



25/01/2014  
Vigo

# Jesús Amieiro

@JesusAmieiro



# Qué es?



factura\_v5.ods



20140125\_CV.docx



Backup\_20140125

# Problemas

# Locales

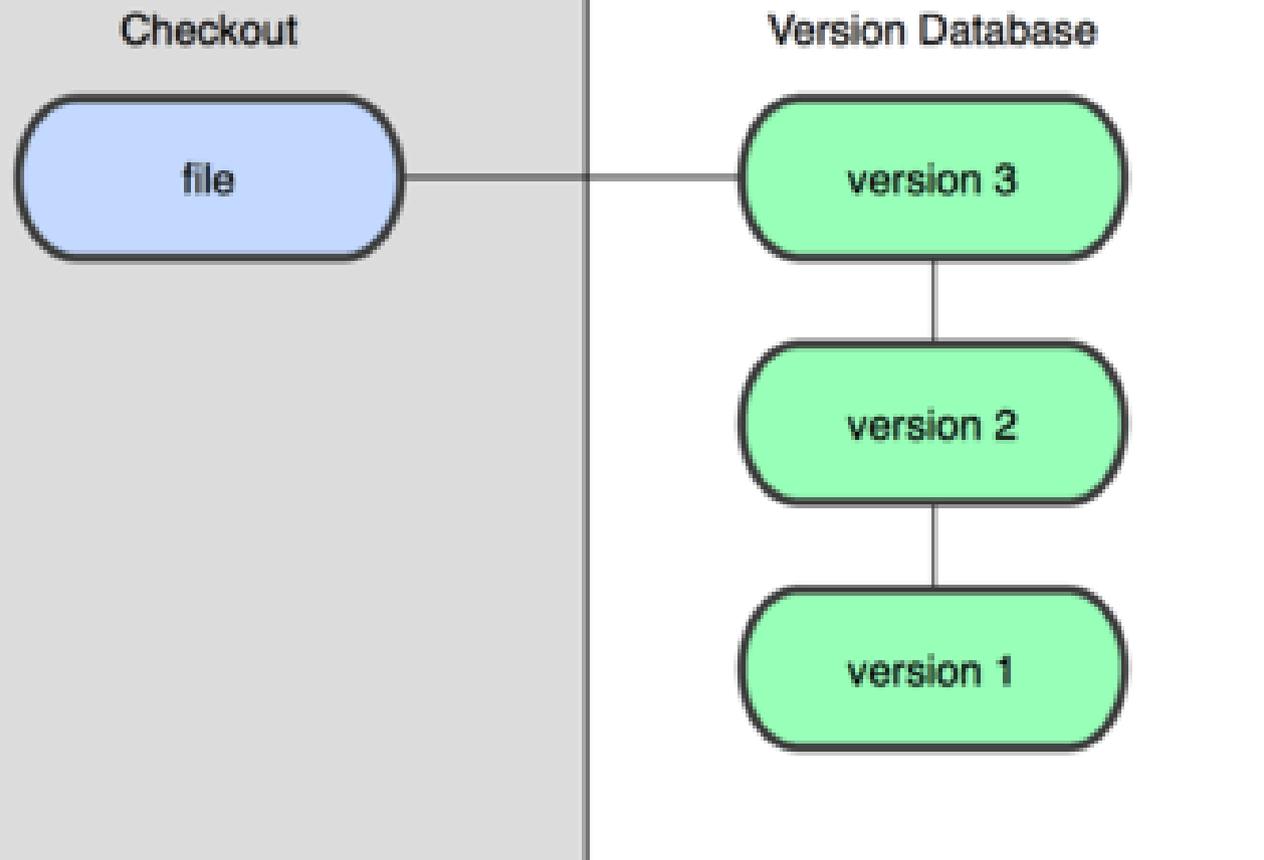
SCCS

1972

RCS

1982

## Local Computer



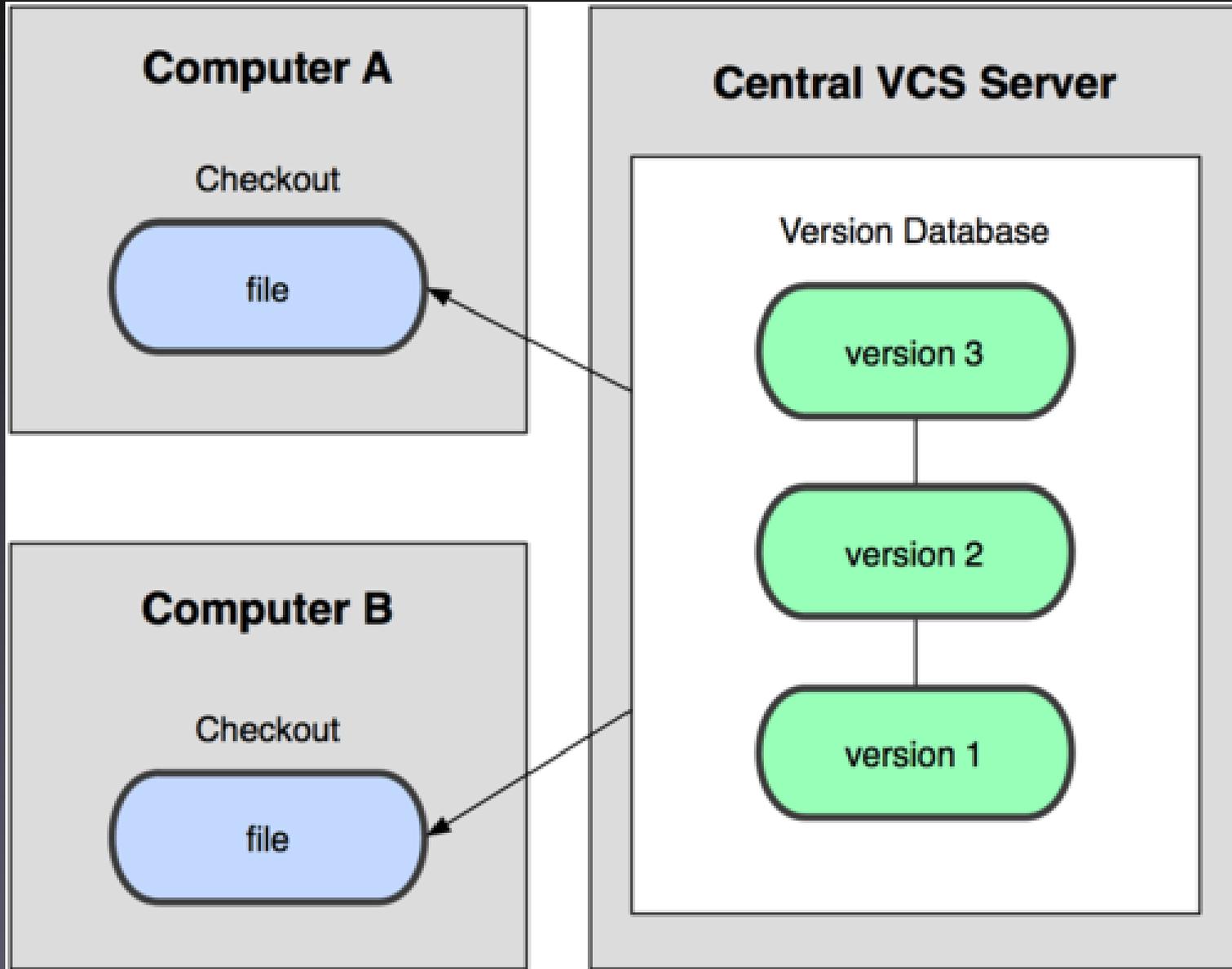
# Cliente servidor

CVS

