



Git desde cero
Aprende Git de una vez



jesus@jesusamieiro.com



Características

- Introducción
- Instalación
- Configuración
- Conceptos básicos
- Ejercicio
 - Comandos básicos
 - Ramas

Qué é?



backup_20201016



2020_10_16_web.zip



cartel_v5.jpg



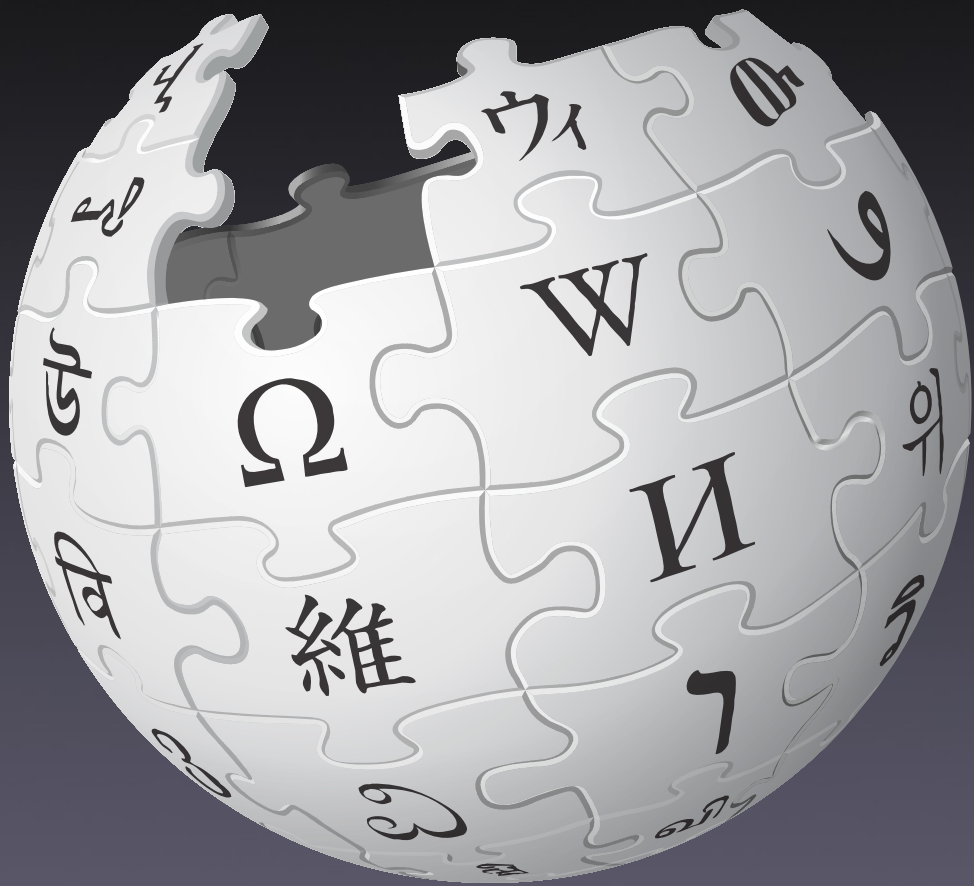
20201016_CV.docx



factura_final_4.ods

Problemas





VCS: Version Control System

SCM: Source Control Management

Para quen?

- Linguaxes interpretadas
- Linguaxes compiladas
- Diseñadores
- PDF, docx, odt,...

