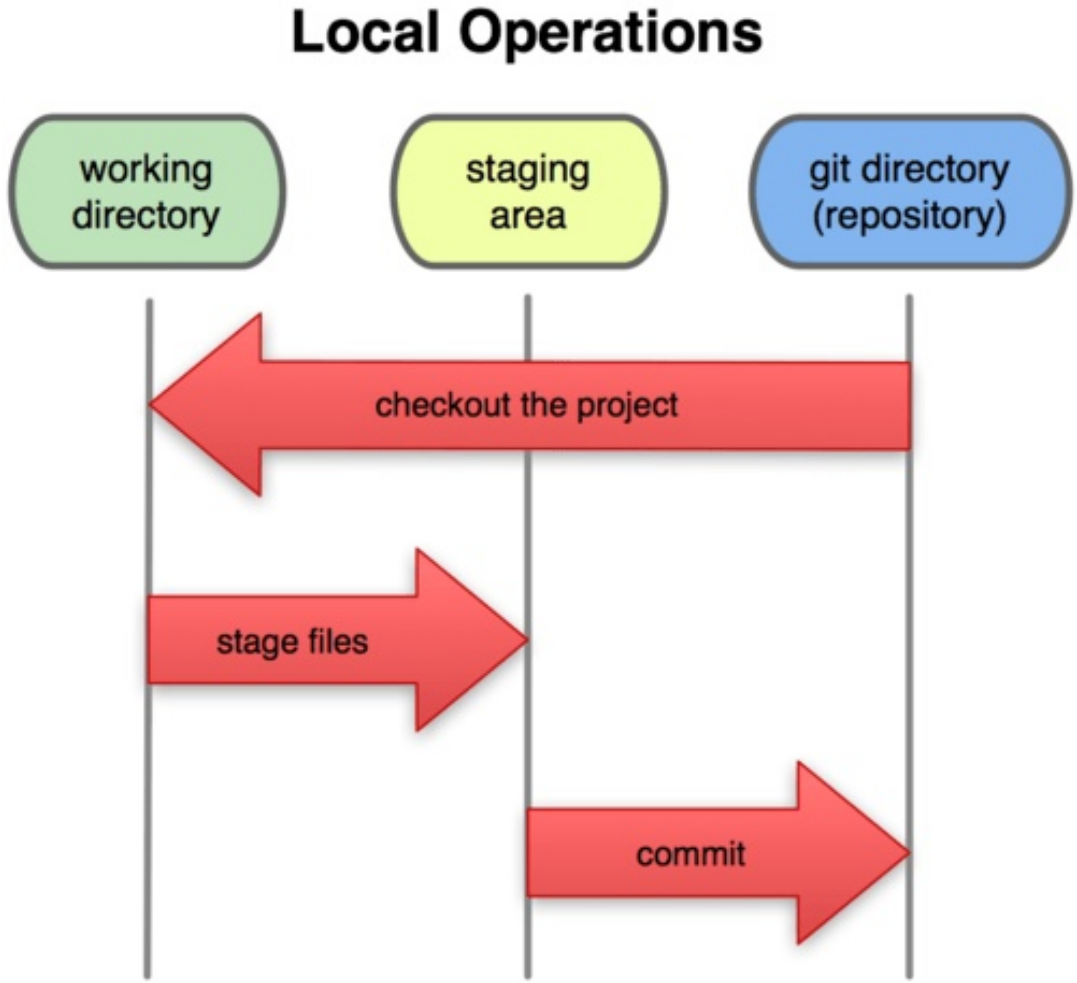
* Eficiente forma de almacenar la historia.
* Log de modificaciones.
* Rápido.

```
commit 662b951963ce3258b15ad738df6c578ef60b3de2
Author: Franco N. Bellomo <fnbellomo@gmail.com>
Date:   Fri Jan 22 20:10:16 2016 -0300

    update notebook

commit 881901f134ba2a424ca313401be21df888bcd649
Author: Franco N. Bellomo <fnbellomo@gmail.com>
Date:   Fri Jan 22 20:08:32 2016 -0300

    calculo de tau con trampas

commit 16835fce3456c55e090e6f43dd5e457584fcbad3
Author: Lucas Esequiel Bellomo <lbellomo@gmail.com>
Date:   Thu Dec 10 01:55:08 2015 -0300

    add notebook 9 and 10

commit cf830329bc65c1063959cdee68864c2a9f4f060d
Author: Lucas Esequiel Bellomo <lbellomo@gmail.com>
Date:   Thu Dec 10 01:53:01 2015 -0300

    remove data TPP

commit e5190c389cc55a1e626d5c83975e3b64cfa007a5
Author: Lucas Esequiel Bellomo <lbellomo@gmail.com>
Date:   Tue Dec 1 00:47:12 2015 -0300

    add speed grafic
:
```

