

---

# GIT - Getting It Together!

- Campus Party Argentina -  
31/Oct/2024

---



Pablo Carboni

Camila Fernández  
“Doña Assembler”





UTNiana, DevOps Engineer, fan  
de volver al futuro ✨

Miembro de @kcd.argentina!  
◎ ✨: @doniaassembler

Unix, DNS, y otras nerdeadas.  
Miembro de @sysarmy!

XM: @pcarboni

Mastodon:@pcarboni@bsd.network



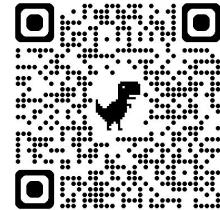
**¿Qué es el versionado de código?  
¿Para qué me sirve?  
¿Y GIT?**

# Agenda

- Breve historia versioning e Implementaciones Pre-GIT
- Conceptos
- Comandos básicos
- Servicios (Github, Gitlab)
- Caso curioso: TIG
- Workshop - Hands On !

A BRIEF HISTORY OF

# VERSION CONTROL



## Early 1960s

IEBUPDTE for IBM OS/360  
(punched card system)

## 1982

RCS (Revision Control System) created  
by Walter Tichy

## 2000

Subversion (SVN) created by  
CollabNet

## 1972

SCCS (Source Code Control  
System) created by Marc  
Rochkind

## 1986

CVS (Concurrent Versions Systems)  
created by Dick Grune

## 2005

Git created by Linus Torvalds

<https://blog.tarynmcmillan.com/a-history-of-version-control>

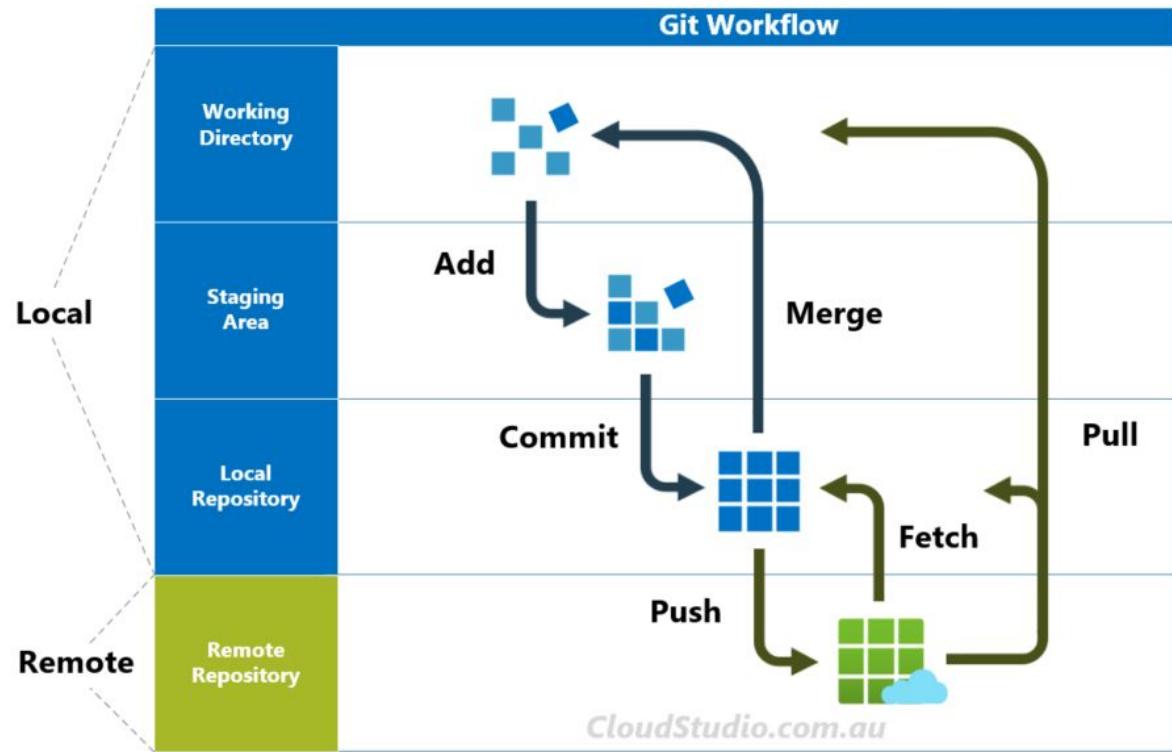
CREATED BY TARYN MCMILLAN

# Sistemas de versioning

| Nombre   | Año  | Features   |
|--|------|--|
| IEBUPDTE - IBM - <a href="https://ibmmainframes.com/about145.html">https://ibmmainframes.com/about145.html</a>   | 1962 | IBM OS/360 - <i>Tarjetas perforadas</i><br><b>Crea y actualiza código de librerías.</b>  |
| SCCS (Source Code Control System) - Bell Labs - Marc Rochkind.<br><a href="http://sccs.sourceforge.net">http://sccs.sourceforge.net</a>  | 1972 | Un solo usuario:<br>✓ crear, editar, <b>trackear</b><br>✓ un archivo por versión (⌚)   |
| RCS (Revision Control System) - Walter Tichy.<br><a href="http://et.engr.iupui.edu/~dskim/tutorials/print.php?article=rcs">http://et.engr.iupui.edu/~dskim/tutorials/print.php?article=rcs</a> | 1982 | Implementó <b>Deltas</b> de cambios en archivos.<br>► Más rápido y eficiente.  |
| CVS (Concurrent Versions Systems) - Dick Grune.<br><a href="https://dickgrune.com/Programs/CVS.orig/">https://dickgrune.com/Programs/CVS.orig/</a>   | 1986 | Concurrencia de edición:<br>Modelo cliente  servidor. |
| SVN (Subversion) - CollabNet - <a href="https://subversion.apache.org/">https://subversion.apache.org/</a>   | 2000 | CVCS: Cambios a copia en servidor centralizado (bajado por cada usuario 😐)   |
| GIT - Linus Torvalds - <a href="https://git-scm.com/">https://git-scm.com/</a>   | 2005 | DVCS: Copia de repo e historial (también, bajado por cada usuario 🤝 😎)   |

# Conceptos

- **Repositorio Distribuido**
- **Workflow**
  - Repo Local (Working Dir)
  - INDEX
  - HEAD
- Branch(es)
  - Merge (OMG)
  - Rebase (OMG<sup>2</sup>)
  - Diff
- TAGGING
- LOGs (GIT no miente)
- RollBack
  - Revert
  - Checkout
  - Fetch origin
  - Reset



<https://cloudstudio.com.au/2021/06/26/git-command/>

# Conceptos

- Repositorio Distribuido
- Workflow
  - Repo Local (Working Dir)
  - INDEX
  - HEAD
- **Branch(es)**
  - Merge (OMG)
  - Rebase (OMG<sup>2</sup>)
  - Diff
- TAGGING
- LOGs (GIT no miente)
- RollBack
  - Revert
  - Checkout
  - Fetch origin
  - Reset



<https://lutece.paris.fr/support/wiki/git.html>

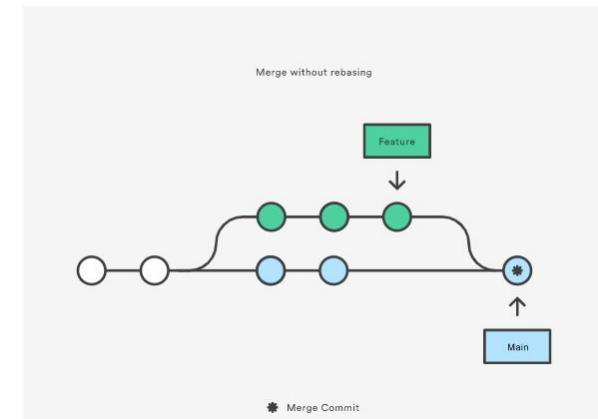
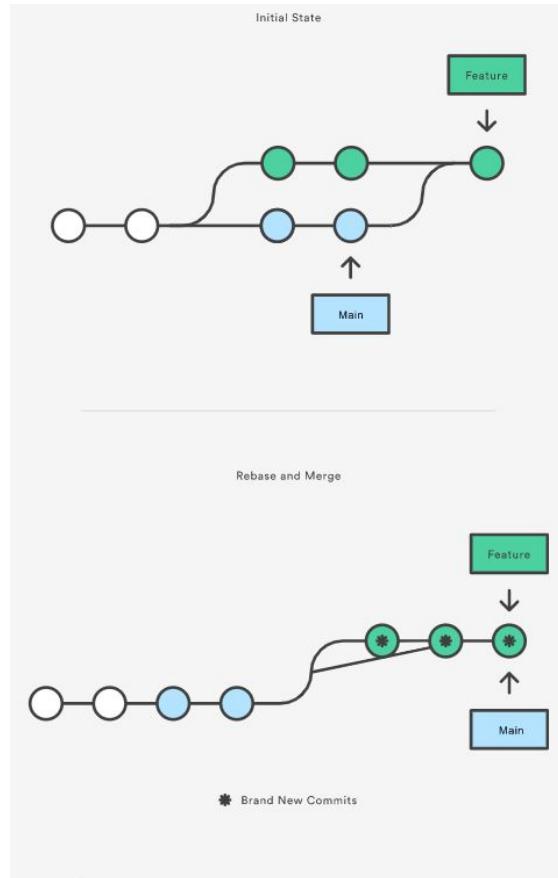
# Conceptos

