



git





jesus@jesusamieiro.com





QUADR ALIA



Contido

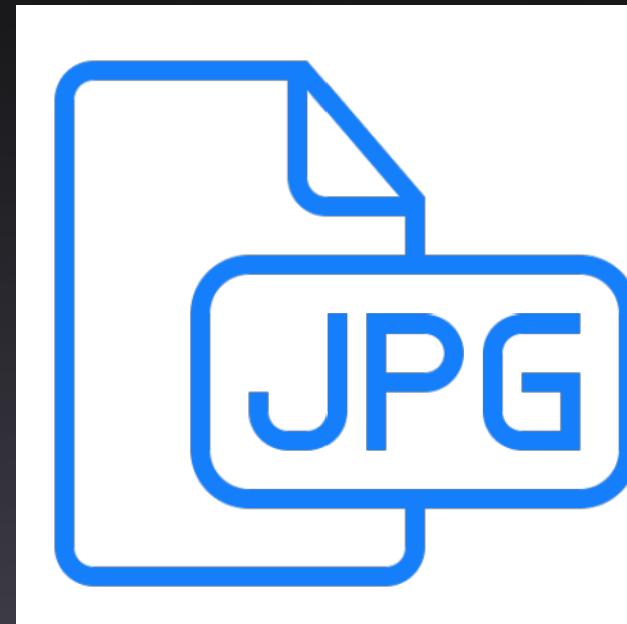
- Introducción
- Instalación
- Configuración
- Conceptos básicos
- Exercicio
 - Comandos básicos
 - Ramas
 - Repositorios remotos



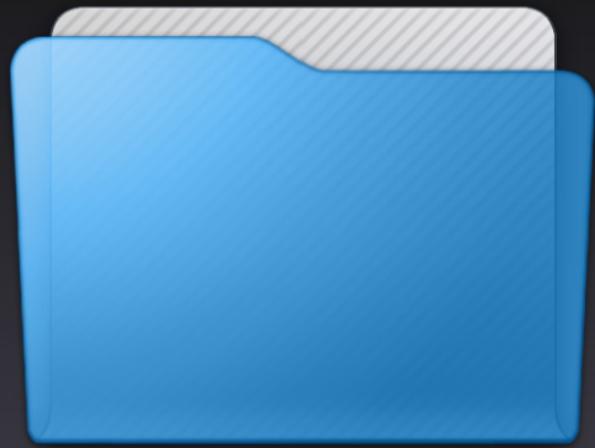
Que é?