Los tres estados

Git tiene tres estados principales en los que se pueden encontrar los archivos:

* Confirmado (committed)
* Modificado (modified)
* Preparado (staged)



Comencemos

Primero, configuremos git

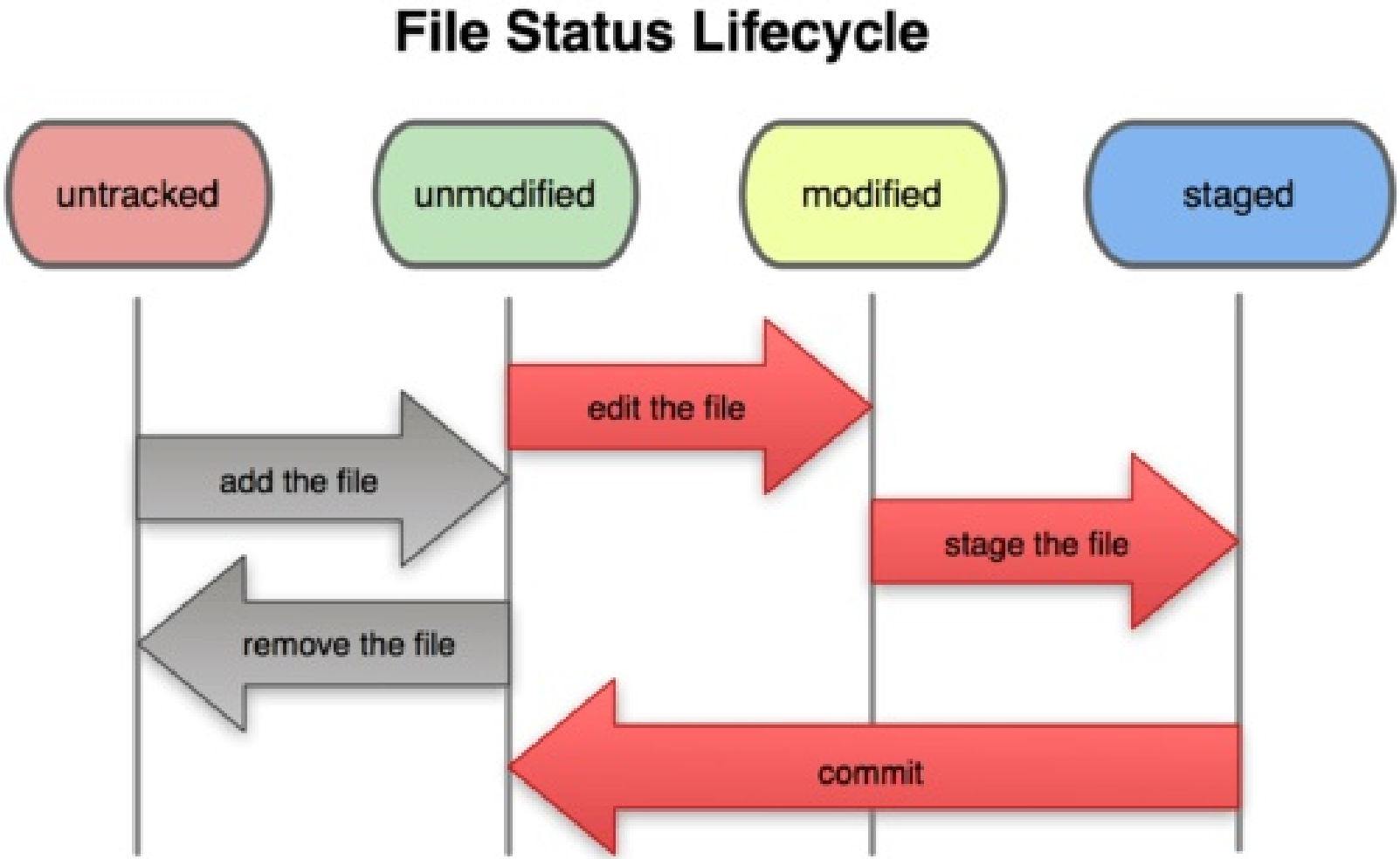
```
$ git config --global user.name "Jon Snow"
$ git config --global user.email jonsnow@example.com
$ git config --global core.editor emacs
```