- Repositorio Distribuido
- Workflow
  - Repo Local (Working Dir)
  - INDEX
  - HEAD
- **Branch(es)**
  - **Merge (OMG)**
  - Rebase (OMG<sup>2</sup>)
  - Diff
- TAGGING
- LOGs (GIT no miente)
- RollBack
  - Revert
  - Checkout
  - Fetch origin
  - Reset



# Conceptos

- Repositorio Distribuido
- Workflow
  - Repo Local (Working Dir)
  - INDEX
  - HEAD
- **Branch(es)**
  - Merge (OMG)
  - **Rebase (OMG<sup>2</sup>)**
  - Diff
- TAGGING
- LOGs (GIT no miente)
- RollBack
  - Revert
  - Checkout
  - Fetch origin
  - Reset

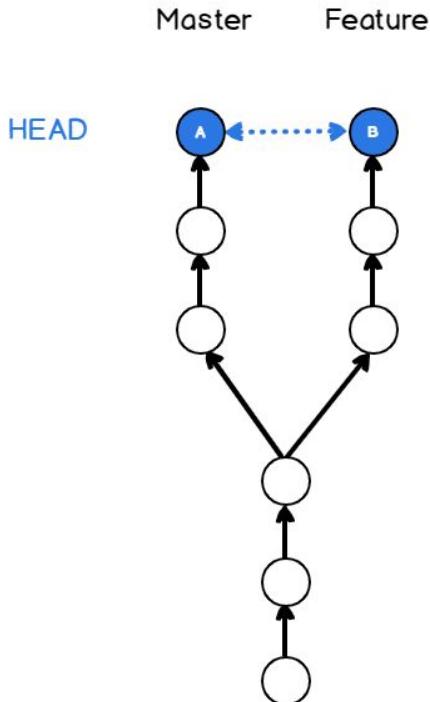


<https://www.atlassian.com/es/git/tutorials/merging-vs-rebasing>

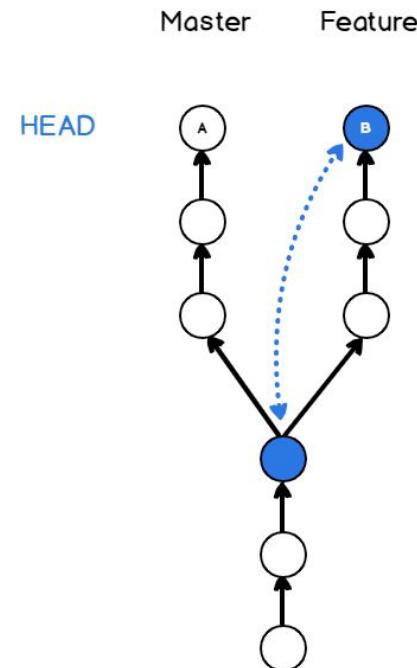
# Conceptos

- Repositorio Distribuido
- Workflow
  - Repo Local (Working Dir)
  - INDEX
  - HEAD
- **Branch(es)**
  - Merge (OMG)
  - Rebase (OMG<sup>2</sup>)
  - **Diff**
- TAGGING
- LOGs (GIT no miente)
- RollBack
  - Revert
  - Checkout
  - Fetch origin
  - Reset

git diff master..feature

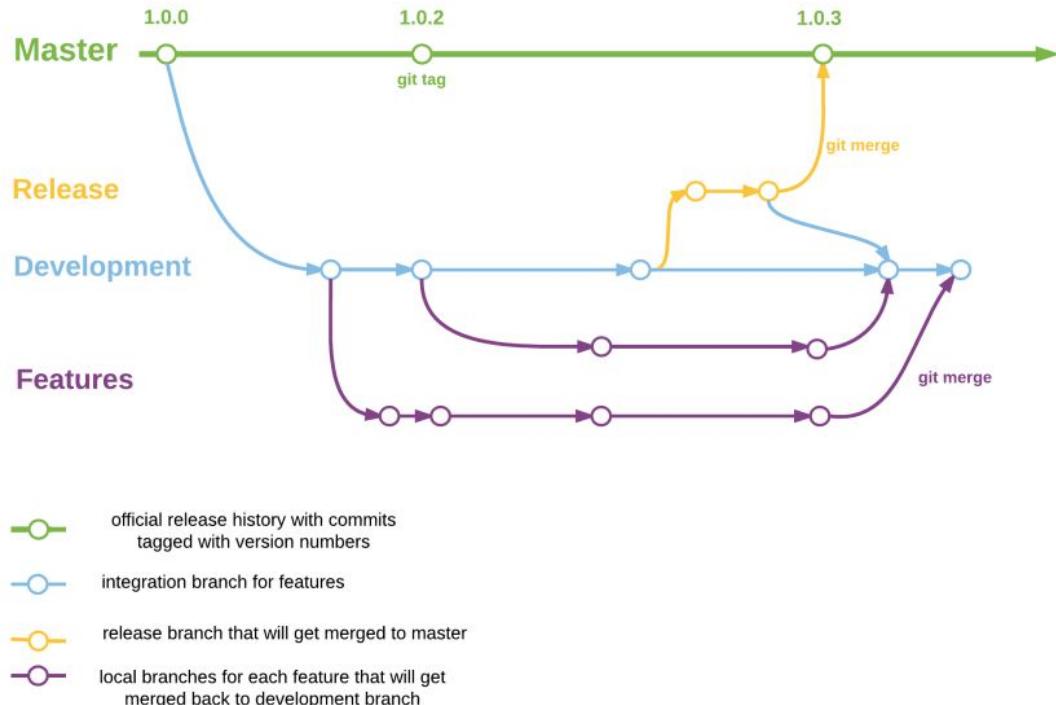


git diff master...feature  
triple dot



# Conceptos

- Repitorio Distribuido
  - Workflow
    - Repo Local (Working Dir)
    - INDEX
    - HEAD
  - Branch(es)
    - Merge (OMG)
    - Rebase (OMG<sup>2</sup>)
    - Diff
  - **TAGGING**
  - LOGs (GIT no miente)
  - RollBack
    - Revert
    - Checkout
    - Fetch origin
    - Reset



<https://dev.to/saurabhpro/git-tags-ftw-1i77>

# Conceptos

- Repositorio Distribuido
- Workflow
  - Repo Local (Working Dir)
  - INDEX
  - HEAD
- Branch(es)
  - Merge (OMG)
  - Rebase (OMG<sup>2</sup>)
  - Diff
- TAGGING
- **LOGs (GIT no miente)**
- Blame (Quien fue?)
- RollBack
  - Revert
  - Checkout
  - Fetch origin
  - Reset



# Conceptos

- Repositorio Distribuido
- Workflow
  - Repo Local (Working Dir)
  - INDEX
  - HEAD
- Branch(es)
  - Merge (OMG)
  - Rebase (OMG<sup>2</sup>)
  - Diff
- TAGGING
- LOGs (GIT no miente)
- Blame (Quien fue?)
- RollBack
  - Revert
  - Checkout
  - Fetch origin
  - Reset

```
$ git log
5bf53ef (HEAD, origin/master, origin/HEAD, master) Normalize indents to 4 spaces for all source files
Mon Jul 21 08:27:50 2014 -0700 Will Anderson <will@itsananderson.com>

bb82a7a Bump version to 0.2.4
Mon Jul 21 01:11:38 2014 -0700 Will Anderson <will@itsananderson.com>

d9bb2ff Update .npmignore with new dev files
Mon Jul 21 01:11:18 2014 -0700 Will Anderson <will@itsananderson.com>

c75067e Add full coverage for application mixin
Sun Jul 20 23:00:44 2014 -0700 Will Anderson <will@itsananderson.com>

98ee9cb Add full coverage for routeParams utility
Sun Jul 20 14:34:18 2014 -0700 Will Anderson <will@itsananderson.com>

62aa4c9 Add full coverage for normalizeType utility
Sun Jul 20 14:27:36 2014 -0700 Will Anderson <will@itsananderson.com>

4d66823 Remove if check for 'next' in initialDeps utility
Sun Jul 20 14:05:43 2014 -0700 Will Anderson <will@itsananderson.com>

11a8b02 Add full coverage for contentDisposition utility
Sun Jul 20 14:02:29 2014 -0700 Will Anderson <will@itsananderson.com>

be8cd24 Add full coverage for acceptParams utility
Sun Jul 20 13:43:01 2014 -0700 Will Anderson <will@itsananderson.com>

f551196 Add full coverage for send provider
Sun Jul 20 11:40:12 2014 -0700 Will Anderson <will@itsananderson.com>
```