Orzamento_v2.doc

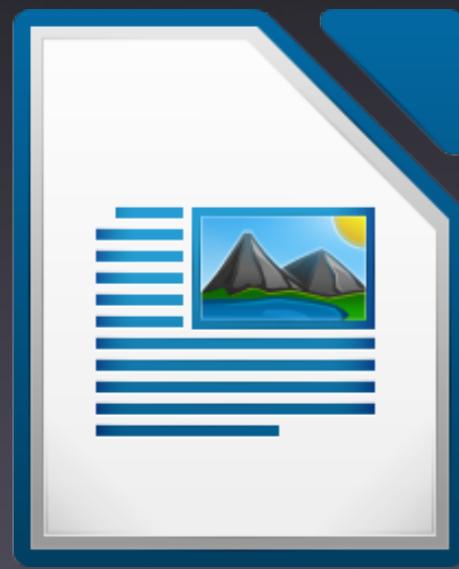


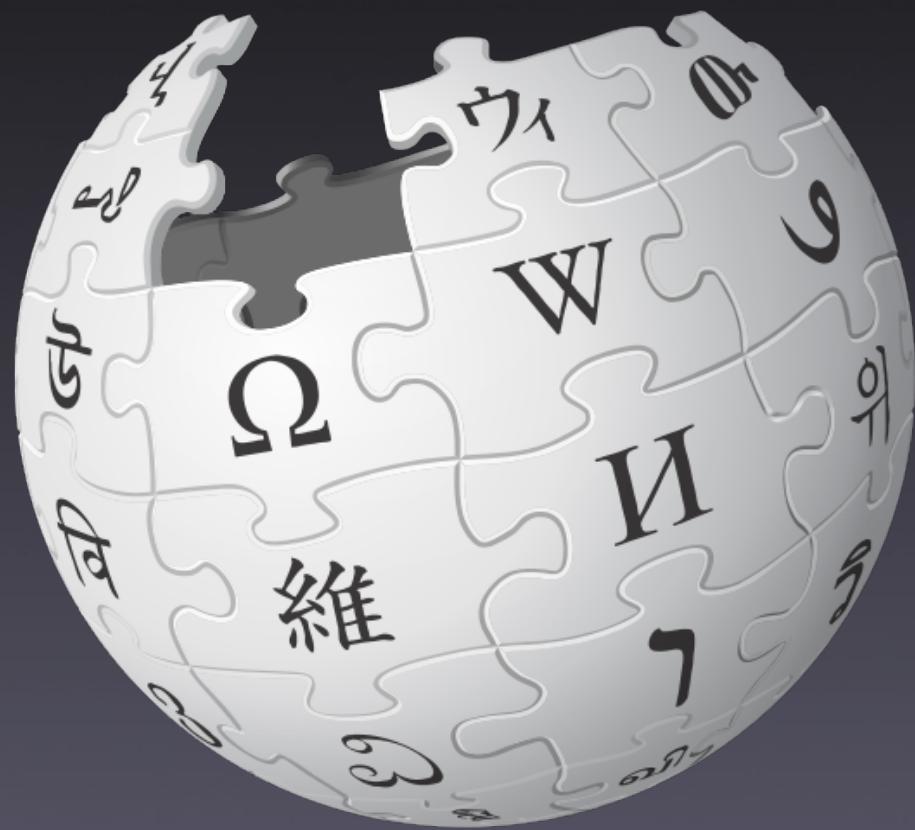
Cartel_v5.jpg



2010_05_17_web

Problemas





VCS: Version Control System

SCM: Source Control Management



Commit



Para quen?

- Linguaxes interpretados
- Linguaxes compilados
- Deseñadores
- PDF, docx, odt,...