Commit

Tipos de SCM

Locais

SCCS

1972

RCS

1982

Equipo local



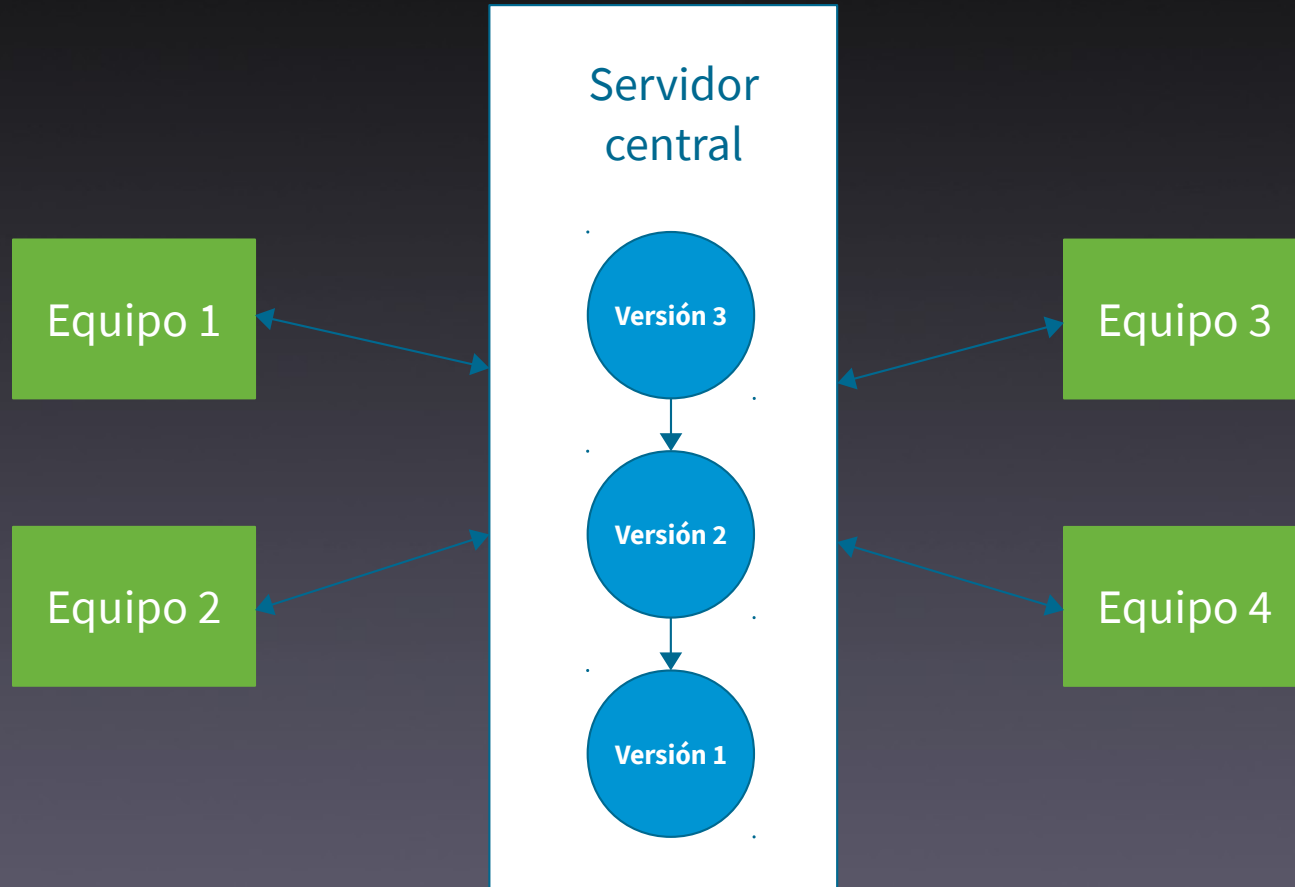
Cliente servidor

CVS

1990

Subversion

2000



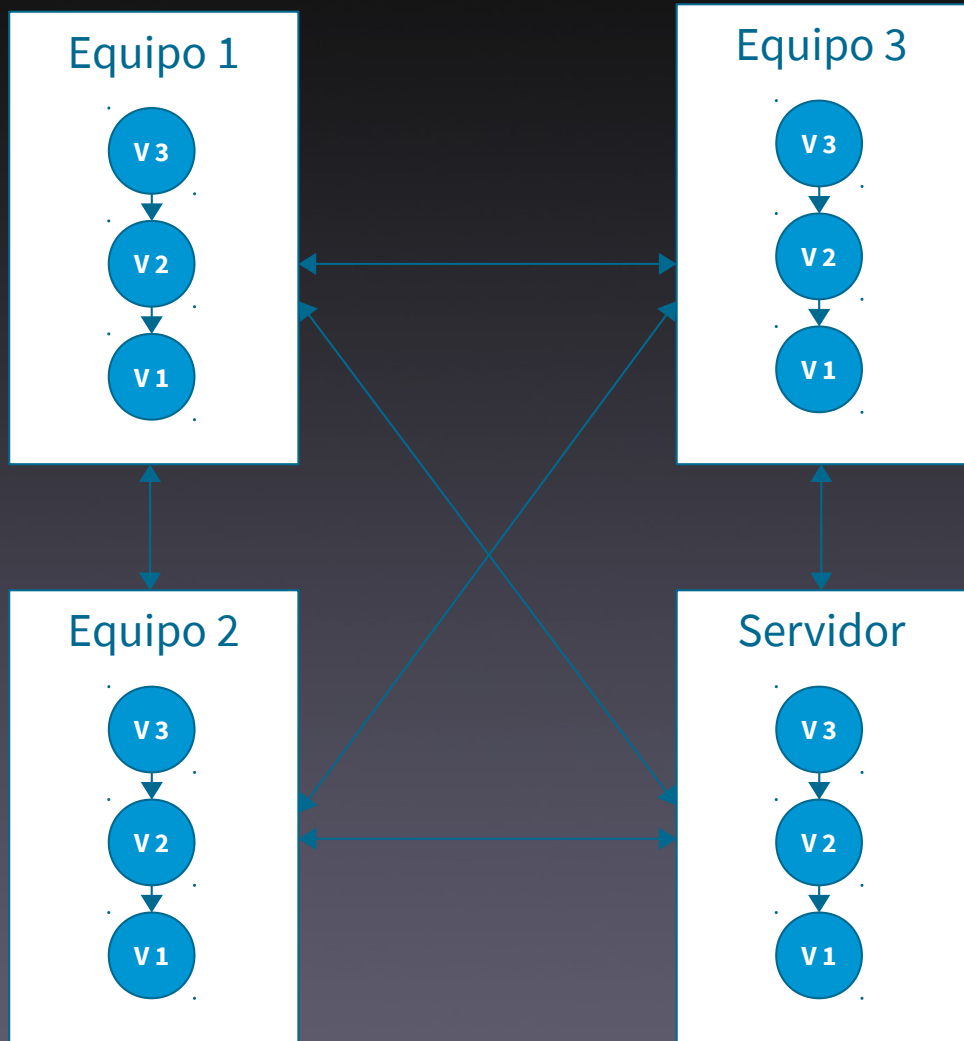
Distribuidos

Mercurial

2005

Git

2005



A creación de Git



BitKeeper

2002 - 2005

Características

- Rápido e escalable
- Cópia completa
- Desenvolvimento distribuído
- Trabalho local
- Favorece as ramas
- Instantâneas

Características

- Múltiples protocolos
- Robustez: SHA-1 (SHA-256)
- Libre
- Gratuito

Instalación (CLI)

- Windows: Git for Windows
- Linux:
 - `apt install git`
 - `yum/dnf install git`
- OSX: `brew install git`

Axuda

git help [comando]
git help init

git [commando] -help
git init --help

man git-[comando]
man git-init

man git [commando]
man git init

Configuración inicial de Git

Configuración

- Sistema
- Usuario
- Repositorio

Configuración

- Sistema
 - `git config --system`
 - `/etc/gitconfig`
 - `C:\Program Files\Git\etc\gitconfig`

Configuración

- Usuario
 - `git config --global`
 - `~/.gitconfig`
 - `C:\Users\miusuario\.gitconfig`

Configuración

- Repositorio (por defecto)
 - `git config --local`
 - `.git/config`

Configuración

```
git config --global user.name "Jesús Amieiro"
```

```
git config --global user.email amieiro@gmail.com
```

Configuración

```
git config --global core.editor emacs
```

```
git config --global core.editor vim
```

```
git config --global core.editor Code.exe
```

Configuración

```
git config --global merge.tool vimdiff
```

```
git config --global color.ui true
```

```
git config --global core.autocrlf true
```

```
git config --global core.autocrlf input
```

Configuración

```
git config --global user.name
```

```
git config --global user.email
```