<https://willi.am/blog/2015/02/19/customize-your-git-log-format/>

# Conceptos

- Repositorio Distribuido
- Workflow
  - Repo Local (Working Dir)
  - INDEX
  - HEAD
- Branch(es)
  - Merge (OMG)
  - Rebase (OMG<sup>2</sup>)
  - Diff
- TAGGING
- LOGs (GIT no miente)
- **Blame** (Quien fue?)
- RollBack
  - Revert
  - Checkout
  - Fetch origin
  - Reset



# Conceptos

- Repitorio Distribuido
- Workflow
  - Repo Local (Working Dir)
  - INDEX
  - HEAD
- Branch(es)
  - Merge (OMG)
  - Rebase (OMG<sup>2</sup>)
  - Diff
- TAGGING
- LOGs (GIT no miente)
- Blame (Quien fue?)
- RollBack
  - Revert
  - Checkout
  - Fetch origin
  - Reset

```
biswa@DESKTOP-LABEVR ~ /Desktop/web_dev/sideGithub/git_blame_demo/create-react-app (main)
$ git blame CONTRIBUTING.md
46cf3fc43 (Dan Abramov) 2016-09-02 14:29:08 +0100 1) # Contributing to Create React
App
d7ded9e33 (Max Stoiber) 2016-07-24 18:34:31 +0200 2)
46cf3fc43 (Dan Abramov) 2016-09-02 14:29:08 +0100 3) Loving Create React App and wan
t to get involved? Thanks! There are plenty of ways you can help.
d7ded9e33 (Max Stoiber) 2016-07-24 18:34:31 +0200 4)
1d03579f5 (Ian Sutherland) 2019-10-24 11:14:15 -0700 5) Please take a moment to review
this document in order to make the contribution process straightforward and effective for everyone i
nvolved.
d7ded9e33 (Max Stoiber) 2016-07-24 18:34:31 +0200 6)
d7ded9e33 (Max Stoiber) 2016-07-24 18:34:31 +0200 7) Following these guidelines help
s to communicate that you respect the time of the developers managing and developing this open sourc
e project. In return, they should reciprocate that respect in addressing your issue or assessing pat
ches and features.
d7ded9e33 (Max Stoiber) 2016-07-24 18:34:31 +0200 8)
46cf3fc43 (Dan Abramov) 2016-09-02 14:29:08 +0100 9) ## Core Ideas
d7ded9e33 (Max Stoiber) 2016-07-24 18:34:31 +0200 10)
46cf3fc43 (Dan Abramov) 2016-09-02 14:29:08 +0100 11) As much as possible, we try to
avoid adding configuration and flags. The purpose of this tool is to provide the best experience for
people getting started with React, and this will always be our first priority. This means that some
times we [sacrifice additional functionality](https://gettingreal.37signals.com/ch05_Half_Not_Half_A
ssed.php) (such as server rendering) because it is too hard to solve it in a way that wouldn't requi
re any configuration.
d7ded9e33 (Max Stoiber) 2016-07-24 18:34:31 +0200 12)
f2a8535b5d (Alexander James Phillips) 2017-02-22 18:37:18 +0000 13) We prefer **convention, heurist
```

# Conceptos

- Repositorio Distribuido
- Workflow
  - Repo Local (Working Dir)
  - INDEX
  - HEAD
- Branch(es)
  - Merge (OMG)
  - Rebase (OMG<sup>2</sup>)
  - Diff
- TAGGING
- LOGs (GIT no miente)
- Blame (Quien fue?)
- RollBack
  - Revert
  - Checkout
  - Fetch origin
  - Reset



# Conceptos

- Repitorio Distribuido
- Workflow
  - Repo Local (Working Dir)
  - INDEX
  - HEAD
- Branch(es)
  - Merge (OMG)
  - Rebase (OMG<sup>2</sup>)
  - Diff
- TAGGING
- LOGs (GIT no miente)
- Blame (Quien fue?)
- RollBack
  - **Revert**
  - **Checkout**
  - **Fetch origin**
  - **Reset**

| Command                   | Scope        | Common use cases  |
|---------------------------|--------------|---|
| <code>git reset</code>    | Commit-level | Discard commits in a private branch or throw away uncommitted changes |
| <code>git reset</code>    | File-level   | Unstage a file  |
| <code>git checkout</code> | Commit-level | Switch between branches or inspect old snapshots                      |
| <code>git checkout</code> | File-level   | Discard changes in the working directory                              |
| <code>git revert</code>   | Commit-level | Undo commits in a public branch                                       |
| <code>git revert</code>   | File-level   | (N/A)   |

# Conceptos

- Repositorio Distribuido
- Workflow
  - Repo Local (Working Dir)
  - INDEX
  - HEAD
- Branch(es)
  - Merge (OMG)
  - Rebase (OMG<sup>2</sup>)
  - Diff
- TAGGING
- LOGs (GIT no miente)
- Blame (Quien fue?)
- RollBack
  - Revert
  - **Checkout**
  - Fetch origin
  - Reset

```
~/Documents/WS/linux-stable > git status
On branch master
Your branch is up to date with 'origin/master'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   README

no changes added to commit (use "git add" and/or "git commit -a")
~/Documents/WS/linux-stable > git checkout HEAD README
Updated 1 path from 116e94c55fc7
~/Documents/WS/linux-stable > git status
On branch master
Your branch is up to date with 'origin/master'.

nothing to commit, working tree clean
~/Documents/WS/linux-stable > _
```

# Comandos

- \$ git help git
- \$ git init
- \$ git clone
- \$ git status
- \$ git pull
- \$ git checkout
- \$ git diff
- \$ git add
- \$ git commit
- \$ git push

```
~ : man — Konsole
File Edit View Bookmarks Plugins Settings Help
GIT(1)                                     Git Manual                                     GIT(1)

NAME
    git - the stupid content tracker

SYNOPSIS
git [-v | --version] [-h | --help] [-C <path>] [-c <name>=<value>]
        [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
        [-p|--paginate|-P|--no-pager] [--no-replace-objects] [--bare]
        [--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]
        [--super-prefix=<path>] [--config-env=<name>=<envvar>]
        <command> [<args>]

DESCRIPTION
    Git is a fast, scalable, distributed revision control system with an unusually rich command set that provides both high-level operations and full access to internals.

    See gittutorial\(7\) to get started, then see giteveryday\(7\) for a useful minimum set of commands. The Git User's Manual\[1\] has a more in-depth introduction.

    After you mastered the basic concepts, you can come back to this page to learn what commands Git offers. You can learn more about individual Git commands with "git help command". gitcli\(7\) manual page gives you an overview of the command-line command syntax.

    A formatted and hyperlinked copy of the latest Git documentation can be viewed at https://git.github.io/htmldocs/git.html or https://git-scm.com/docs.

OPTIONS
    -v, --version
        Prints the Git suite version that the git program came from.

Manual page git(1) line 1 (press h for help or q to quit)
```

# Comandos

- \$ git help git
- **\$ git init**
- \$ git clone
- \$ git status
- \$ git pull
- \$ git checkout
- \$ git diff
- \$ git add
- \$ git commit
- \$ git push

```
~/Documents/WS/git101 > git init
hint: Using 'master' as the name for the initial branch. This default branch name
hint: is subject to change. To configure the initial branch name to use in all
hint: of your new repositories, which will suppress this warning, call:
hint:
hint:   git config --global init.defaultBranch <name>
hint:
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this command:
hint:
hint:   git branch -m <name>
Initialized empty Git repository in /home/masardis/Documents/WS/git101/.git/
~/Documents/WS/git101 > ls .git
HEAD  config  description  hooks  info  objects  refs
~/Documents/WS/git101 > _
```

# Comandos

- \$ git help git
- \$ git init
- **\$ git clone**
- \$ git status
- \$ git pull
- \$ git checkout
- \$ git diff
- \$ git add
- \$ git commit
- \$ git push

```
~/Documents/WS > git clone git://git.kernel.org/pub/scm/linux/kernel/git/stable/linux-stable.git
Cloning into 'linux-stable'...
remote: Enumerating objects: 866, done.
remote: Counting objects: 100% (866/866), done.
remote: Compressing objects: 100% (397/397), done.
remote: Total 10769890 (delta 655), reused 540 (delta 469), pack-reused 10769824
Receiving objects: 100% (10769890/10769890), 4.17 GiB | 527.00 KiB/s, done.
Resolving deltas: 100% (8600002/8600002), done.
Updating files: 100% (78088/78088), done.
```

# Comandos

- \$ git help git
- \$ git init
- \$ git clone
- **\$ git status**
- \$ git pull
- \$ git checkout
- \$ git diff
- \$ git add
- \$ git commit
- \$ git push

```
~/Documents/WS/linux-stable > git status
On branch master
Your branch is up to date with 'origin/master'.

nothing to commit, working tree clean
~/Documents/WS/linux-stable > vim README
~/Documents/WS/linux-stable > git status
On branch master
Your branch is up to date with 'origin/master'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
    (use "git restore <file>..." to discard changes in working directory)
            modified:   README

no changes added to commit (use "git add" and/or "git commit -a")
~/Documents/WS/linux-stable > _
```