Tipos de SCM

Locais

SCCS

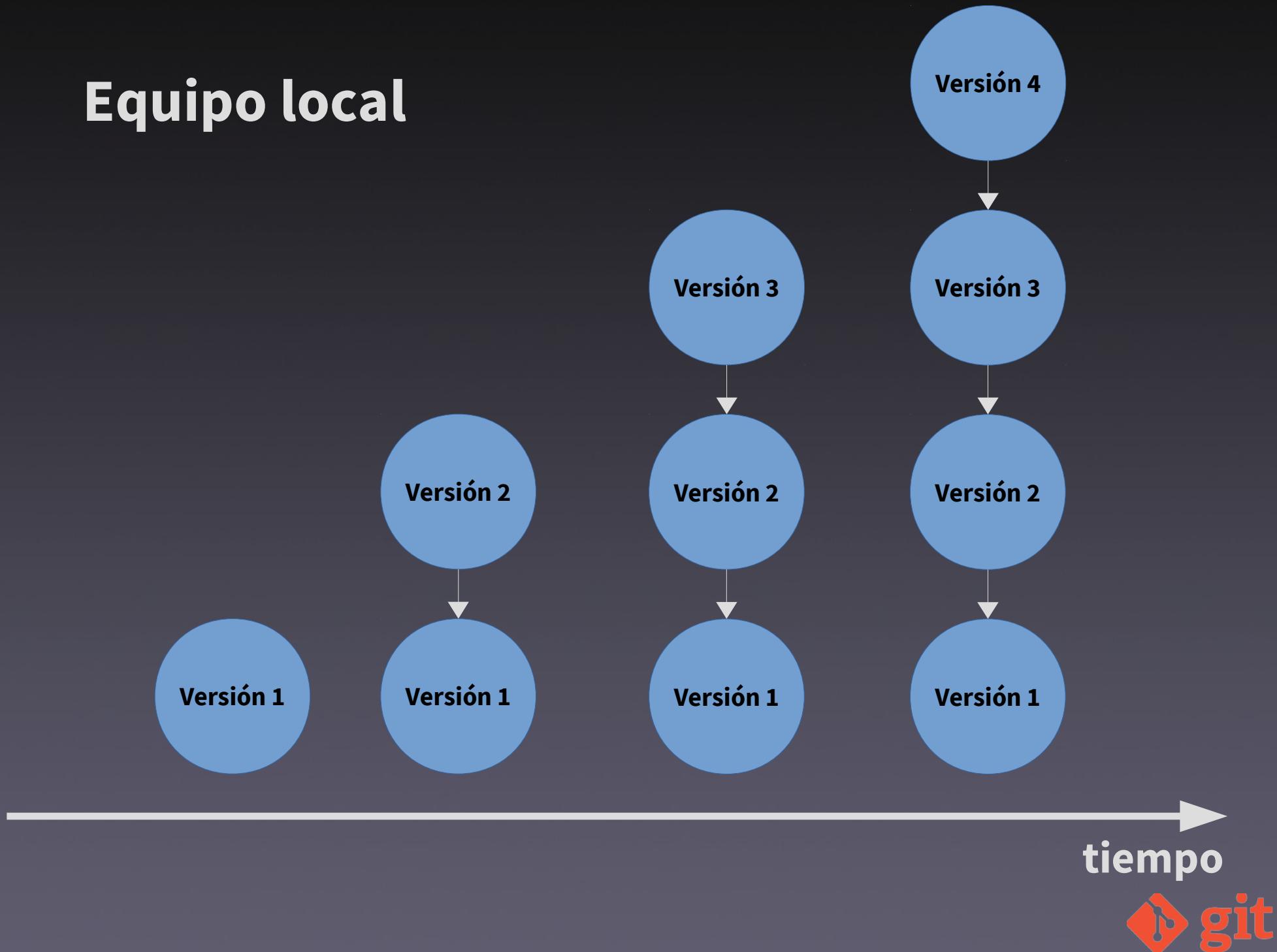
1972

RCS

1982



Equipo local



Cliente servidor

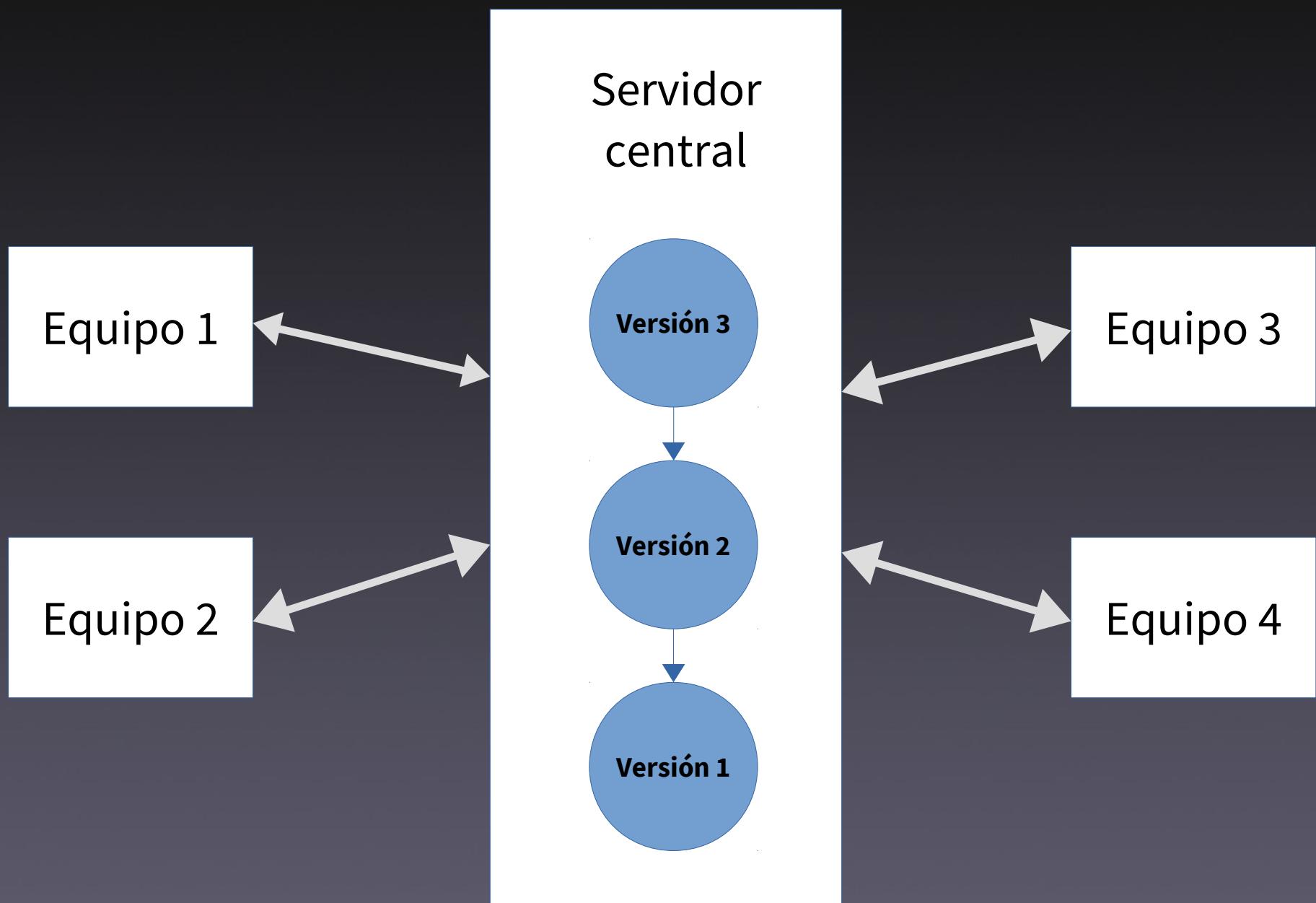
CVS

1990

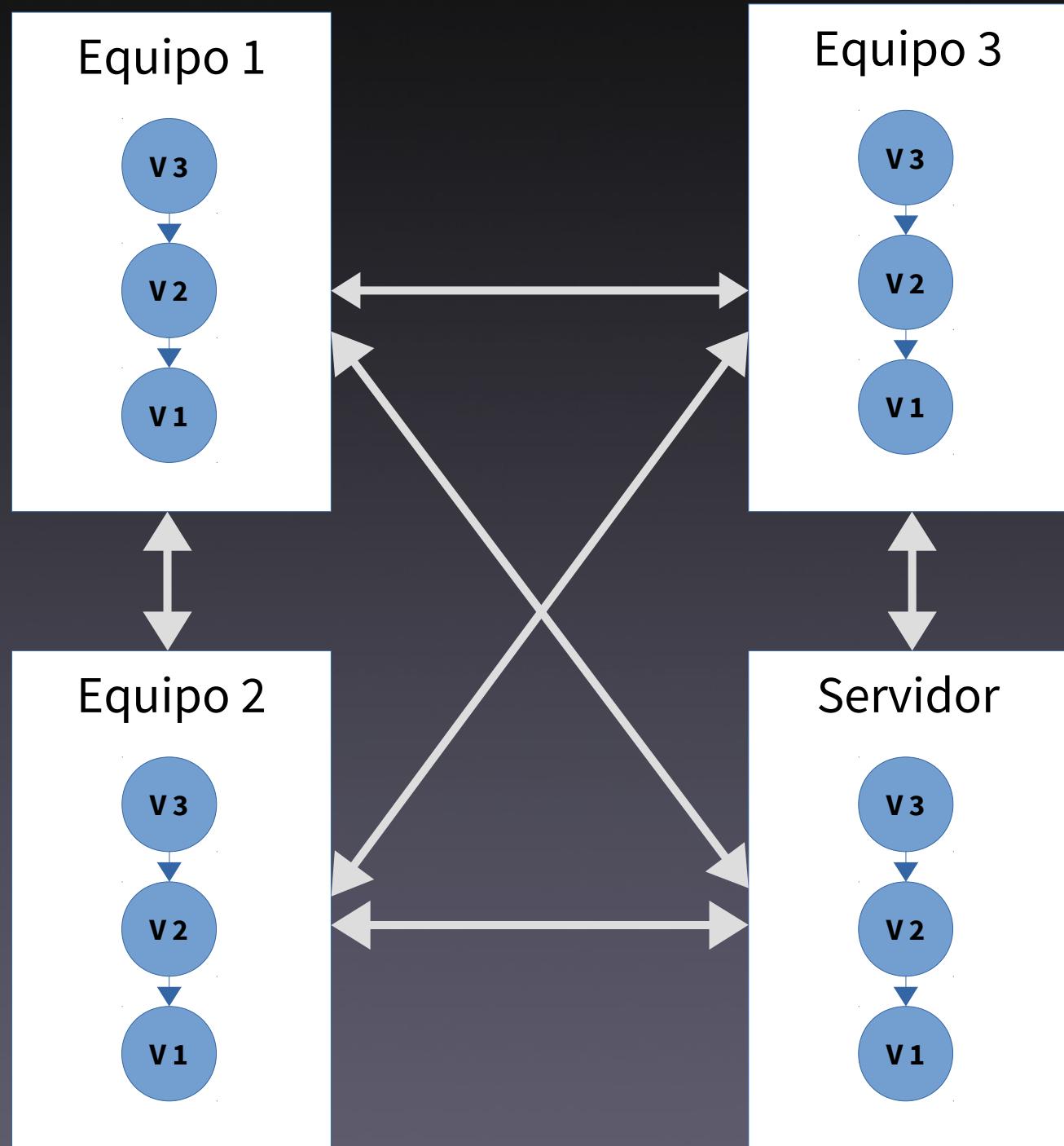
Subversion

2000





Distribuidos



A creación de Git



BitKeeper

2000



Git. Características

- Rápido e escalable
- Copia completa
- Desenvolvimento distribuído
- Traballo local
- Alenta as ramas
- Instantáneas

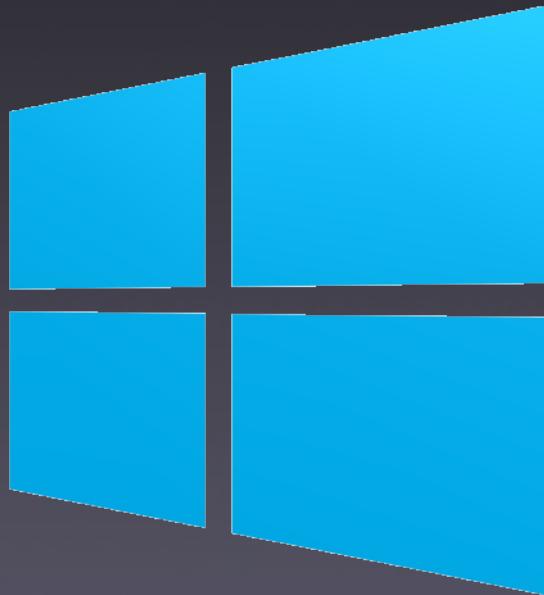


Git. Características (II)

- Múltiples protocolos
- Robustez: SHA-1
- Libre
- Gratuito



Versións



Windows

Git for Windows





apt-get install git





yum install git



Mac

Git-osx-installer
MacPorts



Obter axuda

`git help [comando]`

`git help init`

`git [comando] --help`

`git init --help`



`man git-[comando]`

`man git-init`

`man git [comando]`

`man git init`



Subcomandos
git help -a

Guías de conceptos
git help -g

git help [concepto]
git help glossary



Configuración inicial de Git

Configuración

- Sistema
 - /etc/gitconfig
 - C:\Program Files (x86)\Git\etc\gitconfig
- git config --system