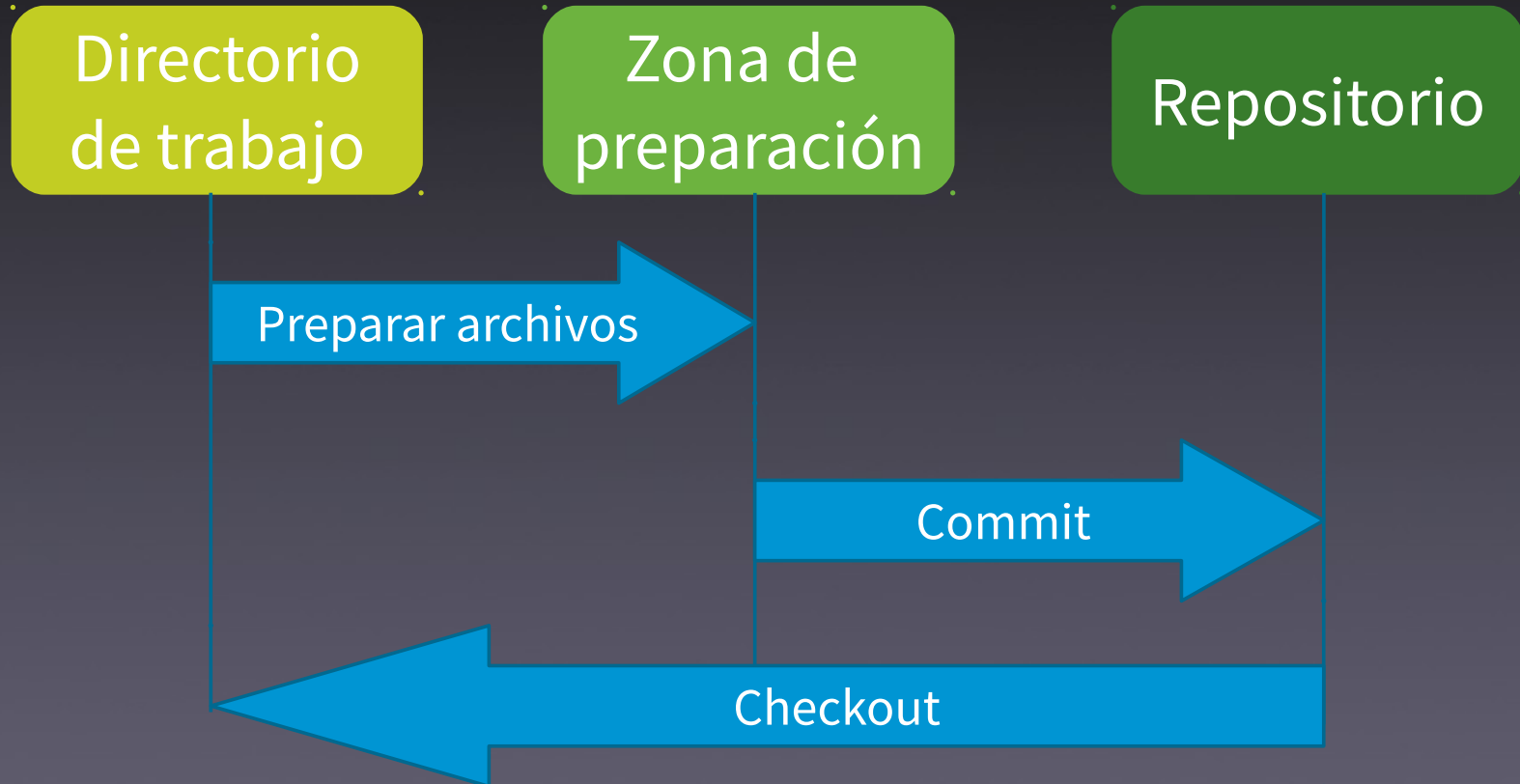
```
git config --list
```


Conceptos básicos

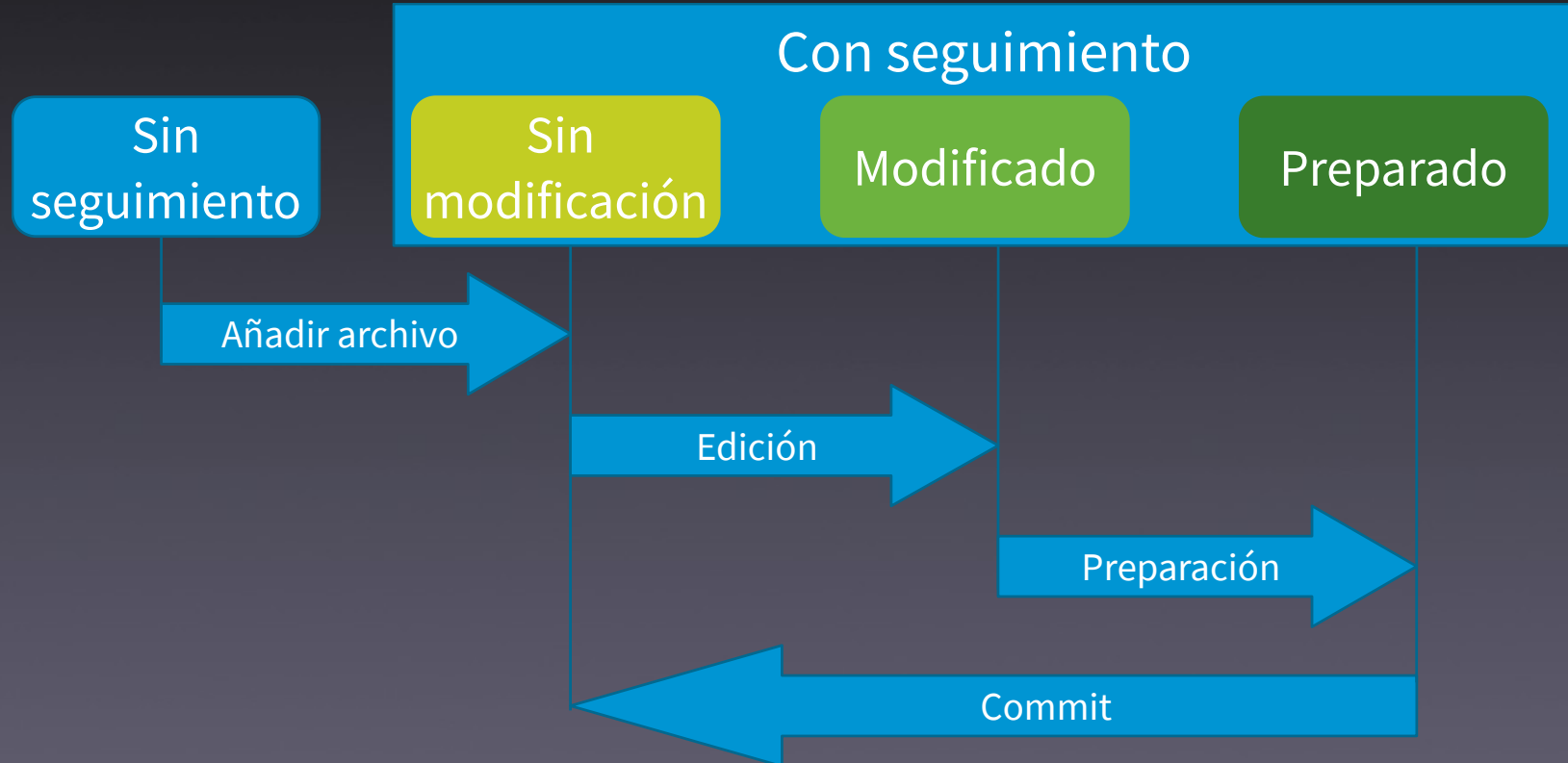
Commit

Repositorio

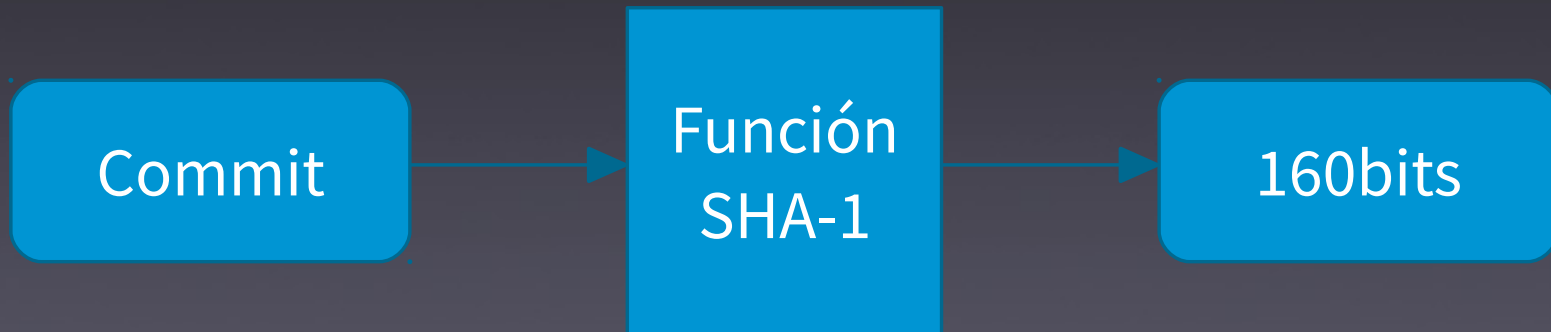
Zonas en Git



Estados e flujo

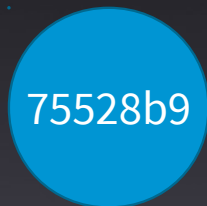


SHA-1

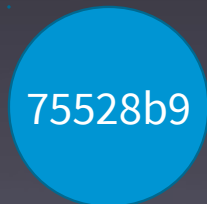


HEAD

Commit 1



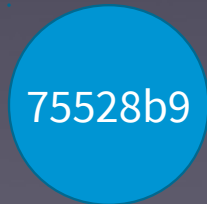
Commit 1



Commit 2



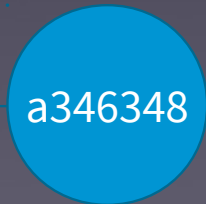
Commit 1



Commit 2



Commit 3



Exercicio

Inicialización₍₁₎

```
cd  
git clone git@github.com:twbs/bootstrap.git bs  
git clone https://github.com/twbs/bootstrap.git  
ls bootstrap  
ls -la bootstrap
```

Inicialización₍₂₎

```
mkdir ~/exemplo
```

```
cd ~/exemplo
```

```
git status
```

```
ls -la
```

```