# Comandos

- \$ git help git
- \$ git init
- \$ git clone
- \$ git status
- **\$ git pull**
- \$ git checkout
- \$ git diff
- \$ git add
- \$ git commit
- \$ git push

```
~/Documents/WS/linux-stable > git pull
remote: Enumerating objects: 21855, done.
remote: Counting objects: 100% (21855/21855), done.
remote: Compressing objects: 100% (118/118), done.
remote: Total 53024 (delta 21769), reused 21759 (delta 21736), pack-reused 31169
Receiving objects: 100% (53024/53024), 21.97 MiB | 229.00 KiB/s, done.
Resolving deltas: 100% (43926/43926), completed with 7323 local objects.
From git://git.kernel.org/pub/scm/linux/kernel/git/stable/linux-stable
  2bca25eaeba6..493ffd6605b2  master      -> origin/master
    f28b7414ab71..6e150d605c9e  linux-5.4.y -> origin/linux-5.4.y
 * [new tag]                  v5.4.217   -> v5.4.217
Updating 2bca25eaeba6..493ffd6605b2
-
```

# Comandos

- \$ git help git
- \$ git init
- \$ git clone
- \$ git status
- **\$ git pull**
- \$ git checkout
- \$ git diff
- \$ git add
- \$ git commit
- \$ git push

```
tools/testing/selftests/vm/Makefile
tools/testing/selftests/vm/gup_test.c
tools/testing/selftests/vm/hmm-tests.c
tools/testing/selftests/vm/ksm_tests.c
virt/kvm/kvm_main.c
5754 files changed, 289284 insertions(+), 96299 deletions(-)
create mode 100644 Documentation/ABI/testing/sysfs-amd-pmc
create mode 100644 Documentation/ABI/testing/sysfs-amd-pmf
create mode 100644 Documentation/ABI/testing/sysfs-bus-iio-bno055
create mode 100644 Documentation/ABI/testing/sysfs-bus-iio-cdc-ad7746
create mode 100644 Documentation/ABI/testing/sysfs-devices-hisi_ptt
```

# Comandos

- \$ git help git
- \$ git init
- \$ git clone
- \$ git status
- \$ git pull
- **\$ git checkout**
- \$ git diff
- \$ git add
- \$ git commit
- \$ git push

```
~/Documents/WS/linux-stable > git --no-pager branch -r | grep linux-6
  origin/linux-6.0.y
~/Documents/WS/linux-stable > git checkout origin/linux-6.0.y
Updating files: 100% (9543/9543), done.
Note: switching to 'origin/linux-6.0.y'.

You are in 'detached HEAD' state. You can look around, make experimental
changes and commit them, and you can discard any commits you make in this
state without impacting any branches by switching back to a branch.

If you want to create a new branch to retain commits you create, you may
do so (now or later) by using -c with the switch command. Example:

  git switch -c <new-branch-name>

Or undo this operation with:

  git switch -

Turn off this advice by setting config variable advice.detachedHead to false

HEAD is now at 4fe89d07dcc2 Linux 6.0
~/Documents/WS/linux-stable > _
```

# Comandos

- \$ git help git
- \$ git init
- \$ git clone
- \$ git status
- \$ git pull
- \$ git checkout
- **\$ git diff**
- \$ git add
- \$ git commit
- \$ git push

```
~/Documents/WS/linux-stable > git --no-pager diff remotes/origin/linux-5.9.y..remotes/origin/linux-6.0.y | head
diff --git a/.clang-format b/.clang-format
index badfc1ba440a..1247d54f9e49 100644
--- a/.clang-format
+++ b/.clang-format
@@ -1,6 +1,6 @@
 # SPDX-License-Identifier: GPL-2.0
 #
-# clang-format configuration file. Intended for clang-format >= 4.
+# clang-format configuration file. Intended for clang-format >= 11.
#
~/Documents/WS/linux-stable > _
```

# Comandos

- \$ git help git
- \$ git init
- \$ git clone
- \$ git status
- \$ git pull
- \$ git checkout
- \$ git diff
- \$ **git add**
- \$ **git commit**
- \$ git push

```
~/Documents/WS/linux-stable > git --no-pager branch
* (HEAD detached at origin/linux-6.0.y)
  master
~/Documents/WS/linux-stable > git checkout linux-6.0.y
branch 'linux-6.0.y' set up to track 'origin/linux-6.0.y'.
Switched to a new branch 'linux-6.0.y'
~/Documents/WS/linux-stable > git --no-pager branch
* linux-6.0.y
  master
~/Documents/WS/linux-stable > vim README
~/Documents/WS/linux-stable > git add README
~/Documents/WS/linux-stable > git commit -m 'Hacking Kernel :)'
[linux-6.0.y ec1c0c44508b] Hacking Kernel :)
  1 file changed, 2 insertions(+)
~/Documents/WS/linux-stable > _
```

# Comandos

- \$ git help git
- \$ git init
- \$ git clone
- \$ git status
- \$ git pull
- \$ git checkout
- \$ git diff
- **\$ git add**
- **\$ git commit**
- \$ git push

```
~/Documents/WS/linux-stable > git --no-pager diff origin/linux-6.0.y..linux-6.0.y README
diff --git a/README b/README
index 669ac7c32292..f1c245578434 100644
--- a/README
+++ b/README
@@ -16,3 +16,5 @@ several of them using the Restructured Text markup notation.
Please read the Documentation/process/changes.rst file, as it contains the
requirements for building and running the kernel, and information about
the problems which may result by upgrading your kernel.
+
+My awesome contribution!
~/Documents/WS/linux-stable > _
```

# Comandos

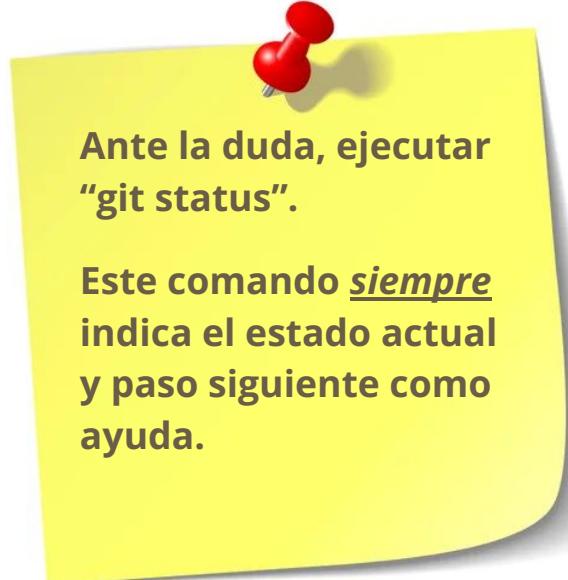
- \$ git help git
- \$ git init
- \$ git clone
- \$ git status
- \$ git pull
- \$ git checkout
- \$ git diff
- \$ git add
- \$ git commit
- **\$ git push**

```
~/Documents/WS/github101 > vim README.md
~/Documents/WS/github101 > git commit -a -m 'Nuevo Commit'
[main b2043af] Nuevo Commit
  1 file changed, 1 insertion(+)
~/Documents/WS/github101 > git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 279 bytes | 279.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To github.com:toscosys/github101.git
  88fdcf7..b2043af  main -> main
~/Documents/WS/github101 > _
```

# Comandos

Aún hay más!

- \$ git branch
- \$ git merge
- \$ git rm
- \$ git mv
- \$ git rebase

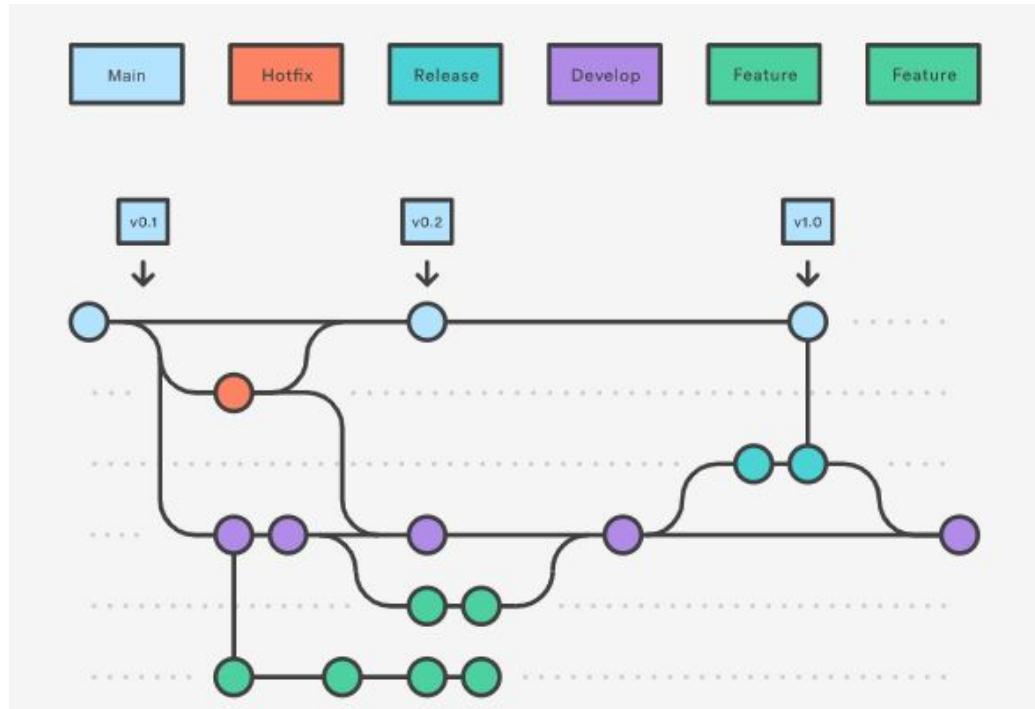


# Buenas prácticas

- Git Flow Workflow
  - Dos ramas principales:
    - develop
    - main
  - Manejo de ramas para:
    - feature
    - hotfix