Configuración

- Usuario

- `~/.gitconfig`
- `C:\Users\Mi usuario\.gitconfig`
- `git config --global`



Configuración

- Repositorio
 - .git/config
 - git config --local



Configuración

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

```
git config --global user.email "jesus@jesusamieiro.com"
```



Configuración

git config --global core.editor emacs

git config --global core.editor vim

git config --global core.editor notepad.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

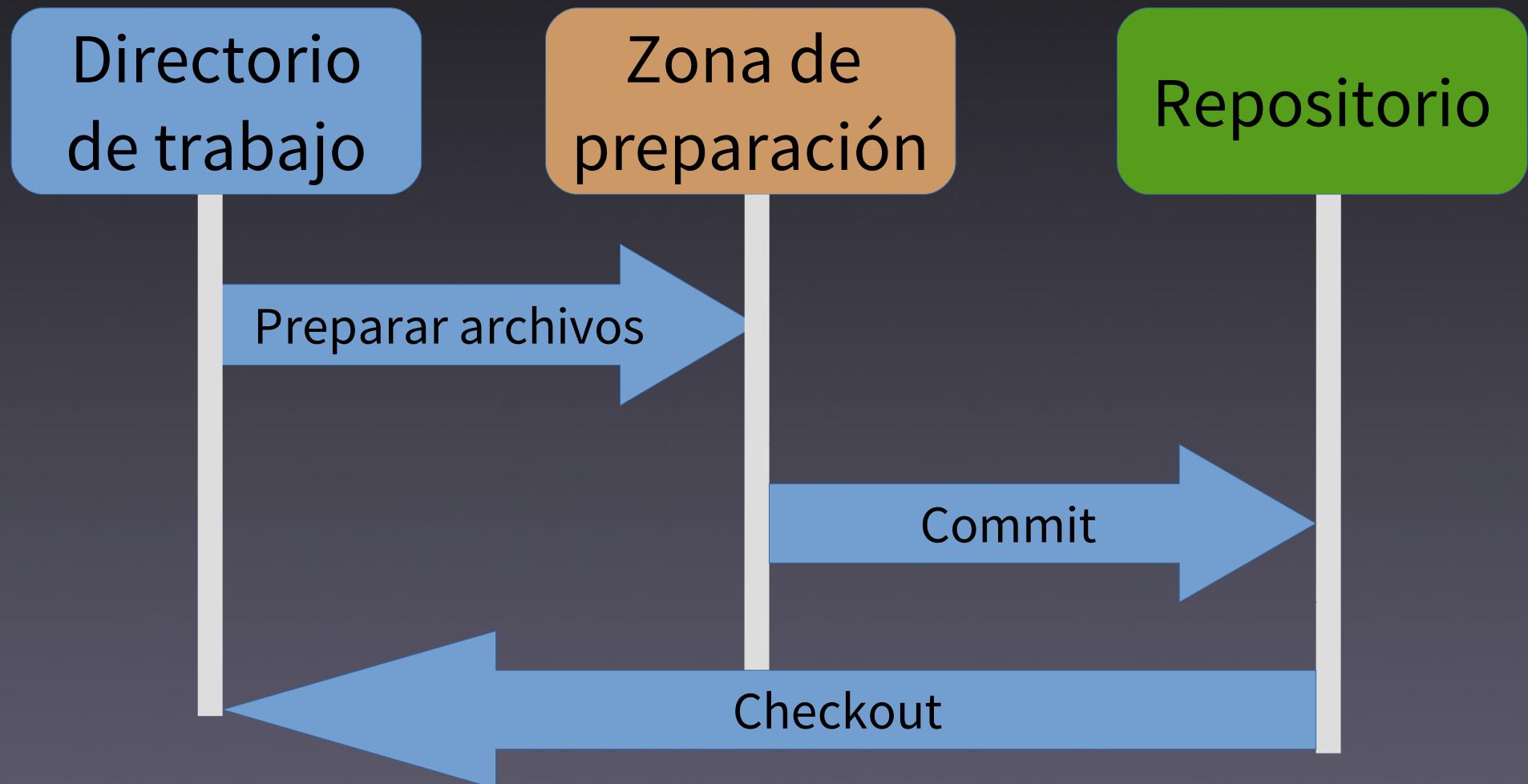