La ardua tarea de crear un repo

```
$ mkdir newProyect
$ cd newProyect
$ git init
```

```
$ git status
On branch master
Initial commit
nothing to commit (create/copy files and use "git add" to track)
```

Estado de los archivos



Agregando archivos

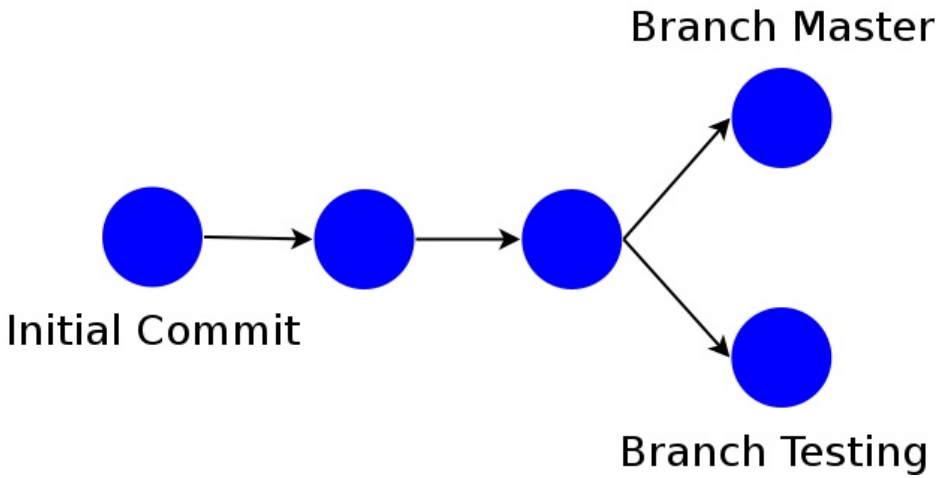
```
$ emacs README.md
$ git add README.md
$ git commit -m 'initial commit'
[master (root-commit) a7d19cd] initial commit
1 file changed, 1 insertion(+)
 create mode 100644 README.md
```

```
$ emacs wind.py
$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
     wind.py
nothing added to commit but untracked files present (use "git
add" to track)
```

Branch

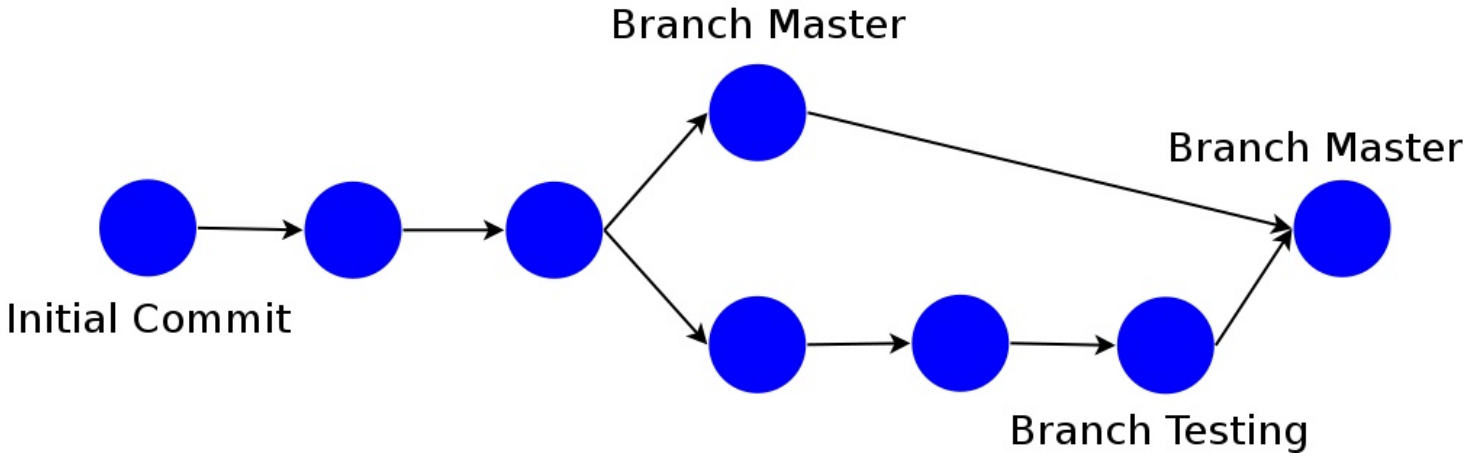
Nuestro repositorio ha crecido bastante y queremos probar algunas modificaciones al código, pero no queremos romper lo que ya esta andando.

```
$ git checkout -b test
Switched to a new branch 'test'
$ ls
README.md  wind.py
```