SIEMPRE sigue los  
lineamientos de tu  
equipo de trabajo. Si  
algo no cuadra,  
comentalo.



# Comparamos GitFlow con modelo tradicional

```
$ git checkout main  
  
$ git checkout -b develop  
  
$ git checkout -b feature_branch  
  
# work happens on feature branch  
  
$ git checkout develop  
  
$ git merge feature_branch  
  
$ git checkout main  
  
$ git merge develop  
  
$ git branch -d feature_branch  
  
$ git checkout main  
  
$ git checkout -b hotfix_branch  
  
# work is done commits are added to the hotfix_branch  
  
$ git checkout develop  
  
$ git merge hotfix_branch  
  
$ git checkout main  
  
$ git merge hotfix_branch
```

```
$ git flow init  
  
$ git flow feature start feature_branch  
  
$ git flow feature finish feature_branch  
  
$ git flow release start 0.1.0  
  
$ git flow release finish '0.1.0'  
  
$ git flow hotfix start hotfix_branch  
  
$ git flow hotfix finish hotfix_branch
```

# Demo de operaciones básicas

- Configuramos nuestra instancia
- Creamos nuestras firmas
- Instalamos GIT  
(<https://git-scm.com/downloads>)
- Creamos un repo (git init)
- Creamos código (crear archivos)
- Añadimos los cambios (git add)
- Commit de código (git commit)
- Excluimos archivos sensibles  
(passwords y otros, cuidado!)



```
robertito@gentoo ~ $ git config --global user.name "Robertito"
robertito@gentoo ~ $ git config --global user.email robertito@gmail.com
robertito@gentoo ~ $ git --no-pager config --list
user.name=Robertito
user.email=robertito@gmail.com
robertito@gentoo ~ $ ssh-keygen -t ed25519 -C "robertito@gmail.com"
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/robertito/.ssh/id_ed25519):
/home/robertito/.ssh/id_ed25519 already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/robertito/.ssh/id_ed25519
Your public key has been saved in /home/robertito/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:fy0VvY060hBrBfULDQfn/FesWpcmuk1FENnKdUOTb4 robertito@gmail.com
The key's randomart image is:
+--[ED25519 256]--+
|          .*=|
|          .o++|
|          o  .=|
|          . * o . |=
|          o S B + E |
|          . . B o +..o|
|          o = * *+.|
|          = = + 0o..|
|          . . o oo. |
+---[SHA256]-----+
robertito@gentoo ~ $ cat ~/.ssh/id_ed25519
id_ed25519      id_ed25519.pub
robertito@gentoo ~ $ cat ~/.ssh/id_ed25519.pub
ssh-ed25519 AAAAC3NzaC1ZDI1NTESAAAIIlIqY0it+fNkNbKEtQTeWAqp01T9eBlzks67ih76w3H robertito@gmail.com
robertito@gentoo ~ $
```

# Demo de operaciones básicas

- Configuramos nuestra instancia
- Creamos nuestras firmas
- **Instalamos GIT**  
[\*\*\(https://git-scm.com/downloads\)\*\*](https://git-scm.com/downloads)
- **Creamos un repo (git init)**
- **Creamos código (crear archivos)**
- **Añadimos los cambios (git add)**
- Commit de código (git commit)
- Excluimos archivos sensibles  
(passwords y otros, cuidado!)



```
robertito@gentoo ~ $ mkdir -p ~/Repos/git101
robertito@gentoo ~ $ cd ~/Repos/git101
robertito@gentoo ~/Repos/git101 $ git init
hint: Using 'master' as the name for the initial branch. This default branch name
hint: is subject to change. To configure the initial branch name to use in all
hint: of your new repositories, which will suppress this warning, call:
hint:
hint:   git config --global init.defaultBranch <name>
hint:
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this command:
hint:
hint:   git branch -m <name>
Initialized empty Git repository in /home/robertito/Repos/git101/.git/
robertito@gentoo ~/Repos/git101 $ echo "# Mi primer Readme en Git" > README.md
robertito@gentoo ~/Repos/git101 $
```

# Demo de operaciones básicas

- Configuramos nuestra instancia
- Creamos nuestras firmas
- Instalamos GIT  
(<https://git-scm.com/downloads>)
- Creamos un repo (git init)
- Creamos código (crear archivos)
- Añadimos los cambios (git add)
- **Commit de código (git commit)**
- Excluimos archivos sensibles  
(passwords y otros, cuidado!)



```
robertito@gentoo ~/Repos/git101 $ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    README.md

nothing added to commit but untracked files present (use "git add" to track)
robertito@gentoo ~/Repos/git101 $ git add README.md
robertito@gentoo ~/Repos/git101 $ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   README.md

robertito@gentoo ~/Repos/git101 $ git commit -m "Initial Commit"
[master (root-commit) 26699d2] Initial Commit
  1 file changed, 1 insertion(+)
  create mode 100644 README.md
robertito@gentoo ~/Repos/git101 $ git status
On branch master
nothing to commit, working tree clean
robertito@gentoo ~/Repos/git101 $ _
```

# Demo de operaciones básicas

- Configuramos nuestra instancia
- Creamos nuestras firmas
- Instalamos GIT  
(<https://git-scm.com/downloads>)
- Creamos un repo (git init)
- Creamos código (crear archivos)
- Añadimos los cambios (git add)
- Commit de código (git commit)
- **Excluimos archivos sensibles  
(passwords y otros, cuidado!)**



```
robertito@gentoo ~/Repos/git101 $ git status
On branch main
Your branch is up to date with 'origin/main'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    secret

nothing added to commit but untracked files present (use "git add" to track)
robertito@gentoo ~/Repos/git101 $ echo "secret" > .gitignore
robertito@gentoo ~/Repos/git101 $ git status
On branch main
Your branch is up to date with 'origin/main'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    .gitignore

nothing added to commit but untracked files present (use "git add" to track)
robertito@gentoo ~/Repos/git101 $ echo ".gitignore" >> .gitignore
robertito@gentoo ~/Repos/git101 $ git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
robertito@gentoo ~/Repos/git101 $ _
```

# User interfaces

(web, y CLI)

## Web

- Github (github.com)
- Gitlab (gitlab.com)
- BitBucket

## Grafica

- gitk

## CLI (ncurses)

- tig(browser only, muy bueno)
- Lazygit (full, excelente!)
- GitUI 0.19 (full, donacion)
- GitHub Desktop

## Tutorial Interactivo:

- <https://learngitbranching.js.org/>

# Github (Free/Pago)

This screenshot shows a GitHub project page for 'octocat-classifier'. The top navigation bar includes links for 'Pull requests', 'Issues', 'Codespaces', 'Marketplace', and 'Explore'. The repository name 'jasonetco / octocat-classifier' is displayed. Below the header, there are sections for 'Code' (with tabs for 'main', 'branches', and 'tags'), 'About' (describing the project as an Octocat classifier), 'Contributors' (listing three contributors with profile pictures), 'Languages' (JavaScript 78%, HTML 21%, Other 1%), and 'Community profile' (showing 66% of recommended community standards). The main content area displays a list of recent commits from 'dependabot' and other contributors, along with files like README.md, LICENSE, and index.js.