Repositorio



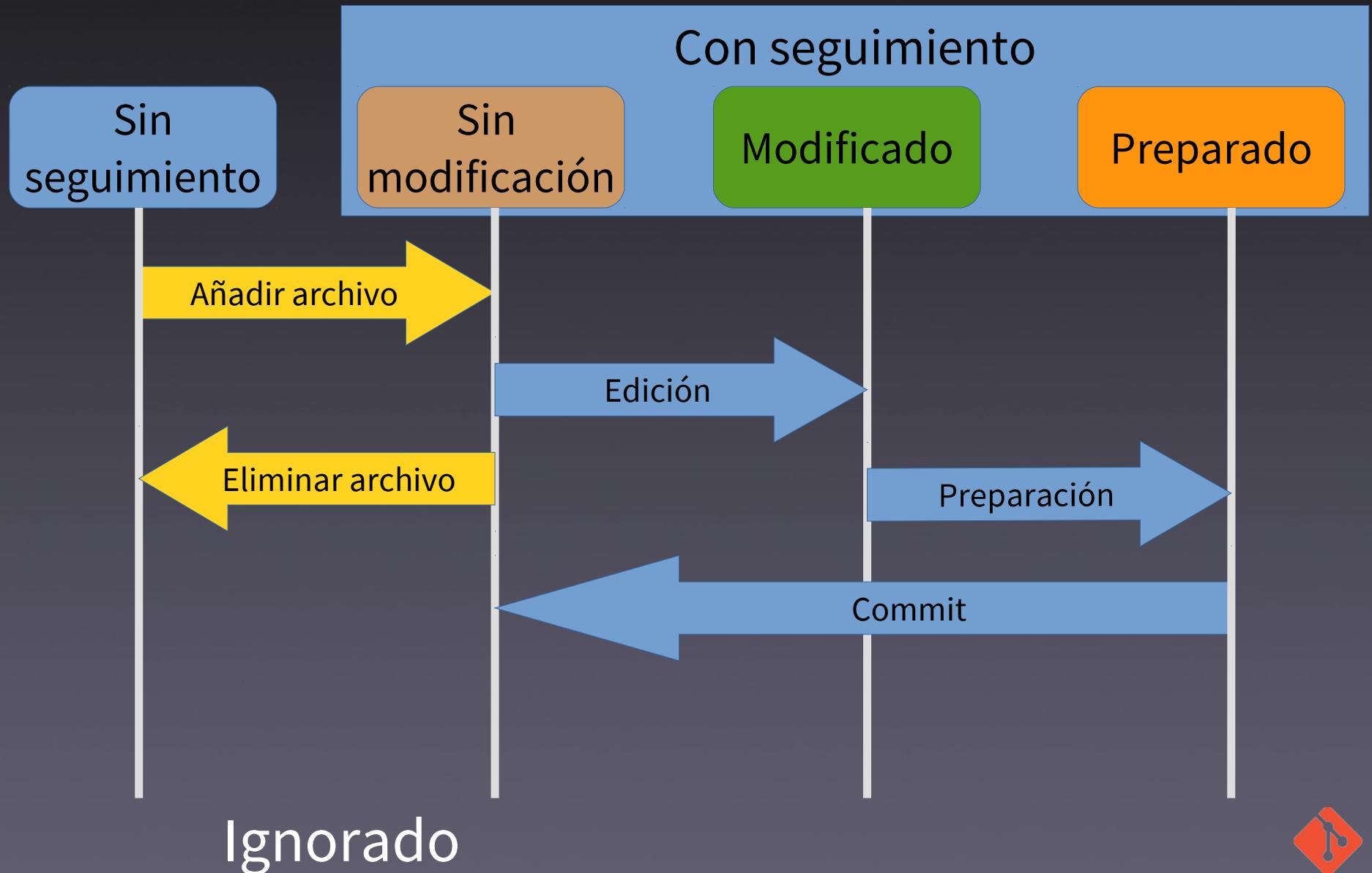
Commit



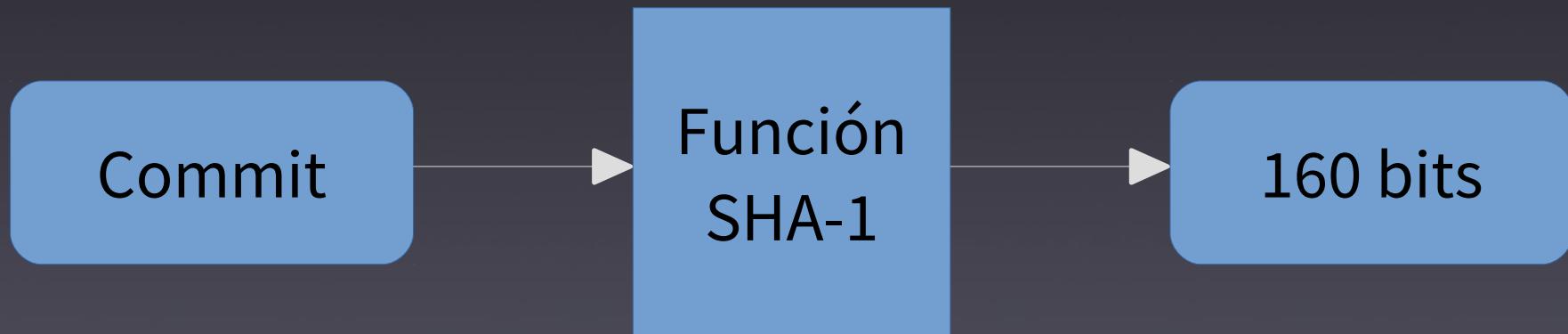
Zonas en Git



Estados e fluxo



SHA-1



HEAD

Commit 1

75528b9

Commit 1

75528b9

Commit 2

8daf16a

Commit 1

75528b9

Commit 2

8daf16a

Commit 3

a346348



EXERCICIO

Inicialización

- Desde repositorio remoto: git clone

```
cd ~
```

```
git clone git@github.com:twbs/bootstrap.git
```

```
git clone https://github.com/twbs/bootstrap.git
```

```
ls bootstrap
```

```
ls -la bootstrap
```