git init
```

```
ls
```

Inicialización₍₃₎

```
ls -la
```

```
ls -la .git
```

```
cat .git/config
```

```
git status
```

Xestión arquivos⁽¹⁾

```
touch primeiro_arquivo.txt
```

```
ls
```

```
git status
```

```
git add primeiro_arquivo
```

```
# Alternativa: git add .
```

```
git status
```

```
git commit -m "Engado o primeiro arquivo baleiro"
```

```
git status
```

Xestión arquivos₍₂₎

```
touch segundo_arquivo.txt
```

```
touch terceiro_arquivo.txt
```

```
git status
```

```
git add segundo_arquivo.txt
```

```
git status
```

```
git commit -m "Engado o segundo arquivo"
```

```
git status
```

```
git add terceiro_arquivo.txt
```

Xestión arquivos₍₃₎

git status

git commit -m "Engado o terceiro arquivo"

git status

git log

git log --oneline

Edición arquivos ⁽¹⁾

```
echo "Creo unha primeira liña no primeiro arquivo" >
```

```
primeiro_arquivo.txt
```

```
git status
```

```
git add primeiro_arquivo.txt
```

```
git status
```

```
echo "Creo unha primeira liña no segundo arquivo." >
```

```
segundo_arquivo.txt
```

```
echo "Creo unha primeira liña no terceiro arquivo." >
```

```
terceiro_arquivo.txt
```

Edición arquivos ⁽²⁾

git status

git add segundo_arquivo.txt

git status

git commit -m "Introduzo unha liña no primeiro e no
segundo arquivo"

git status

git log

git log --oneline

git add terceiro_arquivo.txt

Edición arquivos ⁽³⁾

```
git commit -m "Introduzo unha liña no terceiro arquivo"
```

```
git status
```

```
git log
```

```
git log --oneline
```