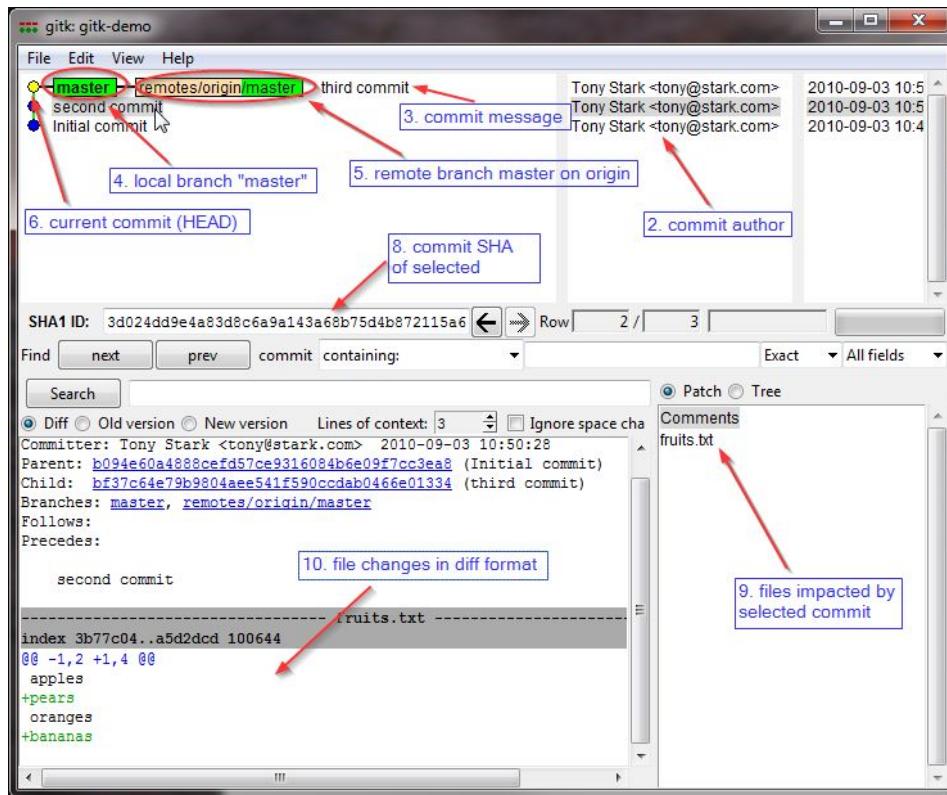
# Gitlab (Free/Pago/Software Open Source)

This screenshot shows a GitLab project page for 'gitlab-ce'. The top navigation bar includes links for 'Merge requests', 'Project', 'Issues', 'Requirements', 'CI/CD', 'Deployments', 'Monitor', 'Packages & Registries', and 'Analytics'. The repository name 'gitlab / gitlab-ce' is displayed. Below the header, there are sections for 'About' (describing GitLab as an open source end-to-end software development platform), 'Merge requests' (listing 1,415 open merge requests), 'Project' (showing 244K commits, 37K branches, 1.97TB of storage, 8.5GB files, 64.4TB storage, and 113 releases), and a detailed commit history table. The commit history table lists various commits from different authors, including 'gitlab' and 'SeeMcGraw', with details like commit message, author, date, and last update.

# BitBucket (Free/Pago)

This screenshot shows a BitBucket pipeline for the repository 'alpha-team-app'. The top navigation bar includes links for 'Your work', 'Repositories', 'Projects', 'More', and 'Create'. The pipeline name is 'Pipeline #14'. The pipeline consists of four stages: 'build', 'test', 'Deploy to staging', and 'Deploy to production'. The 'build' stage is currently 'In progress'. The pipeline log on the right side shows the command 'npm install' and its execution details, including registry requests and file downloads. The log output is extensive, showing multiple lines of npm command logs.

# Ejemplo de un cliente GIT: gitk (gtk)



# Caso curioso: tig

“GIT Browser basado en ncurses”

```
2019-04-28 13:14 +0100 Olaf Alders
2019-04-28 12:48 +0200 Mickey
2019-04-28 10:57 +0100 Olaf Alders
2019-04-27 17:39 +0100 Olaf Alders
2019-04-27 17:37 +0100 Leo Lapworth
2019-04-27 16:35 +0100 Olaf Alders
2019-04-27 17:21 +0200 Mickey Nasriachi
2019-04-27 15:38 +0100 Shawn Sorichetti
2019-04-27 15:34 +0100 Leo Lapworth
2019-04-27 15:21 +0100 Shawn Sorichetti
2019-04-27 13:23 +0100 Olaf Alders
2019-04-27 14:51 +0100 Leo Lapworth
2019-04-27 14:40 +0100 Leo Lapworth
2019-04-27 14:33 +0100 Olaf Alders
2019-04-27 15:07 +0200 Mickey Nasriachi
2019-04-27 15:50 +0300 Mickey
2019-04-27 13:35 +0100 Olaf Alders
2019-04-27 13:19 +0100 Olaf Alders
2019-04-27 13:32 +0100 Olaf Alders
2019-04-27 12:35 +0100 Olaf Alders
2019-04-27 12:25 +0100 Olaf Alders
2019-04-27 13:02 +0200 Mickey Nasriachi
2019-04-27 12:59 +0200 Mickey Nasriachi
2019-04-26 13:10 +0200 Mickey Nasriachi
2019-04-27 14:20 +0300 Mickey
2019-04-18 09:09 +0900 Kang-min Liu
2019-04-27 12:00 +0100 Shawn Sorichetti
2019-04-27 11:20 +0100 Olaf Alders
2019-04-27 11:18 +0100 Shawn Sorichetti
2019-04-27 10:53 +0100 Olaf Alders
2019-04-27 10:46 +0100 Olaf Alders
[main] 7c5a3e57baa6497c1c1bc4fa4b8e47b77e8d27a4 - commit 4 of 2741 15
Line 4 matches 'docker' (1 of 10)

o [oalders/bin-prove] {origin/oalders/bin-prove} /bin/prove no longer
M- [master] {origin/master} Merge pull request #911 from metacpan/o
o [oalders/latest-timing] Tweak index latest jobs to coincide (all
M- Merge pull request #909 from metacpan/leo/docker conditional bu
o make conditional deploy actually work
M- Merge pull request #908 from metacpan/mickey/gh736
o Fix size parameter reading for reverse_dependencies (GH#736)
M- Merge pull request #907 from metacpan/leo/create_docker_image
o only deploy on the snapshot version
M- Merge pull request #902 from metacpan/oalders/getopt-long-desc
o [oalders/getopt-long-descriptive] Upgrade Getopt::Long::Descript
M- Merge pull request #906 from metacpan/leo/create_docker_image
o add deploy to docker image, just on master merge
M- Merge pull request #905 from metacpan/mickey/fix_dist_creation
o Check if distribution exists before creating
M- Merge pull request #904 from metacpan/oalders/rogue
o Tidy use statements
o Don't index Acme::DependOnEverything
o Tidy constants
o Tidy
M- Merge pull request #903 from metacpan/mickey/remove_unused_scrip
o Removed unused Script::PerlMongers
o Removed unused Script::Pagerank
o Removed unused Script::ReindexDist
M- Merge pull request #894 from gugod/release-query-size
o Set th "size" of resultset to be the same as input size.
M- Merge pull request #901 from metacpan/oalders/docker-user
o Run as root in Docker container
M- Merge pull request #900 from metacpan/oalders/server-status
o Remove ServerStatus::Lite plugin for now
o Sort use statements in MetaCPAN::Server
```

# Manos a la obra! (GIT interactivo)

<https://learngitbranching.js.org/>



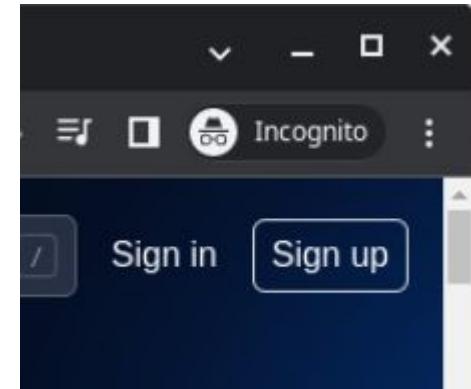
The screenshot shows a desktop application window titled "Learn Git Branching". On the left, a terminal window displays the command history:

```
$ help
$ levels
```

On the right, there is a large blue area containing a diagram of a git repository's history. The diagram consists of two nodes, each represented by a pink circle with a white letter inside. A vertical arrow points from the bottom node to the top one. The bottom node is labeled "c1" and has a pink arrow pointing to it from the right, labeled "main\*". The top node is labeled "c8". At the bottom right of the blue area, there is a small toolbar with icons for help, search, and other functions.

