Control de versions

Marc Garcia Guerrero
DAM1
M03
2021/2022

Index

Introducció

Que es el control de versions?

El control de versions es la pràctica de rastrejar i gestionar el canvis en el codi d'un software. Els sistemes de control de versions són eines de programari que ajuden els equips de programari a gestionar els canvis en el codi font al llarg del temps. A mesura que els entorns de desenvolupament s'acceleren, els sistemes de control de versions ajuden els equips de software a treballar de forma més ràpida i intel·ligent. El software de control de versions permet que en cas d'error, és pugui recuperar codi o canvis que s'han efectuat en un projecte al llarg del temps.

Els desenvolupadors de software que treballen en equips estan escrivint contínuament nou codi o canviant el que ja existeix. El codi d'un projecte, una aplicació o un component de software normalment s'organitza en una estructura de carpetes o "arbre de fitxers". Un desenvolupador de l'equip podria estar treballant en una nova funció mentre un altre desenvolupador soluciona un error no relacionat canviant codi. Cada desenvolupador podria fer els canvis a diverses parts de l'arbre de fitxers i tot aixo es registraria en el software de control de versions, simplificant així el treball en equip i la resolució d'errors.

Pregunta 1. La meva història de por

En el meu cas, en un exercici de la UF3, al haver borrat un exercici des del portatil de clase, quan vaig fer un git pull/push desde una maquina virtual a l'ordinador de casa, vaig perdre tot el contingut de la carpeta de l'exercici, i el codi ya no estaba a cap dels 2 ordinadors, per sort mirant els commits de github vaig poder recuperar tot el codi que havia perdut.

Pregunta 2. Comparem sistemes de control de versions

Mercurial
El navegador marca com: Warning: Potential Security Risk Ahead i no permet entrar a la web
link: https://mercurial.selenic.com/

CVS

link: http://cvs.nongnu.org/
Descripció:
CVS is a version control system, an important component of Source Configuration Management (SCM). Using it, you can record the history of sources files, and documents. It fills a similar role to the free software RCS, PRCS, and Aegis packages.
CVS is a production quality system in wide use around the world, including many free software projects.
Bondats:

- It can run scripts which you can supply to log CVS operations or enforce site-specific policies.
- **Client/server CVS** enables developers scattered by geography or slow modems to function as a single team.
- In cases where several developers or teams want to each maintain their own version of the files, because of geography and/or policy, CVS's **vendor branches** can import a version from another team (even if they don't use CVS), and then CVS can merge the changes from the vendor branch with the latest files if that is what is desired.
- **Unreserved checkouts**, allowing more than one developer to work on the same files at the same time.CVS provides a flexible **modules database** that provides a symbolic mapping of names to components of a larger software distribution.
- CVS servers run on most unix variants, and clients for Windows NT/95, OS/2 and VMS are also available. CVS will also operate in what is sometimes called server mode against local repositories on Windows 95/NT.

Primera impressió:
Es un software antic que s'utilitza des de fa bastants anys, y que porta la darrera actualització va ser fa alguns anys.

SUBVERSION

link:
https://subversion.apache.org/
Descripció:
Is an open-source, centralized version control system characterized by its reliability as a safe haven for valuable data; the simplicity of its model and usage; and its ability to support the needs of a wide variety of users and projects, from individuals to large-scale enterprise operations.
Bondats:

- Most CVS features.
- Directories are versioned.

- Copying, deleting, and renaming are versioned.
- Free-form versioned metadata ("properties").
- Atomic commits.
- Branching and tagging are cheap (constant time) operations.
- Merge tracking.
- File locking.
- Symbolic links can be versioned.
- Executable flag is preserved.
- Apache network server option, with WebDAV/DeltaV protocol.
- Standalone server option (svnserve).
- And more…

Primera impressió:
Em sembla un software antic, igual que l'anterior però aquest sembla més actualitzat

PERFORCE/HELIXCORE

link:
https://www.perforce.com/
Descripció:
Perforce version control — Helix Core — tracks and manages changes to your source code, digital assets, and large binary files. It creates a single source of truth and collaboration platform that helps teams move faster, even as they develop complex products with thousands of contributors.
Bondats:
Totes les seves característiques estan descrites (i algunes comparant-se am git) aqui:
https://www.perforce.com/sites/default/files/pdfs/pf-core-product-brief-web.pdf
Primeres impressions:
Es un producte dirigit a empreses que millora alguns aspectes de git, per a que sigui més adequat per a grans projectes.