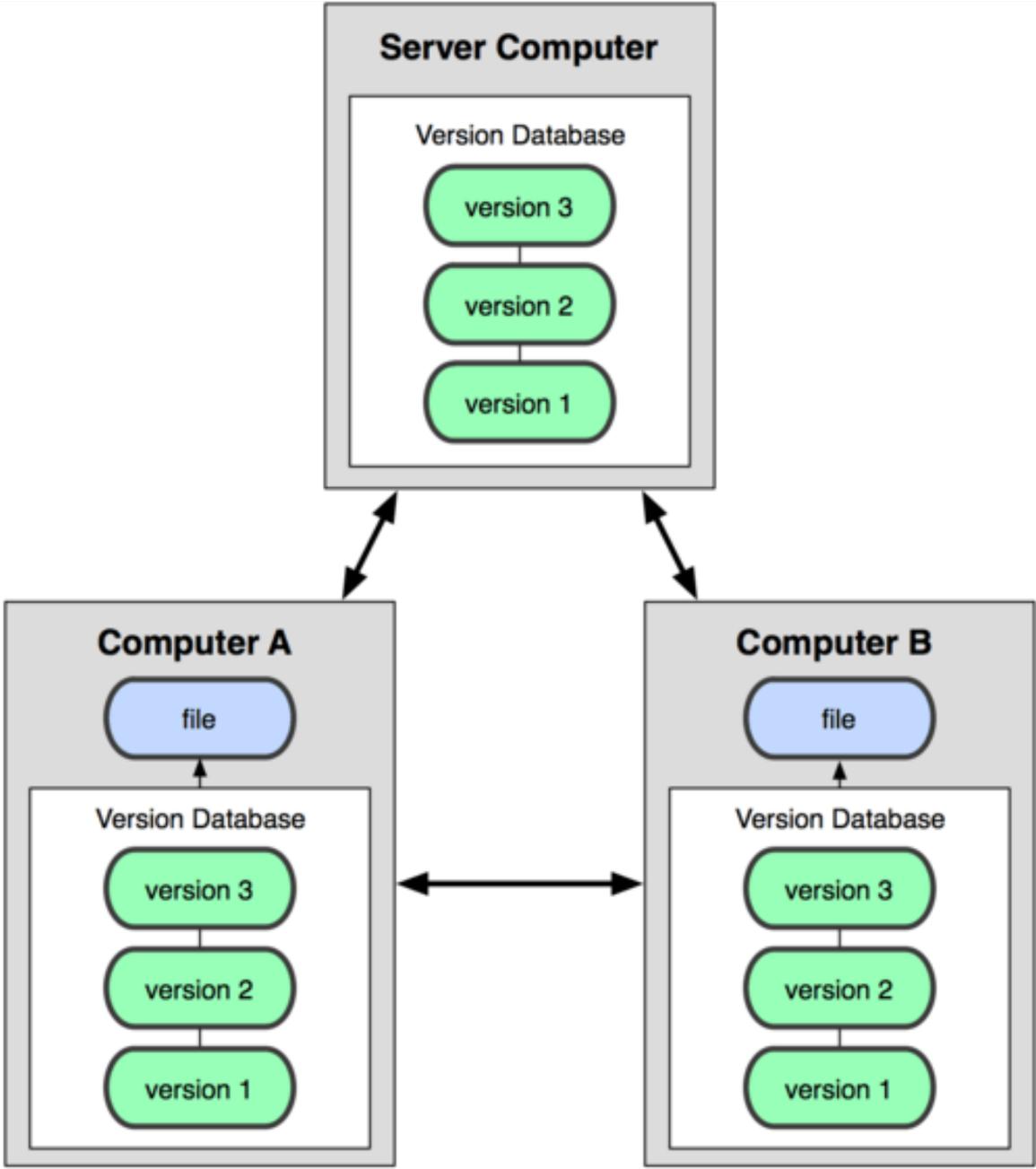
1990

Subversion

2000



# Distribuidos



# BitKeeper

2000

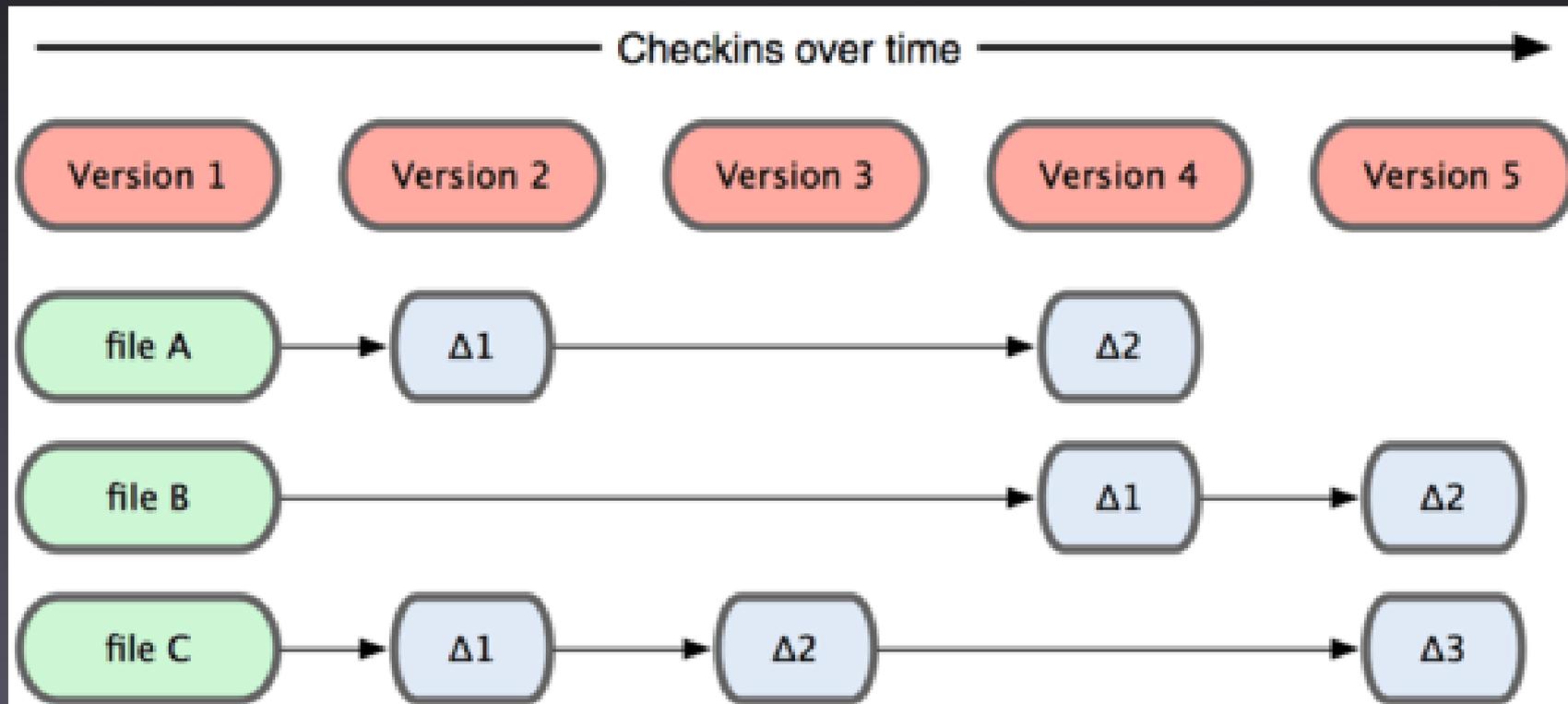