# Demo de GitHub

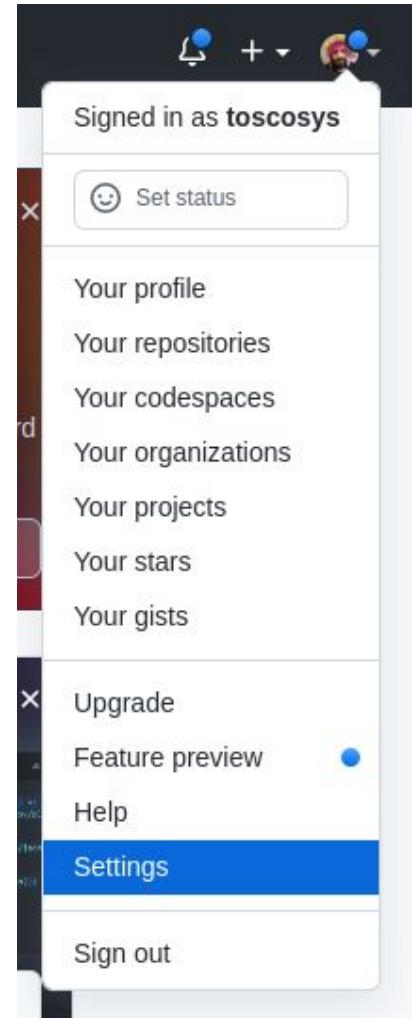
- **Creamos nuestra cuenta en GitHub**
- Configuramos nuestras Keys
- Creamos nuestro repositorio
- Añadimos Origin en nuestro repo
- Subimos y bajamos cambios
- Empezamos a branchear
- Empezamos a taggear



A screenshot of a web browser window showing the GitHub homepage. The window title bar says 'GitHub: Where the world buil...'. The address bar shows 'github.com'. The page has a dark blue header with the GitHub logo and navigation links. A search bar at the top right contains the placeholder 'Search GitHub'. Below the header, there are links for 'Product', 'Solutions', 'Open Source', and 'Pricing'. On the right side of the page, there are 'Sign in' and 'Sign up' buttons.

# Demo de GitHub

- Creamos nuestra cuenta en GitHub
- **Configuramos nuestras Keys**
- Creamos nuestro repositorio
- Añadimos Origin en nuestro repo
- Subimos y bajamos cambios
- Empezamos a branchear
- Empezamos a taggear



# Demo de GitHub

- Creamos nuestra cuenta en GitHub
- **Configuramos nuestras Keys**
- Creamos nuestro repositorio
- Añadimos Origin en nuestro repo
- Subimos y bajamos cambios
- Empezamos a branchear
- Empezamos a taggear

## Access

- Billing and plans
- Emails
- Password and authentication
- SSH and GPG keys
- Organizations
- Moderation



## SSH keys

New SSH key

This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.

# Demo de GitHub

- Creamos nuestra cuenta en GitHub
- **Configuramos nuestras Keys**
- Creamos nuestro repositorio
- Añadimos Origin en nuestro repo
- Subimos y bajamos cambios
- Empezamos a branchear
- Empezamos a taggear

SSH keys / Add new

---

Title

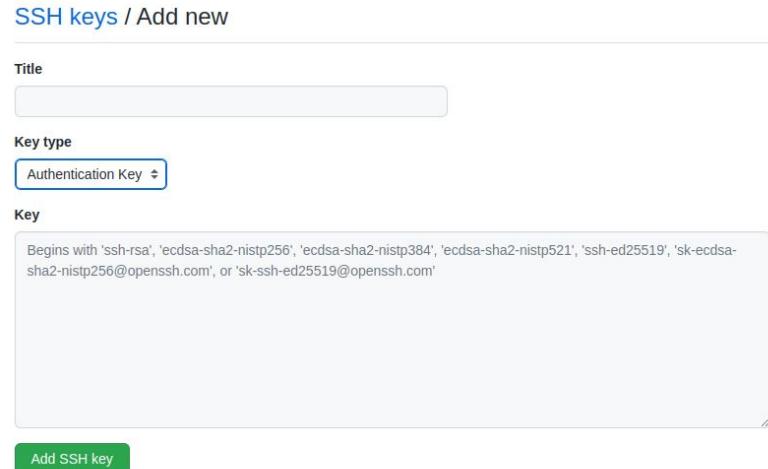
Key type

Authentication Key

Key

Begins with 'ssh-rsa', 'ecdsa-sha2-nistp256', 'ecdsa-sha2-nistp384', 'ecdsa-sha2-nistp521', 'ssh-ed25519', 'sk-ecdsa-sha2-nistp256@openssh.com', or 'sk-ssh-ed25519@openssh.com'

Add SSH key

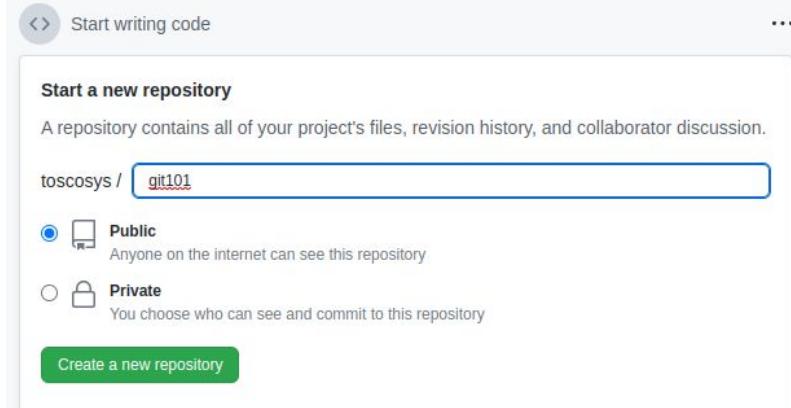


# Demo de GitHub

- Creamos nuestra cuenta en GitHub
- Configuramos nuestras Keys
- **Creamos nuestro repositorio**
- Añadimos Origin en nuestro repo
- Subimos y bajamos cambios
- Empezamos a branchear
- Empezamos a taggear

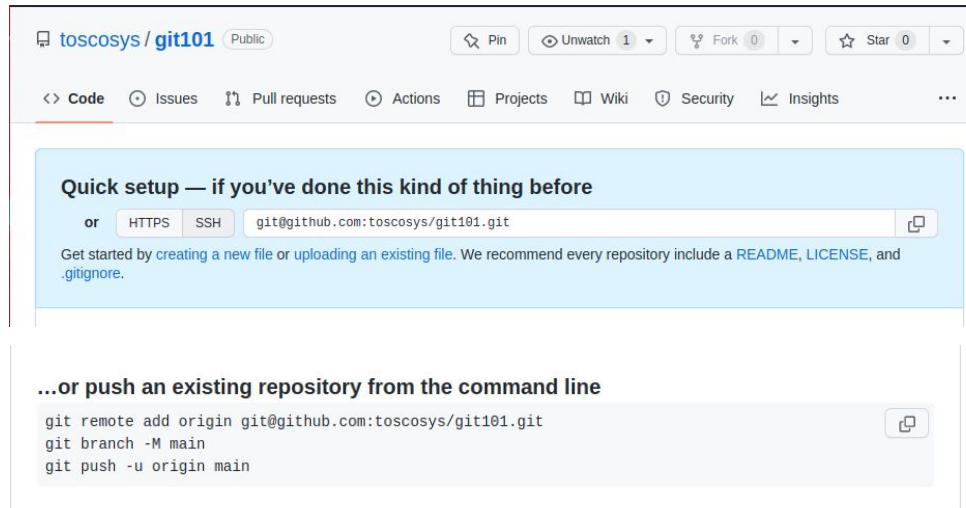
The home for all developers — including you.

Welcome to your personal dashboard, where you can find an introduction to how GitHub works, tools to help you build software, and help merging your first lines of code.



# Demo de GitHub

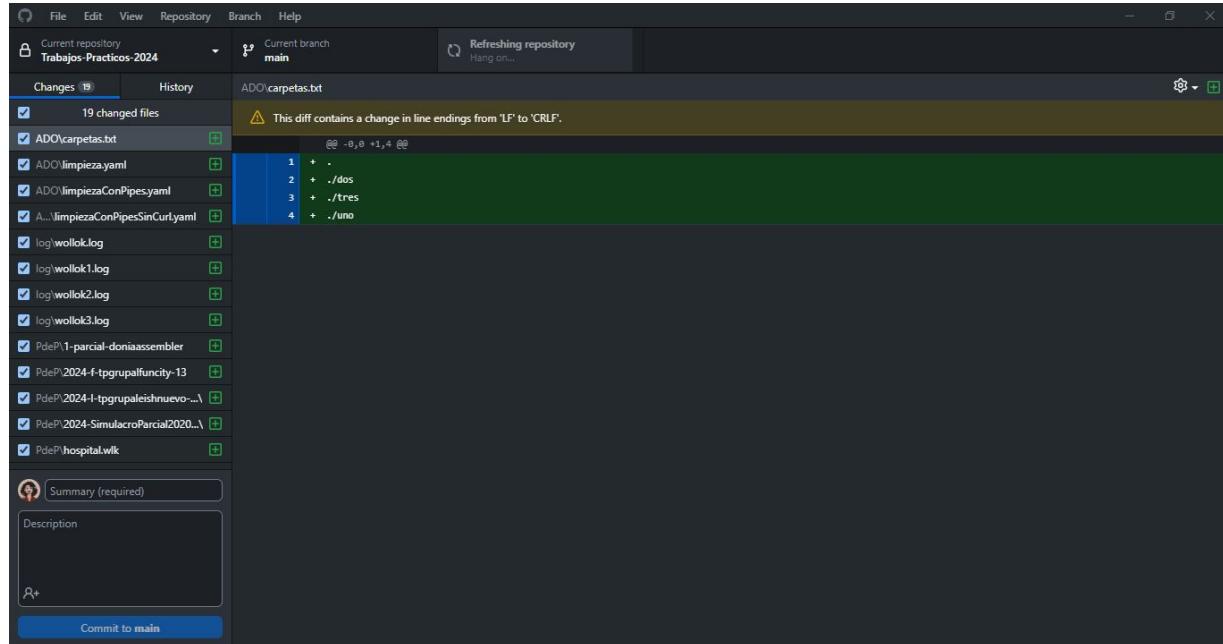
- Creamos nuestra cuenta en GitHub
- Configuramos nuestras Keys
- **Creamos nuestro repositorio**
- Añadimos Origin en nuestro repo
- Subimos y bajamos cambios
- Empezamos a branchear
- Empezamos a taggear



# GitHub Desktop

Si aún le tenés miedo a los comandos, podés empezar a familiarizarte con GitHub Desktop.