BAZAAR
link:
http://bazaar.canonical.com/en/
Descripció:
Bazaar is a version control system that helps you track project history over time and to collaborate easily with others. Whether you're a single developer, a co-located team or a community of developers scattered across the world, Bazaar scales and adapts to meet your needs. Part of the GNU Project, Bazaar is free software sponsored by Canonical.
Bondats:
- Version control for everyone
- Work offline
- Any workflow
- Cross platform support
- Rename tracking and smart merging
- High storage efficiency and speed
- Any workspace model
- Plays well with others

- Launchpad
- Plugins and bzrlib

Primera impressió:
Es un software que el seu desenvolupament es va aturar al 2017 i que segurament ara estarà desfasat en alguns aspectes.

DARCS

link:
http://darcs.net/
Descripció:
Darcs is a free and open source, cross-platform version control system, like git, mercurial or svn but with a very different approach: focus on changes rather than snapshots. Darcs offers a freer way of working, and a simpler user interface. Darcs *does not require a central server*, and works perfectly in offline mode
Bondats:

- Offline mode
- Local preparation
- Easy branching and merging
- Easy collaboration by e-mail
- Parallel development
- Cherry-picking
- Interactivity
- Hosting

Primeres impressions:
Podria ser una gran alternativa per a projectes petits o estudiants pero git ofereix moltes més utilitats.

GIT

link:
http://git-scm.com/
Descripció:
Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency.
Bondats:

- Branching and Merging
  The Git feature that really makes it stand apart from nearly every other SCM out there is its branching model.
  Git allows and encourages you to have multiple local branches that can be entirely independent of each other. The creation, merging, and deletion of those lines of development takes seconds.

- Small and Fast

  Git is fast. With Git, nearly all operations are performed locally, giving it a huge speed advantage on centralized systems that constantly have to communicate with a server somewhere.

- Distributed

  One of the nicest features of any Distributed SCM, Git included, is that it's distributed. This means that instead of doing a "checkout" of the current tip of the source code, you do a "clone" of the entire repository.

- Data Assurance

  The data model that Git uses ensures the cryptographic integrity of every bit of your project. Every file and commit is checksummed and retrieved by its checksum when checked back out. It's impossible to get anything out of Git other than the exact bits you put in.

- Staging Area

  Unlike the other systems, Git has something called the "staging area" or "index". This is an intermediate area where commits can be formatted and reviewed before completing the commit.

Primeres impressions:
Amb una primera impressió git em sembla el millor del sistemes de control de versions, sense haver provat els altres, principalment per tenir github

Pregunta 3. Quantes versions guardem?

versió	nombre de fitxers guardats
0	3
1	2
2	1
3	1
4	1

Pregunta 4. Configuració global

core.repositoryformatversion=0
core.filemode=true
core.bare=false
core.logallrefupdates=true
remote.origin.url=git@github.com:magagu77/introprg.git
remote.origin.fetch=+refs/heads/*:refs/remotes/origin/*
branch.main.remote=origin
branch.main.merge=refs/heads/main
user.name=Marc
user.email=marc.garcia0502@gmail.com
pull.ff=false

Pregunta 5. Ajut d'algunes comandes interessants

- clone

  Clones a repository into a newly created directory, creates remote-tracking branches for each branch in the cloned repository (visible using git branch --remotes), and creates and checks out an initial branch that is forked from the cloned repository's currently active branch.

- init

  This command creates an empty Git repository - basically a .git directory with subdirectories for objects, refs/heads, refs/tags, and template files. An initial HEAD file that references the HEAD of the master branch is also created.

- add

  This command updates the index using the current content found in the working tree, to prepare the content staged for the next commit.

- mv

  Move or rename a file, directory or symlink.

- reset

  In the first three forms, copy entries from <tree-ish> to the index. In the last form, set the current branch head (HEAD) to <commit>, optionally modifying index and working tree to match. The <tree-ish>/<commit> defaults to HEAD in all forms.

- rm

  Remove files from the index, or from the working tree and the index.

- log

  Shows the commit logs.

- status

  Displays paths that have differences between the index file and the current HEAD commit, paths that have differences between the working tree and the index file, and paths in the working tree that are not tracked by Git (and are not ignored by gitignore(5)).

- checkout

  Updates files in the working tree to match the version in the index or the specified tree.

- commit

  Create a new commit containing the current contents of the index and the given log message describing the changes.