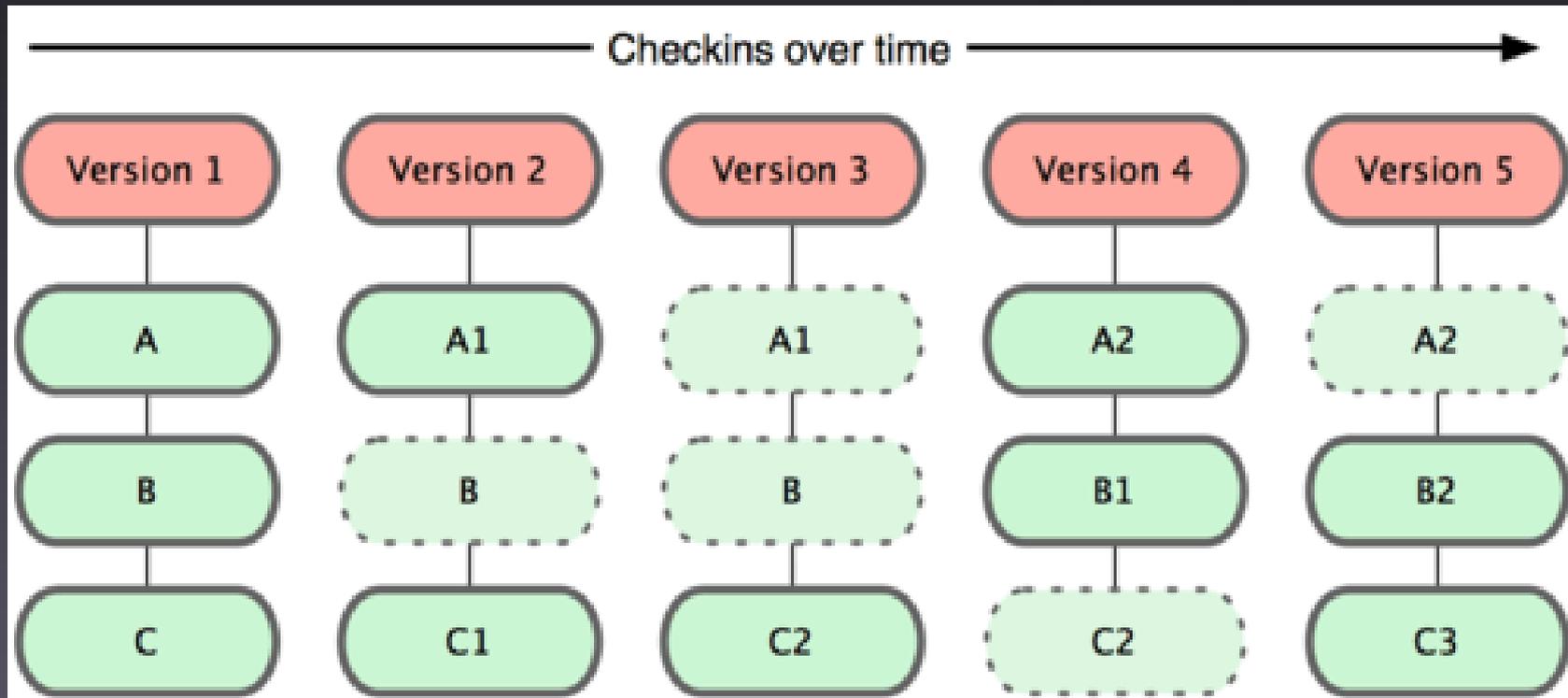
- Velocidad: proyectos grandes
- Desarrollo no lineal
- Distribuido
- Compatible con protocolos existentes