Ver cambios⁽¹⁾

```
echo "Creo unha segunda liña no primeiro arquivo" >>
```

```
primeiro_arquivo.txt
```

```
cat primeiro_arquivo.txt
```

```
git status
```

```
git diff
```

```
echo "Creo unha segunda liña no segundo arquivo" >>
```

```
segundo_arquivo.txt
```

```
cat segundo_arquivo.txt
```

Ver cambios⁽²⁾

git status

git diff

git diff segundo_arquivo.txt

Ver cambios ⁽³⁾

```
git add segundo_arquivo.txt
```

```
git status
```

```
git diff
```

```
git diff --staged
```

```
git diff --cached
```

```
git diff HEAD
```

```
git add primeiro_arquivo.txt
```

```
git status
```

Ver cambios₍₄₎

git diff

git diff HEAD

git diff --staged

git commit -m "Engado liñas no primeiro e no
segundo arquivo"

git status

git diff

git diff HEAD

git diff --staged

Ver cambios ⁽⁵⁾

```
git log --oneline
```

```
git diff $SHA_1..$SHA2
```

```
git diff $SHA_1..$SHA2 -- segundo_arquivo.txt
```

Desfacer cambios ⁽¹⁾

```
git status # Zona de traballo
```

```
echo "Engado unha terceira liña ao primeiro arquivo" >>
```

```
primeiro_arquivo.txt
```

```
cat primeiro_arquivo.txt
```

```
git status
```

```
git diff
```

```
git restore primeiro_arquivo.txt
```

```
cat primeiro_arquivo.txt
```

```
git status
```

Desfacer cambios ⁽²⁾

```
git status # Índice
```

```
echo "Engado unha terceira liña ao primeiro arquivo" >>
```

```
primeiro_arquivo.txt
```

```
cat primeiro_arquivo.txt
```

```
git status
```

```
git add primeiro_arquivo.txt
```

```
git status
```

```
git restore --staged primeiro_arquivo.txt
```

```
git status
```

```
cat primeiro_arquivo.txt
```


Desfacer cambios ⁽³⁾

```
git log --oneline
```

```
cat primeiro_arquivo.txt
```

```
git checkout $SHA_commit primeiro_arquivo.txt
```

Ignorar arquivos ⁽¹⁾

```
touch arquivo_temporal.txt
```

```
git status
```

```
touch .gitignore
```

```
git status
```

```
echo "arquivo_temporal.txt" >> .gitignore
```

```
git status
```

```
mkdir log
```

```
touch log/untrack01.log
```

```
mkdir images
```

Ignorar archivos ⁽²⁾

```
touch images/untrack01.jpg
```

```
touch images/logo.png
```

```
echo "*.zip" >> .gitignore
```

```
echo "*.gz" >> .gitignore
```

```
echo "log/*.log" >> .gitignore
```

```
echo "log/*.log[0-9]" >> .gitignore
```

```
echo "images/*" >> .gitignore
```

```
echo "! images/logo.png" >> .gitignore
```

Ignorar arquivos ⁽³⁾

```
git status
```

```
git add .gitignore
```

```
git commit -m "Engado o arquivo .gitignore"
```

```
git status
```

```
git log --oneline
```