Pregunta 6. Configuració inicial

Ha generat la mateixa sortida que a la pregunta 4
En el meu cas, bare.core=false vol dir que el meu repositori té un arbre de treball, en cas de ser bare.core=true, el directori de git ni tindria arbre de treball

Pregunta 7. Resum de comandes

-git clone
	clona un repositorio de git
-git init
	crea un nuevo repositorio de git
-git add
	añade loas archivos o carpetas indicados a un arbol de trabajo
-git log
	muestra el log de comentarios del repositorio
-git pull
	“Descarga” las actualizaciones del repositorio de git indicado
-git push
	Actualiza el repositorio del sservidor a partir de la copia local
-git commit
	Añade un comentario a una modificacion de un fichero que se haya añadido a un repositorio

Pregunta 8. Comptem objectes

Pas 0:
git count-objects
0 objects, 0 kilobytes

Pas 1:
touch test.txt
git count-objects
0 objects, 0 kilobytes

Pas 2:
git add test.txt
git count-objects
1 objects, 4 kilobytes

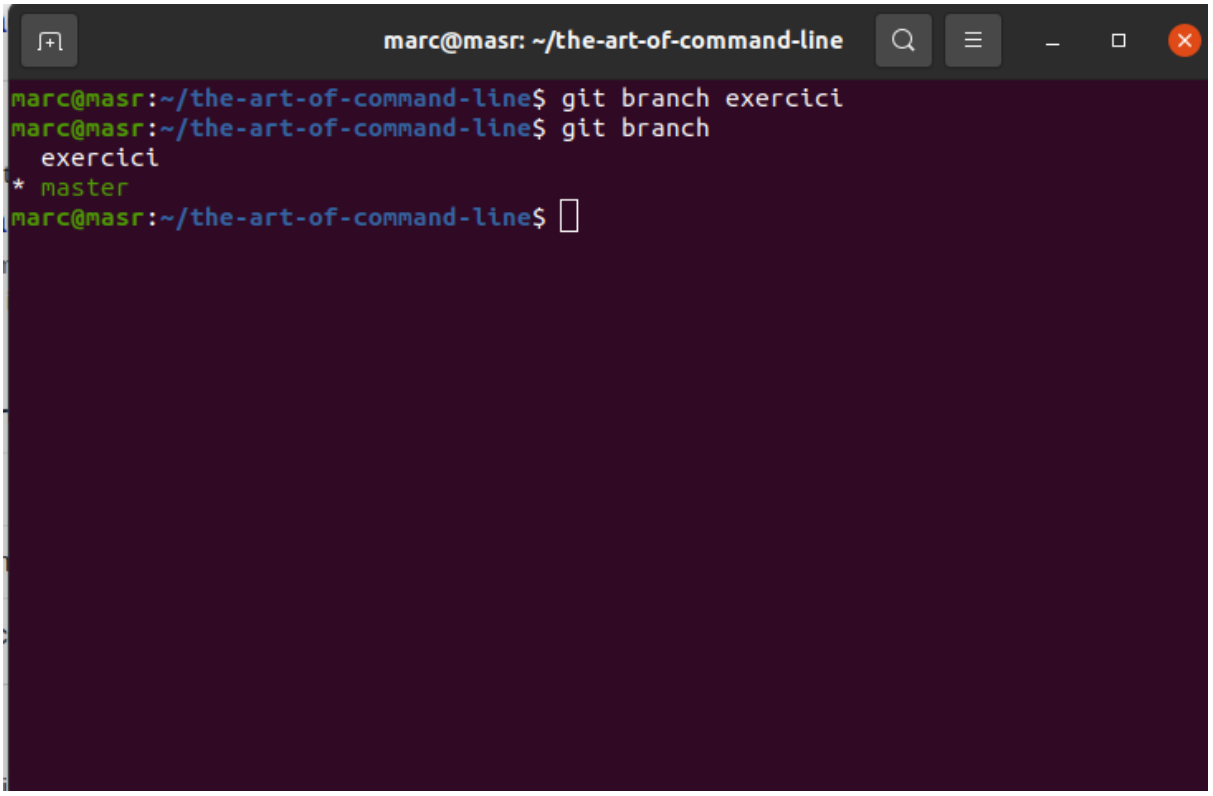
Pas 3:
git commit -am "Añadido test.txt"
git count-objects
3 objects, 12 kilobytes