Inicialización (II)

- Local: git init

```
mkdir ~/exemplo
```

```
cd ~/exemplo
```

```
git status
```

```
ls -la
```

git init

```
ls
```

```
ls -la
```



Inicialización (III)

ls -la .git

cat .git/config

git status



Xestión arquivos

```
touch primeiro_arquivo.txt
```

```
ls
```

```
git status
```

```
git add primeiro_arquivo.txt
```

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

```
git status
```

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

```
git status
```



Xestión arquivos (II)

```
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 (III)

git status

git commit -m "Engado o terceiro arquivo"

git status

git log

git log --oneline



Edición arquivos

```
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 (II)

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 (III)

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

```
git status
```



Ver cambios

```
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 (II)

git status

git diff

git diff segundo_arquivo.txt



Ver cambios (III)

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 (IV)

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 (V)

git diff \$SHA_1..\$SHA2

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



Desfacer cambios

```
# Zona de traballo
```

```
git status
```

```
echo "Engado unha terceira liña ao primeiro arquivo" >>  
primeiro_arquivo.txt
```

```
cat primeiro_arquivo.txt
```

```
git status
```

```
git diff
```

```
git checkout -- primeiro_arquivo.txt
```

```
cat primeiro_arquivo.txt
```

```
git status
```



Desfacer cambios (II)

Índice

git status

```
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 reset HEAD primeiro_arquivo.txt

git status

cat primeiro_arquivo.txt



Desfacer cambios (III)

```
git log --oneline
```

```
cat primeiro_arquivo.txt
```

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

```
# Dous commit antes
```

```
cat primeiro_arquivo.txt
```

```
git diff --staged
```

```
git checkout -- primeiro_arquivo.txt
```

```
# git checkout HEAD primeiro_arquivo.txt
```

```
git status
```



Ignorar archivos

```
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 (II)

```
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 archivos (III)

git status

git add .gitignore

git commit -m "Engado o arquivo .gitignore"

git status

git log --oneline

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



Ignorar archivos (IV)

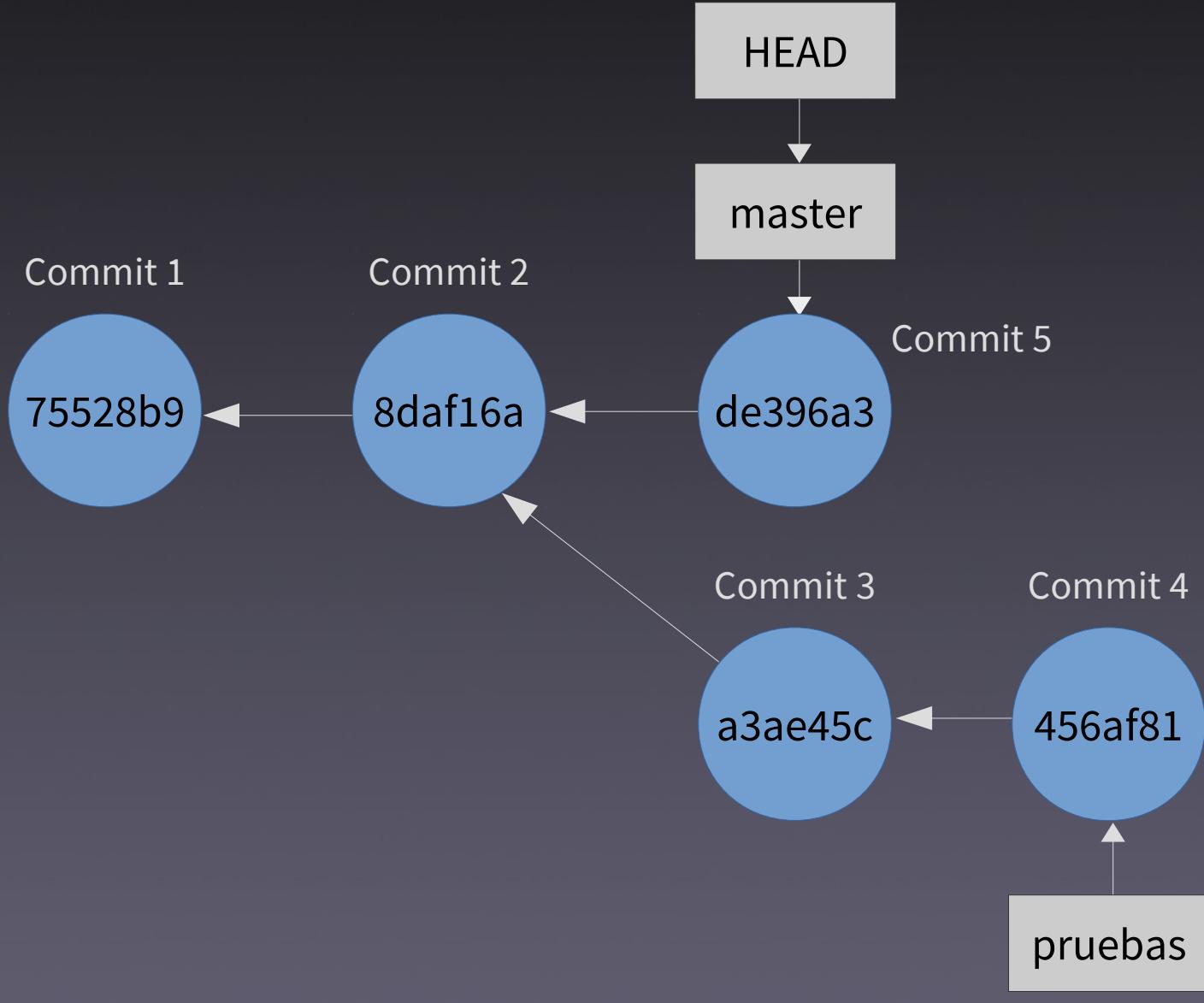
Se existe o arquivo

```
git rm arquivo_temporal.txt # Borro o arquivo
```

```
git rm --cached arquivo_temporal.txt
```