# Características

# Diferencias



# Instantáneas



Diseñado para  
desarrollo  
distribuido



# Escalable

kernel

# Rápido y eficiente

kernel

# Integridad

SHA-1 (40 hex.)

# Trazabilidad

Desarrollador



# Atomicidad

# Alienta las ramas

forks

# Operaciones locales

Repositorios completos



# Software libre

As freedom

Solo añade  
información

# Estados de un archivo

# 3 estados

Sin seguimiento (untracked)

Ignorado (ignored)

Bajo seguimiento (tracked)

# Estados de un archivo bajo seguimiento

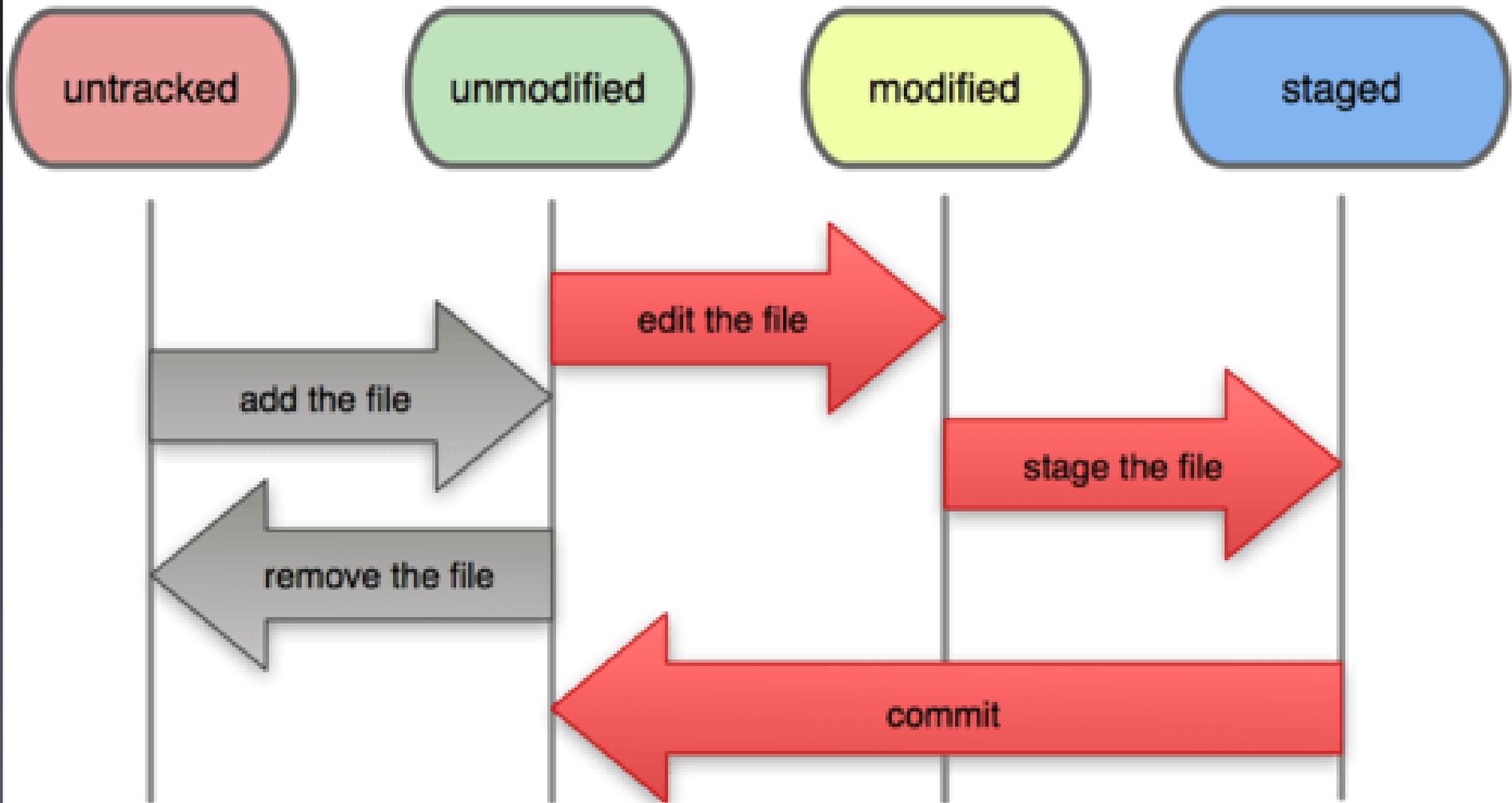
# 3 estados

Modificado (modified)

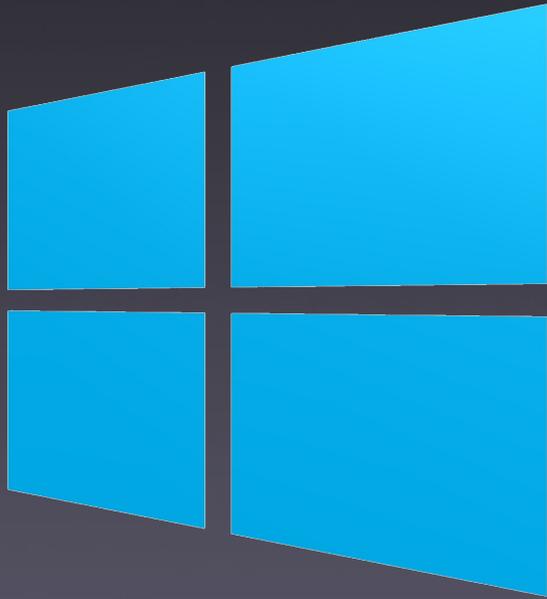
Preparado (staged)

Confirmado (committed)

# File Status Lifecycle



# Versiones



# Linux

`apt-get install git`

`yum install git`

Compilando el código

# Mac

## Git-osx-installer

## MacPorts



# Windows

msysgit

# GUI

git gui  
gitk

# Configuración

- Sistema

- `/etc/gitconfig` `git config --system`

- Usuario

- `~/.gitconfig` `git config --global`

- Repositorio

- `.git/config` `git config --local`



# Configuración

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

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

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

```
git config --list
```

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

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

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

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

```
git config --global core.autocrlf true (Windows)
```



# Ayuda

```
git help <comando>
```

```
git <comando> --help
```

```
man git-<comando>
```

```
git help config
```

# Ejercicio

# git clone

Inicialización

```
cd /my/dir/
```

```
git clone https://github.com/h5bp/html5-boilerplate.git
```