- Instalar [GitHub Desktop](#)
- Seguir instrucciones del instalador
- Configurar cuenta de GitHub
- A romper



# Demo de GitHub

- Creamos nuestra cuenta en GitHub
- Configuramos nuestras Keys
- Creamos nuestro repositorio
- **Añadimos Origin en nuestro repo**
- Subimos y bajamos cambios
- Empezamos a branchear
- Empezamos a taggear

```
robertito@gentoo ~/Repos/git101 $ git remote add origin git@github.com:toscosys/git101.git
robertito@gentoo ~/Repos/git101 $ git branch -M main
robertito@gentoo ~/Repos/git101 $ git push -u origin main
The authenticity of host 'github.com (20.201.28.151)' can't be established.
ED25519 key fingerprint is SHA256:+DiY3wvvV6TuJJhbpZisF/zLDA0zPMSvHdkr4UvC0qU.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'github.com' (ED25519) to the list of known hosts.
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 237 bytes | 237.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:toscosys/git101.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
robertito@gentoo ~/Repos/git101 $ _
```

# Demo de GitHub

- Creamos nuestra cuenta en GitHub
- Configuramos nuestras Keys
- Creamos nuestro repositorio
- Añadimos Origin en nuestro repo
- **Subimos y bajamos cambios**
- Empezamos a branchear
- Empezamos a taggear

```
robertito@gentoo ~/Repos/git101 $ git pull
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 777 bytes | 777.00 KiB/s, done.
From github.com:toscocosys/git101
  26699d2..123698c main      -> origin/main
Updating 26699d2..123698c
Fast-forward
 index.html | 9 ++++++++
 1 file changed, 9 insertions(+)
 create mode 100644 index.html
robertito@gentoo ~/Repos/git101 $ _
```

```
robertito@gentoo ~/Repos/git101 $ git --no-pager log
commit 123698cdcd7d3b52254bd208cb8a4c626670e6c (HEAD -> main, origin/main)
Author: Tosco Sys <sardiscomatias@gmail.com>
Date:   Wed Oct 12 00:18:21 2022 -0300

        Create index.html

commit 26699d24dca47cf2a284b0605457df5b3f3099bc
Author: Robertito <robertito@gmail.com>
Date:   Tue Oct 11 20:41:49 2022 -0300

        Initial Commit
robertito@gentoo ~/Repos/git101 $ _
```

# Demo de GitHub

- Creamos nuestra cuenta en GitHub
- Configuramos nuestras Keys
- Creamos nuestro repositorio
- Añadimos Origin en nuestro repo
- Subimos y bajamos cambios
- **Empezamos a branchear**
- Empezamos a taggear

```
robertito@gentoo ~/Repos/git101 $ git branch develop
robertito@gentoo ~/Repos/git101 $ git branch
robertito@gentoo ~/Repos/git101 $ git --no-pager branch
  develop
* main
robertito@gentoo ~/Repos/git101 $ git checkout develop
Switched to branch 'develop'
robertito@gentoo ~/Repos/git101 $ git --no-pager branch
* develop
  main
robertito@gentoo ~/Repos/git101 $ _
```

# Demo de GitHub

- Creamos nuestra cuenta en GitHub
- Configuramos nuestras Keys
- Creamos nuestro repositorio
- Añadimos Origin en nuestro repo
- Subimos y bajamos cambios
- **Empezamos a branchear**
- Empezamos a taggear

```
robertito@gentoo ~/Repos/git101 $ vim index.html
robertito@gentoo ~/Repos/git101 $ git commit -a -m "Creando dev env"
[develop db92ec9] Creando dev env
 1 file changed, 1 insertion(+), 1 deletion(-)
robertito@gentoo ~/Repos/git101 $ git push
fatal: The current branch develop has no upstream branch.
To push the current branch and set the remote as upstream, use

    git push --set-upstream origin develop

To have this happen automatically for branches without a tracking
upstream, see 'push.autoSetupRemote' in 'git help config'.

robertito@gentoo ~/Repos/git101 $ git push --set-upstream origin develop
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 322 bytes | 322.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
remote:
remote: Create a pull request for 'develop' on GitHub by visiting:
remote:     https://github.com/toscosys/git101/pull/new/develop
remote:
To github.com:toscosys/git101.git
 * [new branch]      develop -> develop
branch 'develop' set up to track 'origin/develop'.
robertito@gentoo ~/Repos/git101 $ _
```

# Demo de GitHub

- Creamos nuestra cuenta en GitHub
- Configuramos nuestras Keys
- Creamos nuestro repositorio
- Añadimos Origin en nuestro repo
- Subimos y bajamos cambios
- **Empezamos a branchear**
- Empezamos a taggear

```
robertito@gentoo ~/Repos/git101 $ git checkout -b preprod
Switched to a new branch 'preprod'
robertito@gentoo ~/Repos/git101 $ git --no-pager branch
  develop
  main
* preprod
robertito@gentoo ~/Repos/git101 $

robertito@gentoo ~/Repos/git101 $ vim index.html
robertito@gentoo ~/Repos/git101 $ git commit -a -m "Creando feature"
[preprod ea22502] Creando feature
  1 file changed, 2 insertions(+), 1 deletion(-)
robertito@gentoo ~/Repos/git101 $ git push origin preprod
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 330 bytes | 330.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
remote:
remote: Create a pull request for 'preprod' on GitHub by visiting:
remote:     https://github.com/toscosys/git101/pull/new/preprod
remote:
To github.com:toscosys/git101.git
 * [new branch]      preprod -> preprod
robertito@gentoo ~/Repos/git101 $
```

# Demo de GitHub

- Creamos nuestra cuenta en GitHub
- Configuramos nuestras Keys
- Creamos nuestro repositorio
- Añadimos Origin en nuestro repo
- Subimos y bajamos cambios
- Empezamos a branchear
- **Empezamos a taggear**

```
robertito@gentoo ~/Repos/git101 $ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
robertito@gentoo ~/Repos/git101 $ git tag v0.1
robertito@gentoo ~/Repos/git101 $ git push origin v0.1
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:toscosys/git101.git
 * [new tag]      v0.1 -> v0.1
robertito@gentoo ~/Repos/git101 $ _
```

# Demo de GitHub

- Creamos nuestra cuenta en GitHub
- Configuramos nuestras Keys
- Creamos nuestro repositorio
- Añadimos Origin en nuestro repo
- Subimos y bajamos cambios
- Empezamos a branchear
- **Empezamos a taggear**

```
robertito@gentoo ~/Repos/git101 $ git --no-pager branch
  develop
* main
  preprod
robertito@gentoo ~/Repos/git101 $ git merge preprod
Updating 123698c..ea22502
Fast-forward
  index.html | 1 +
  1 file changed, 1 insertion(+)
robertito@gentoo ~/Repos/git101 $ 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
robertito@gentoo ~/Repos/git101 $ git push
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:toscosys/git101.git
  123698c..ea22502  main -> main
robertito@gentoo ~/Repos/git101 $ git tag v1.0
robertito@gentoo ~/Repos/git101 $ git push v1.0
fatal: 'v1.0' does not appear to be a git repository
fatal: Could not read from remote repository.

Please make sure you have the correct access rights
and the repository exists.
robertito@gentoo ~/Repos/git101 $ git push origin v1.0
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:toscosys/git101.git
 * [new tag]          v1.0 -> v1.0
```

# En la vida real lo usamos para...

---

- Desarrollo de aplicaciones
- History de configuraciones de infraestructura:
  - Entradas de DNS
  - Backup de config . de equipos de red
  - Configuración dinámica de servicios.
  - Rollback / restauracion estado anterior
- GitOps (Cambio de configuraciones y deployment de nuevas aplicaciones en tiempo real, con solo pushear código)

**¡GRACIAS !**



@donaiassembler (Camila)  
@pcarboni (Pablo)

*Agradecemos especialmente a @toscosys  
(Matias Sardisco)*

*...Co-autor intelectual del delito  
(esta presentación circa2022)*