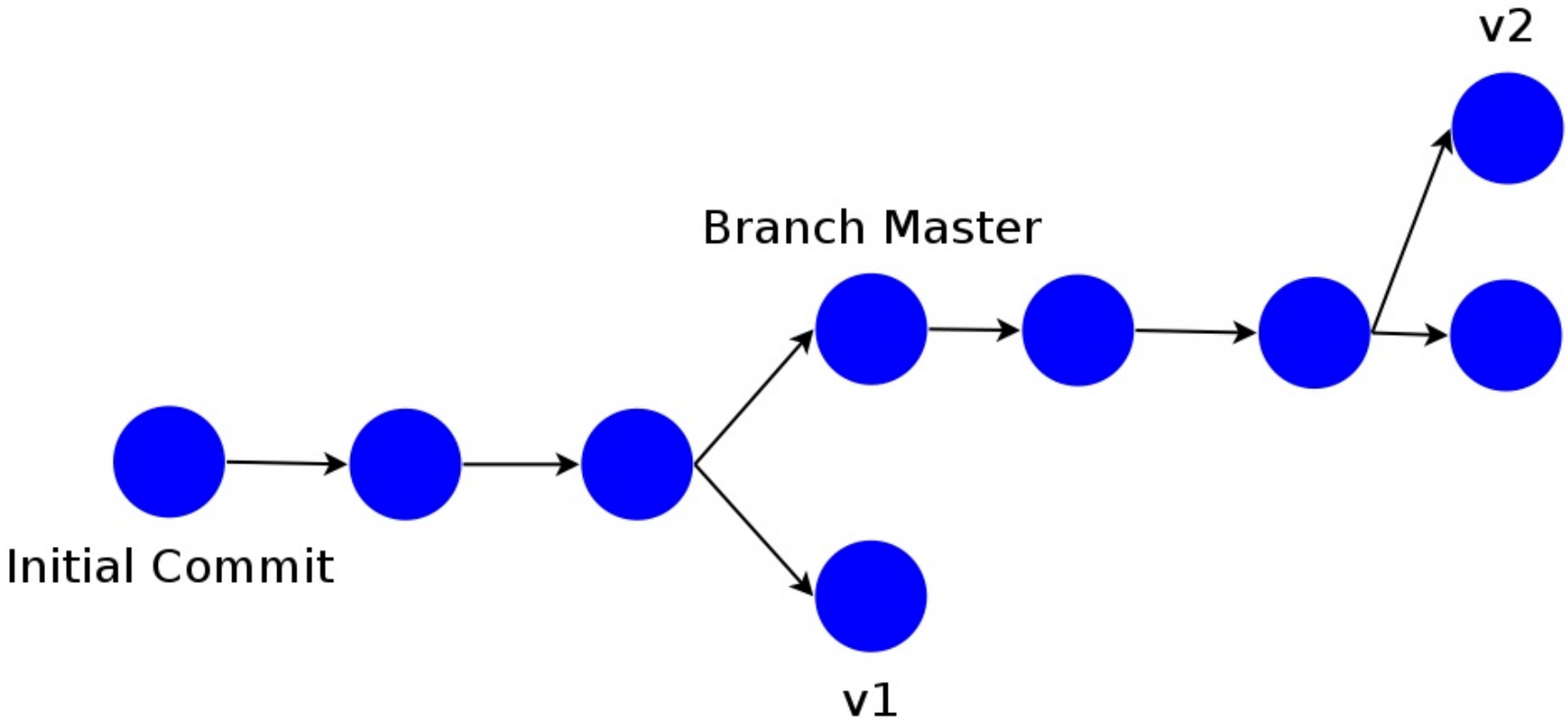
Merge

```
$ git checkout master
Switched to branch 'master'
$ git merge test
```