# git init

Inicialización

# Directorio .git

# Creo archivo

# git add

Añado primer\_archivo.txt

# git commit

## Commit inicial

- `git add .` o `git add nombre_archivo`
- `git commit -m "Commit inicial"`

# git log

man git-log

git log

git log -1

git log --since=2014-01-24

git log --until=2014-01-25

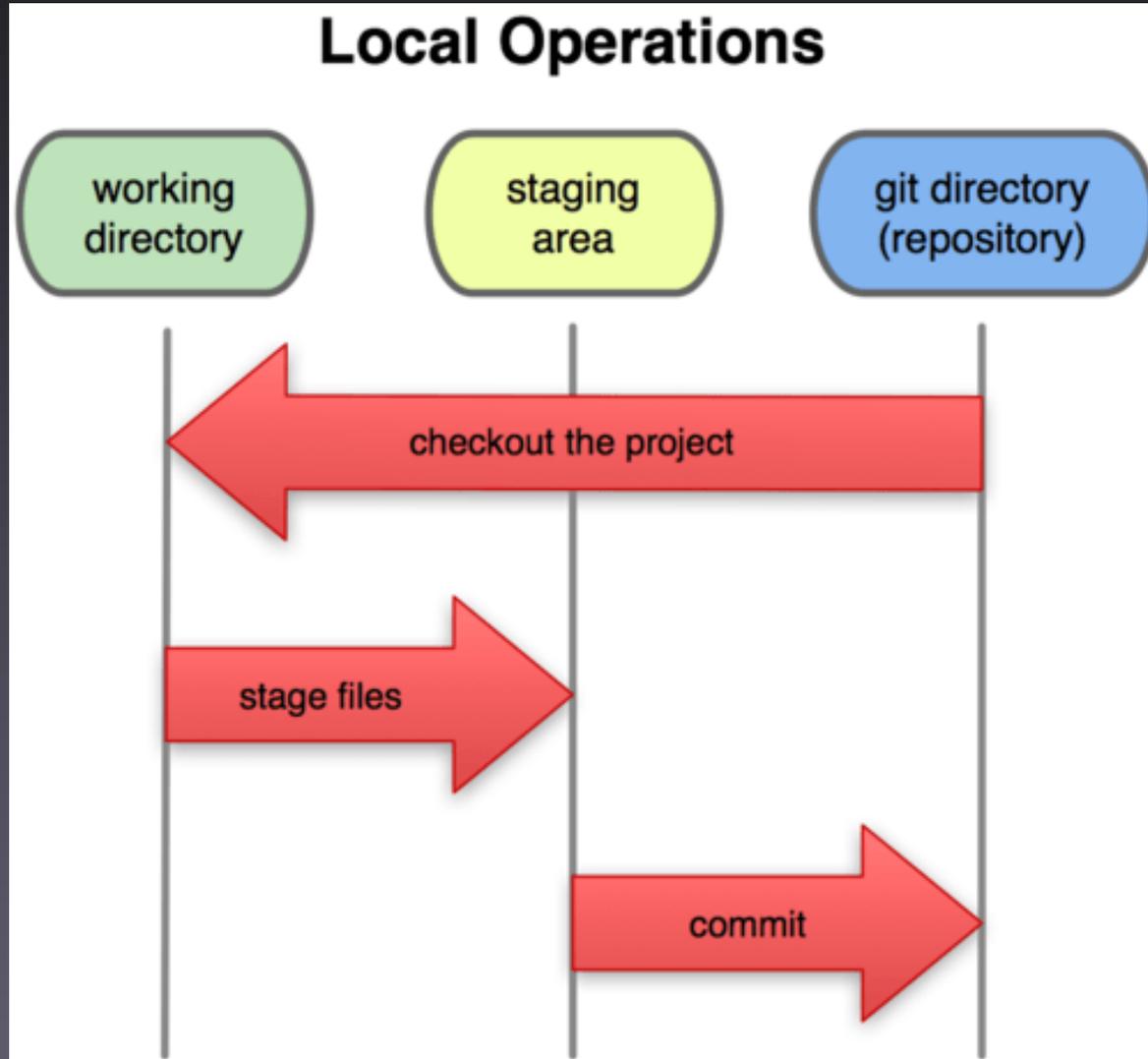
git log --author="Jesus"

git log --grep="archivo"