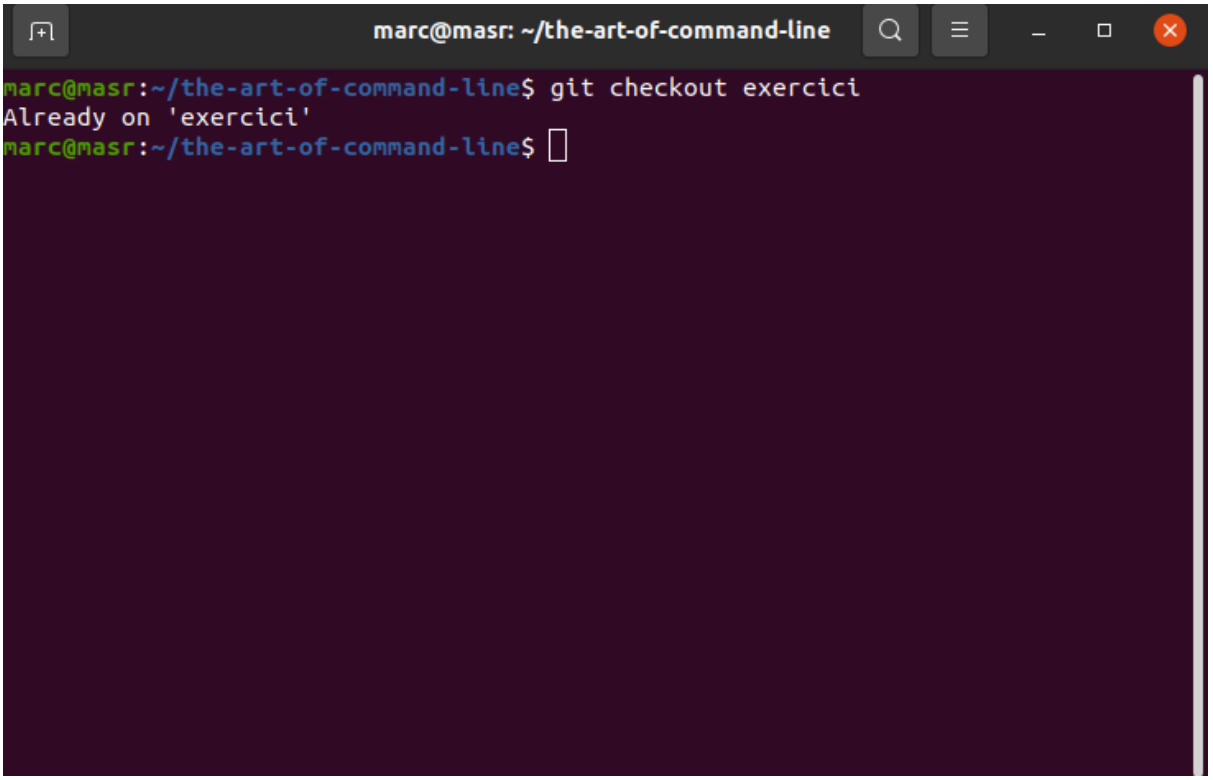
Pas 4:
git config --list > test.txt
git count-objects
3 objects, 12 kilobytes