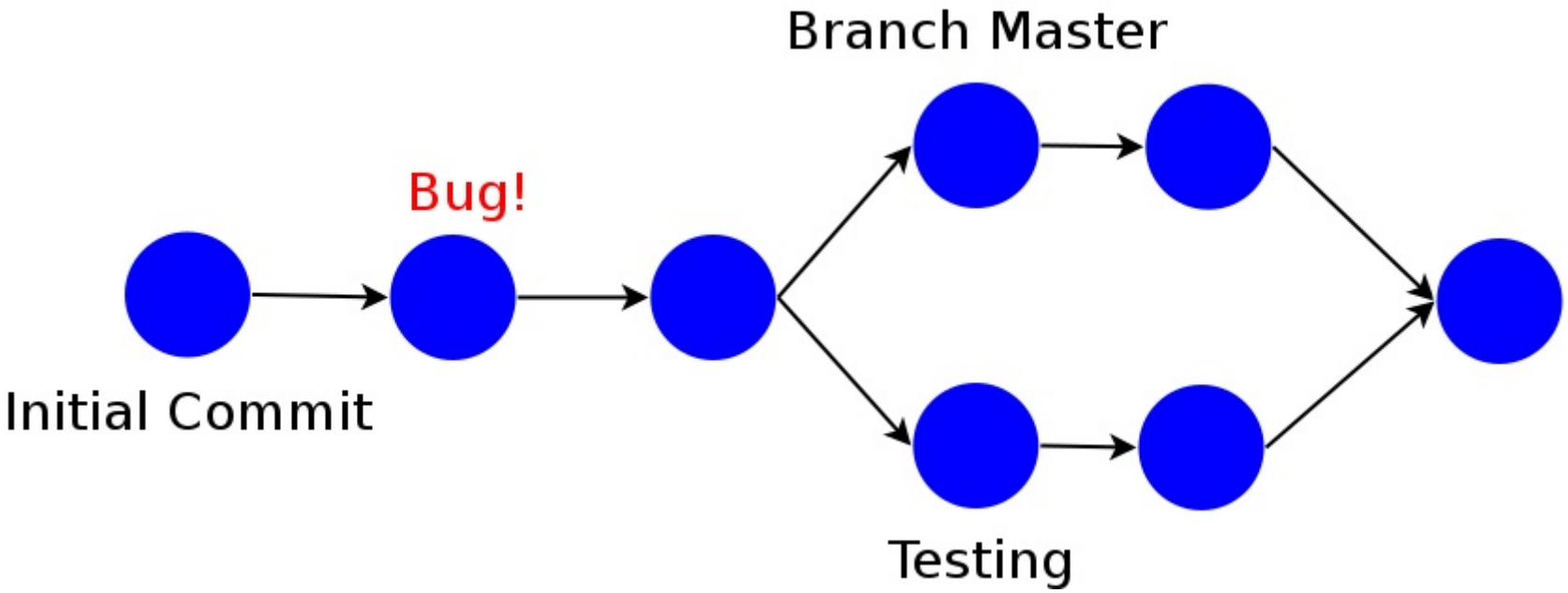


Buena Practica

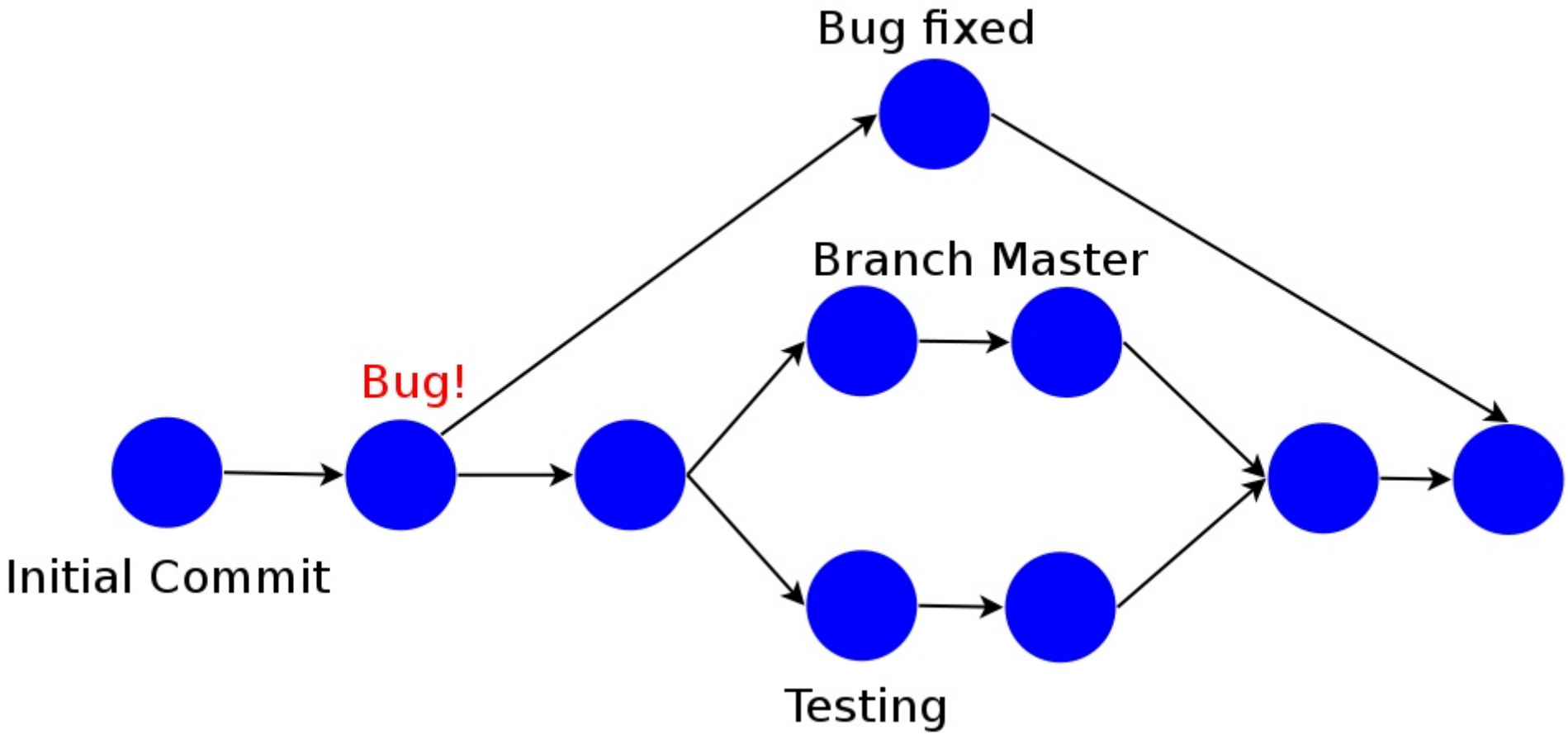
Es una buena practica dejar las versiones release del código de un branch propio.