# SHA-1

# Arquitecturas 2 estados

# 3 estados



# Añado archivos

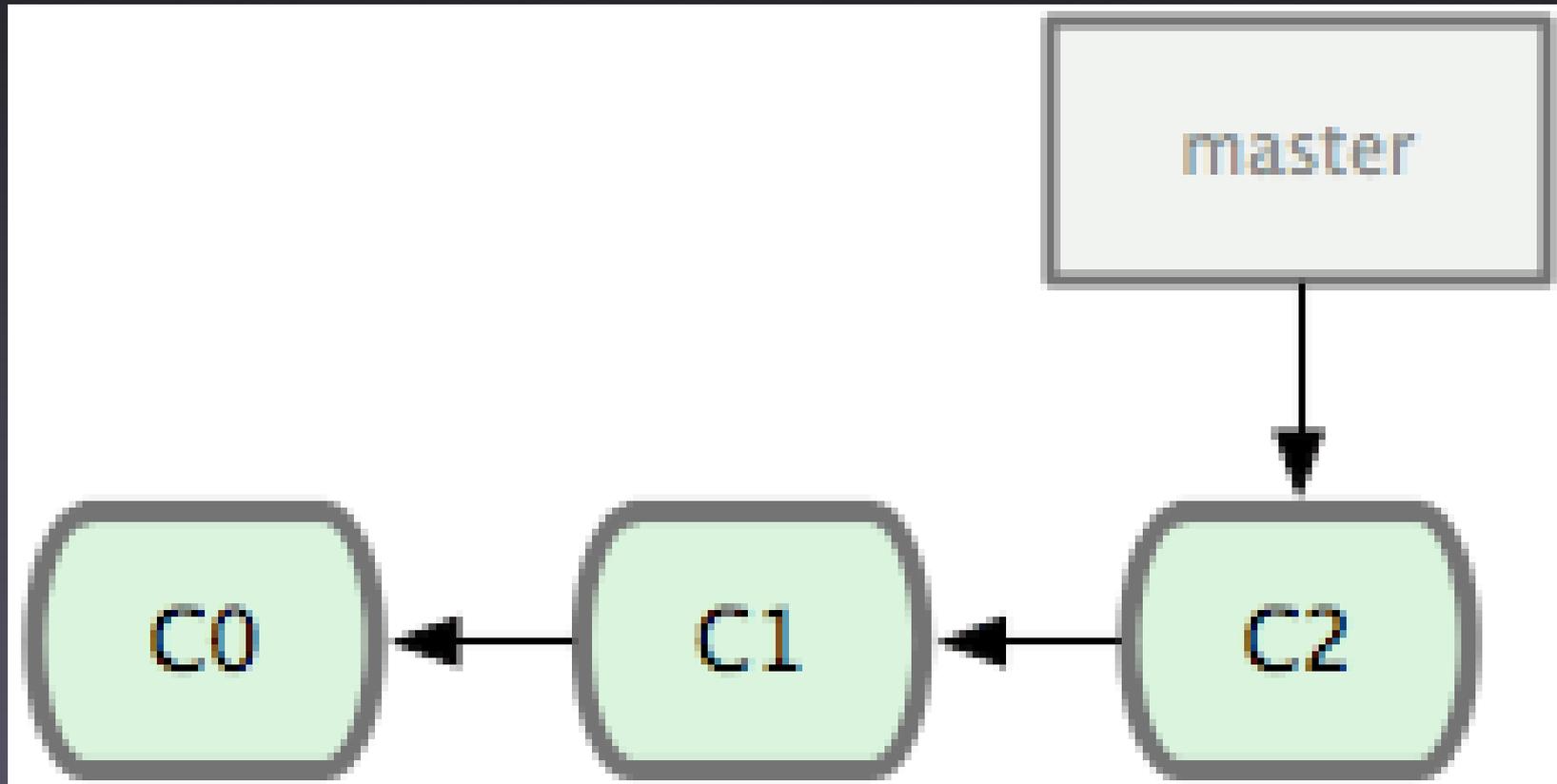
## Añado dos archivos

- segundo\_archivo.txt
- tercer\_archivo.txt

# Flujo de trabajo

# HEAD

# Commits



# git diff

Cambios

`git diff`

`git diff --staged // git diff --cached`

# git rm

Eliminar archivos

# git mv

Mover y renombrar archivos

# git checkout

Deshago cambios en la zona de trabajo

# git reset HEAD

Elimino los archivos del índice

# git commit --amend

Cambio el último commit

# git checkout

Recupero versiones antiguas de archivos

# git revert

Revirtiendo un commit

# git reset

Cambio el HEAD

# git clean

Borrar archivos no seguidos

# .gitignore

No rastrear determinados archivos

# Ramas

git branch

git checkout \$rama

git checkout -b \$rama

git branch -m nombre1 nombre2

git branch -d \$rama

# git log

Visualizando y filtrando los commits

# git show

Visualizando información de un commit u otro objeto

# git diff

Comparando commits

# git merge

Fusionando ramas

# git tag

Etiquetas

# Dónde continuar?

Libro "Pro Git" <http://git-scm.com/book/es/>

Libro "Version Control with Git" de O'Really

Git Cheatsheets



# Gracias

[www.jesusamieiro.com](http://www.jesusamieiro.com)