Ramas



Ramas (II)

```
# Crear 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 (III)

git branch

ls -la .git/refs/heads/

cat .git/refs/heads/master

cat .git/refs/heads/desenvolvimento



Ramas (IV)

Cambio de rama

git branch

cat .git/HEAD

git checkout desenvolvimento

cat .git/HEAD

Crear unha rama e cambiarse a ela

git checkout -b experimento



Ramas (V)

Traballo na nova rama

```
echo "Inserto unha liña no primeiro arquivo" >>  
primeiro_arquivo.txt
```

git status

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

git log --oneline -5

git checkout master

git log --oneline -5



Ramas (VI)

cat primeiro_arquivo.txt

git checkout desenvolvimento

cat primeiro_arquivo.txt

git log --oneline -5

cat .git/HEAD

git show HEAD



Ramas (VII)

Fusión

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

```
git checkout master
```

```
git merge desenvolvimento
```

Problemas de fusión



Repositorios remotos

```
# Usuario 1
```

```
# Creo e comproto o repositorio remoto en Bitbucket
```

```
mkdir -p ~/usuario1/git_universidade
```

```
cd ~/usuario1/git_universidade/
```

```
git init
```

```
touch index.html
```

```
echo "Simulación do arquivo index.html" >>  
index.html
```

```
touch style.css
```



Repositorios remotos (II)

```
echo "Simulación do arquivo style.css" >> style.css
```

```
git add .
```

```
git commit -m "Commit inicial. Engado o index.html e  
o style.css"
```

```
git tag -a v0.0.1 -m "Commit inicial"
```

```
git checkout -b dev
```

```
touch todo.txt
```

```
touch .gitignore
```

```
echo "todo.txt" >> .gitignore
```



Repositorios remotos (III)

```
git add .gitignore
```

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

```
git remote add origin
```

```
https://amieiro@bitbucket.org/amieiro/git_universidade.  
git
```

```
#Tamén git remote add origin  
git@bitbucket.org:amieiro/git_universidade.git).
```

```
git push -u origin master
```

```
git push -u origin dev
```

```
git push --tags
```



Repositorios remotos (IV)

```
# Usuario 2
```

```
mkdir ~/usuario2 && cd ~/usuario2
```

```
git clone
```

```
https://JesusAmieiro@bitbucket.org/amieiro/git_u  
niversidade.git
```

```
cd git_universidade/
```

```
git branch -a -v
```

```
git checkout -b dev origin/dev
```

```
mkdir img
```



Repositorios remotos (V)

```
cd img/
```

```
touch logo.png
```

```
touch logo.xcf
```

```
cd ..
```

```
echo "img/*.png" >> .gitignore
```

```
git status
```

```
git add .gitignore
```

```
git add img/logo.xcf
```



Repositorios remotos (VI)

git status

git commit -m "Engado o logo e modifco o
.gitignore"

git checkout master

git merge dev

git push --all



Repositorios remotos (VII)

```
# Usuario 1
```

```
git checkout master
```

```
git pull --all
```

```
git branch -a -v
```

```
git checkout dev
```

```
git pull
```

```
git branch -a -v
```

```
mkdir js/
```



Repositorios remotos (VIII)

```
cd js/
```

```
touch jquery-2.1.0.min.js
```

```
cd ..
```

```
git add .
```

```
git commit -m "Añado el archivo jquery-2.1.0.min.js"
```

```
git checkout master
```

```
git merge dev
```

```
git push --all
```



Onde continuar?

Libro “Pro Git” <https://progit.org>

Libro “Version Control with Git” de O'Reilly

Tutorial <https://es.atlassian.com/git/tutorials/learn-git-with-bitbucket-cloud>

Git Cheatsheets



www.jesusamieiro.com