<https://github.com/github/gitignore>

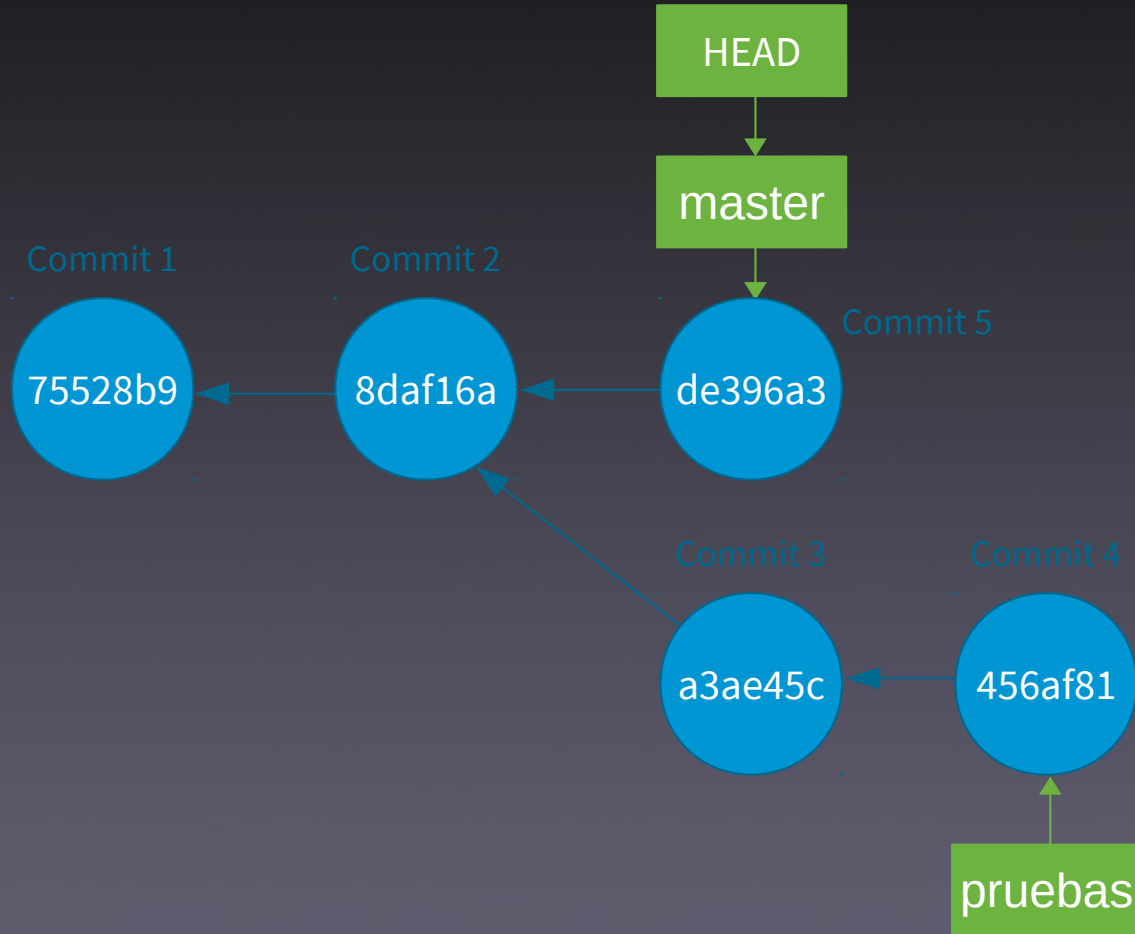
Ignorar arquivos ⁽⁴⁾

Se existe o arquivo

git rm arquivo_temporal.txt # Borro o arquivo

git rm --cached arquivo_temporal.txt

Ramas (1)



Ramas (2)

Criar unha rama

git branch

ls -la .git

cat .git/HEAD

ls -la .git/refs/heads/

cat .git/refs/heads/master

git log -1

git branch desenvolvimento

Ramas ⁽³⁾

git branch

ls -la .git/refs/heads/

cat .git/refs/heads/master

cat .git/refs/heads/desenvolvimento

Ramas ⁽⁴⁾

Cambio de rama

git branch

cat .git/HEAD

git switch desenvolvimento

git checkout desenvolvimento

cat .git/HEAD

Criar unha rama e cambiarse a ela

git checkout -b experimento

Ramas (5)

Trabalho na nova rama

echo "Insero unha liña no primeiro arquivo" >>

primeiro_arquivo.txt

git status

git commit -am "Introduzo unha nova liña no
primeiro_arquivo.txt da rama de desenvolvemento"

git log --oneline -5

git checkout master

git log --oneline -5

Ramas (6)

```
cat primeiro_arquivo.txt
```

```
git checkout desenvolvimento
```

```
cat primeiro_arquivo.txt
```

```
git log --oneline -5
```

```
cat .git/HEAD
```

```
git show HEAD
```

Ramas ⁽⁷⁾

Fusión

```
git log --oneline -6 --graph --all --decorate
```

```
git checkout master
```

```
git merge desenvolvimento
```

Problemas de fusión