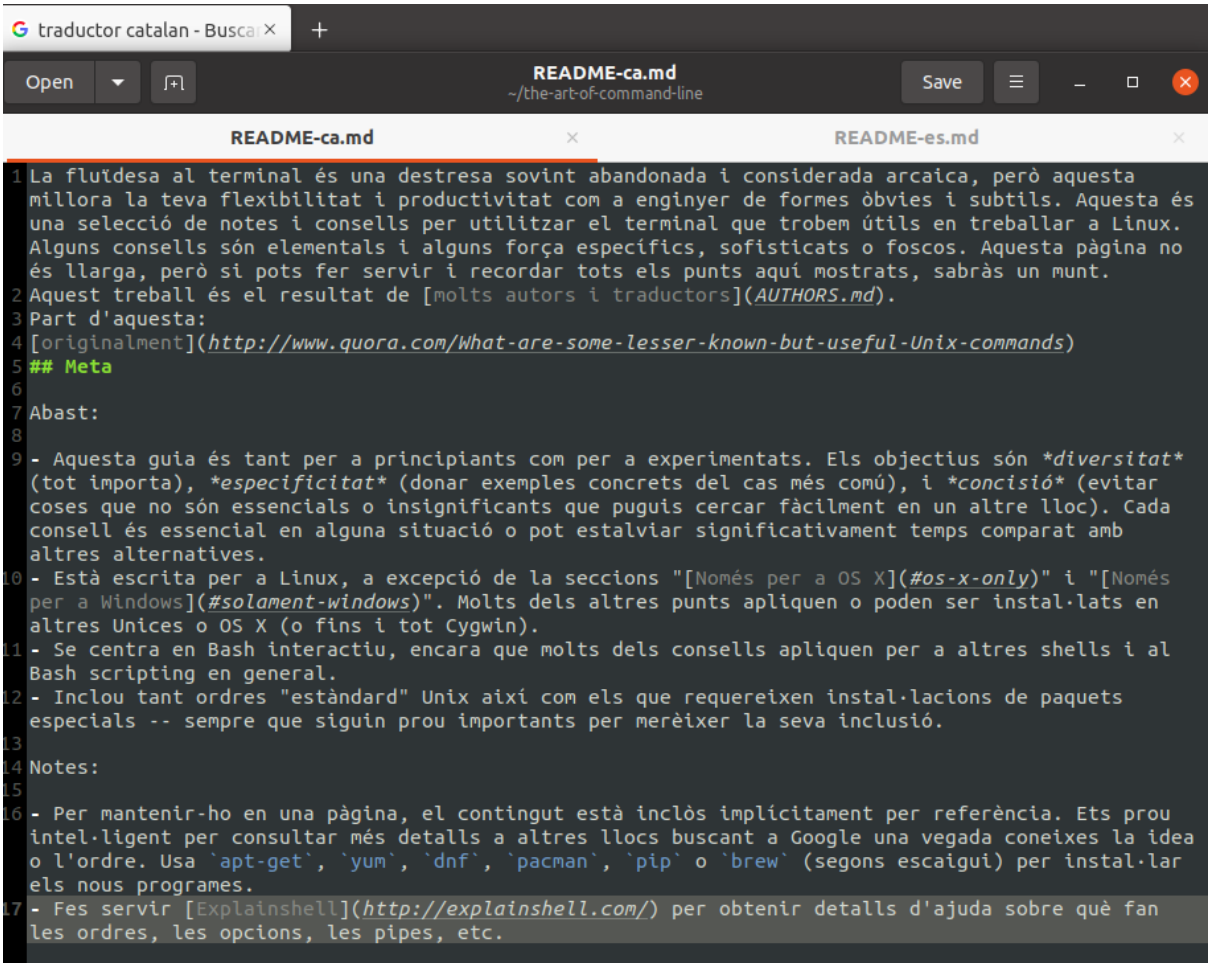
Pas 5:
git commit -am "txt modificado"
git count-objects
6 objects, 24 kilobytes

pas	objectes	kilobytes
0	0	0
1	0	0
2	1	4
3	3	12
4	3	12
5	6	24

Pregunta 11. L'art de la línia de comandes






traductor catalan - Busca
Open
README-ca.md
~/the-art-of-command-line
Save
README-ca.md
README-es.md
La fluïdesa al terminal és una destresa sovint abandonada i considerada arcaica, però aquesta millora la teva flexibilitat i productivitat com a enginyer de formes òbvies i subtils. Aquesta és una selecció de notes i consells per utilitzar el terminal que trobem útils en treballar a Linux. Alguns consells són elementals i alguns força específics, sofisticats o foscos. Aquesta pàgina no és llarga, però si pots fer servir i recordar tots els punts aquí mostrats, sabràs un munt.
Aquest treball és el resultat de [molts autors i traductors](AUTHORS.md).
Part d'aquesta:
[originalment](http://www.quora.com/What-are-some-lesser-known-but-useful-Unix-commands)
Meta
Abast:
- Aquesta guia és tant per a principiants com per a experimentats. Els objectius són *diversitat* (tot importa), *especificitat* (donar exemples concrets del cas més comú), i *concisió* (evitar coses que no són essencials o insignificants que puguis cercar fàcilment en un altre lloc). Cada consell és essencial en alguna situació o pot estalviar significativament temps comparat amb altres alternatives.
- Està escrita per a Linux, a excepció de la seccions "[Només per a OS X](#os-x-only)" i "[Només per a Windows](#solament-windows)". Molts dels altres punts apliquen o poden ser instal·lats en altres Unices o OS X (o fins i tot Cygwin).
- Se centra en Bash interactiu, encara que molts dels consells apliquen per a altres shells i al Bash scripting en general.
- Inclou tant ordres "estàndard" Unix així com els que requereixen instal·lacions de paquets especials -- sempre que siguin prou importants per merèixer la seva inclusió.
Notes:
- Per mantenir-ho en una pàgina, el contingut està inclòs implícitament per referència. Ets prou intel·ligent per consultar més detalls a altres llocs buscant a Google una vegada coneixes la idea o l'ordre. Usa `apt-get`, `yum`, `dnf`, `pacman`, `pip` o `brew` (segons escaigui) per instal·lar els nous programes.
- Fes servir [Explainshell](http://explainshell.com/) per obtenir detalls d'ajuda sobre què fan les ordres, les opcions, les pipes, etc.