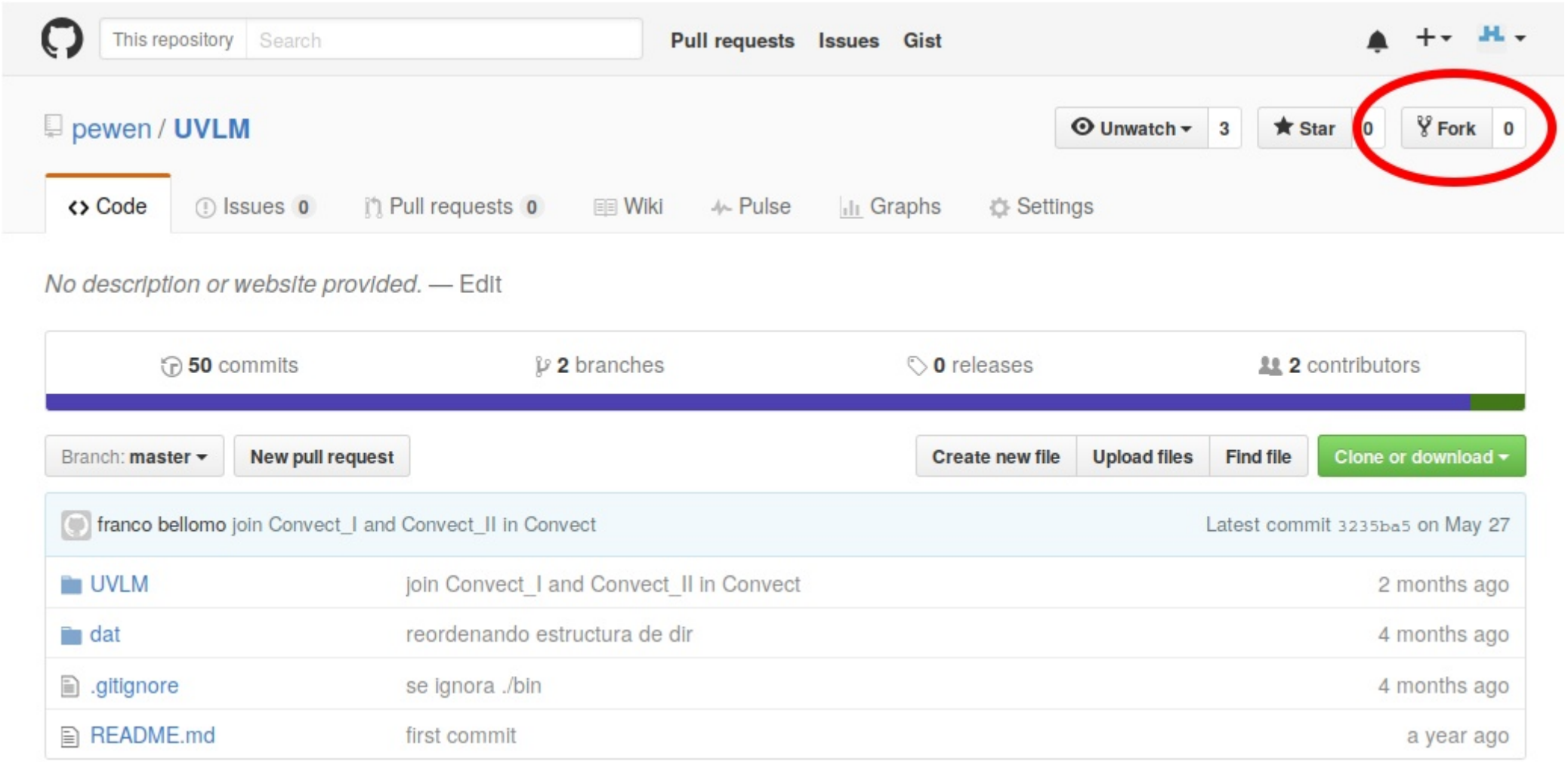
Arreglando la historia



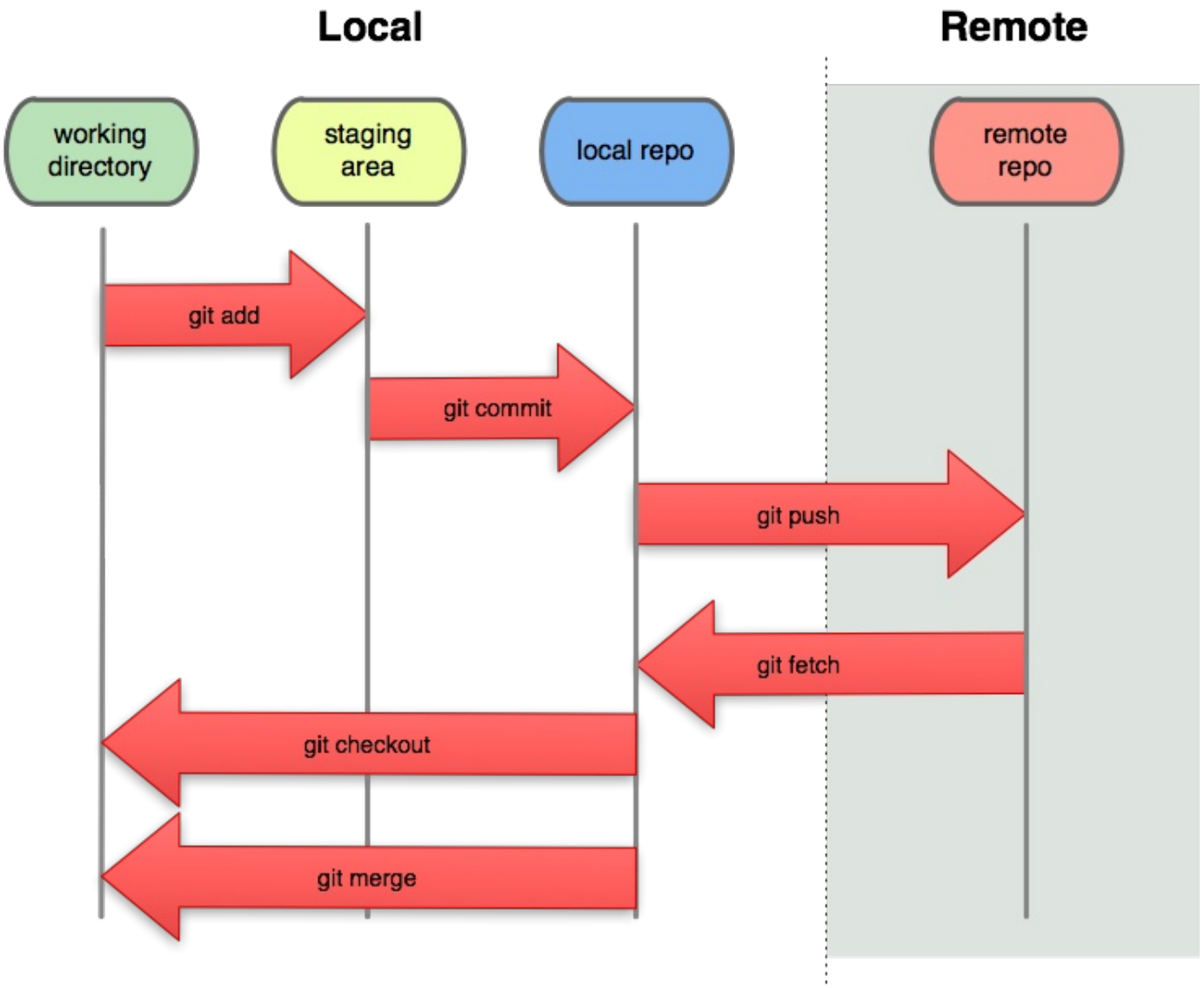
Arreglando la historia



Fork


This repository
Search
Pull requests
Issues
Gist
pewen / UVLM
Unwatch 3
Star 0
Fork 0
Code
Issues 0
Pull requests 0
Wiki
Pulse
Graphs
Settings
No description or website provided. — Edit
50 commits
2 branches
0 releases
2 contributors
Branch: master
New pull request
Create new file
Upload files
Find file
Clone or download
franco bellomo join Convect_I and Convect_II in Convect
Latest commit 3235ba5 on May 27
UVLM
join Convect_I and Convect_II in Convect
2 months ago
dat
reordenando estructura de dir
4 months ago
.gitignore
se ignora ./bin
4 months ago
README.md
first commit
a year ago

Trabajando remotamente



Comandos

```
$ git clone https://github.com/numpy/numpy.gitgit

$ git push

$ git pull
```

Referencias

Documentación oficial de Git

https://git-scm.com/book/es/v1

Servicios gratuitos

https://github.com

https://gitlab.com

https://bitbucket.org