

El Zen de git



© Angel Pablo Hinojosa Gutiérrez



El Zen de git

Un taller de introducción al uso de git del humilde sensei Psicobyte para la honorable Oficina de Software Libre de la UGR.



El problema

Al principio, sin sistemas de control de versiones, todo era caos, desorden, desconcierto y lágrimas.

El problema



borrame



borrame (de verdad)



copias antiguas



backup proyecto.zip



proyecto.zip



proyecto ANTERIOR.
zip



proyecto (copia).zip



proyecto (copia) (3ª
copia).zip



proyecto (copia)
(copia).zip



proyecto (copia)
(copia) (copia)
(copia).zip



proyecto
(otra copia).zip



proyecto (otra
copia).zip



proyecto otra copia.
zip



El Nirvana (¿Qué queremos?)

Guardar una copia de todos los estados anteriores, con todas las modificaciones hechas por todos los participantes, con comentarios y notas asociadas a cada cambio, y que permita gestionar conflictos entre versiones, crear diferentes ramas de proyecto y gestionarlás.

El Nirvana (¿Qué queremos?)

Y que sea sencillo

Koans de git (¿Qué es git?)

git es un Sistema de Control de Versiones

(aunque hay otros: CVS, Subversion, Bazaar...)

Koans de git (¿Qué es git?)

git es distribuido

(aunque se suele usar como si fuera centralizado)

Koans de git (¿Qué es git?)

git es de línea de comandos

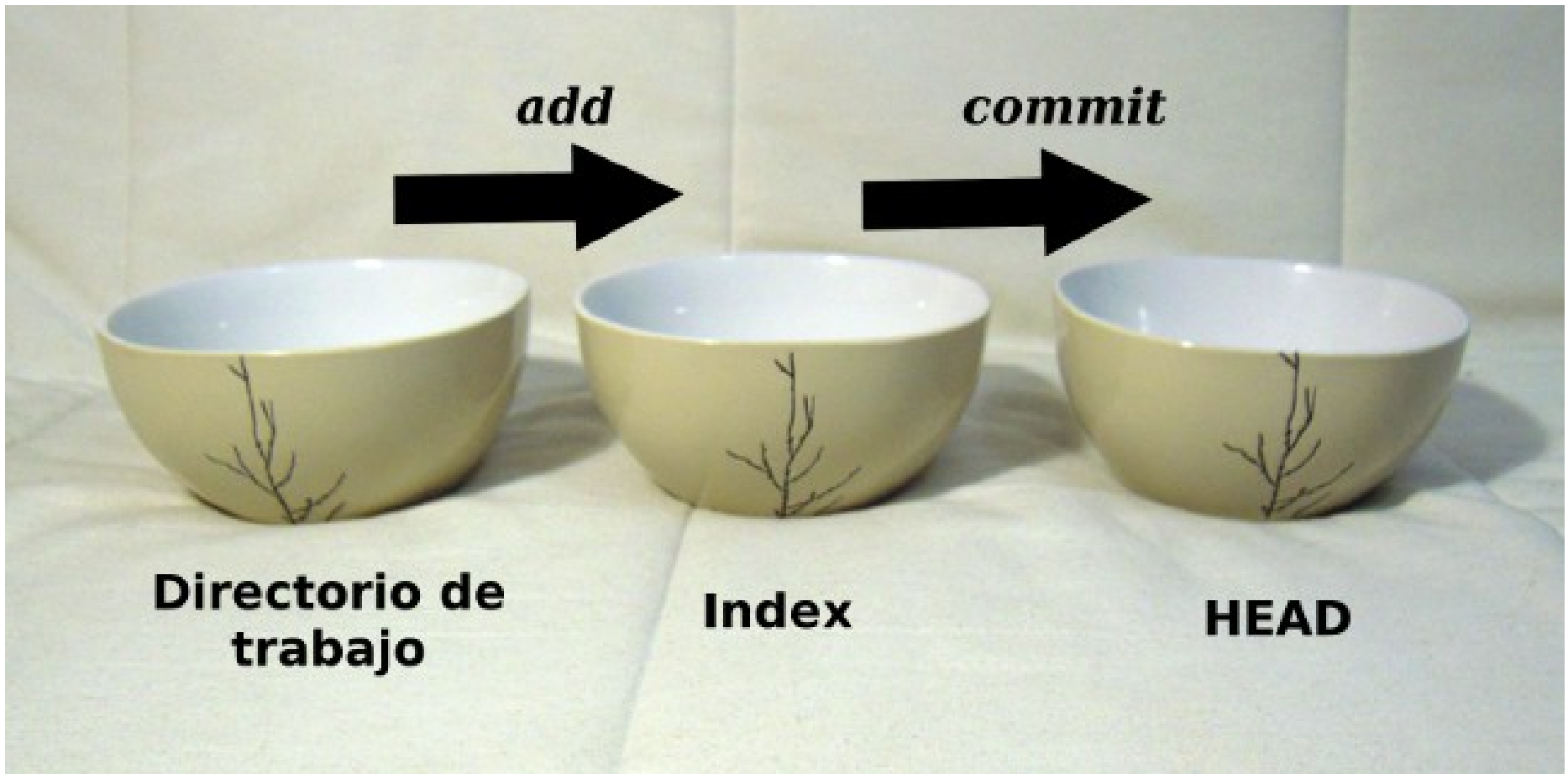
(aunque hay clientes gráficos: Tortoise, gitg, Gigggle...)

Koans de git (¿Qué es git?)

git se creó en y para Linux

(aunque se usa en y para todas las plataformas)

Los tres cuencos



Comenzar el Tao (Instalando git)

Debian: # aptitude install git

Archlinux: # pacman -S git

Windows: <http://windows.github.com> Clic aceptar...

Mac: <http://mac.github.com> Clic aceptar...

Android: Playstore -> Buscar "git" -> Instalar -> Aceptar permisos

La vía interior (Configuración)

```
git config --global user.name "Psicobyte Sensei"
```

```
git config --global user.email "psicobyte@gmail.com"
```

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

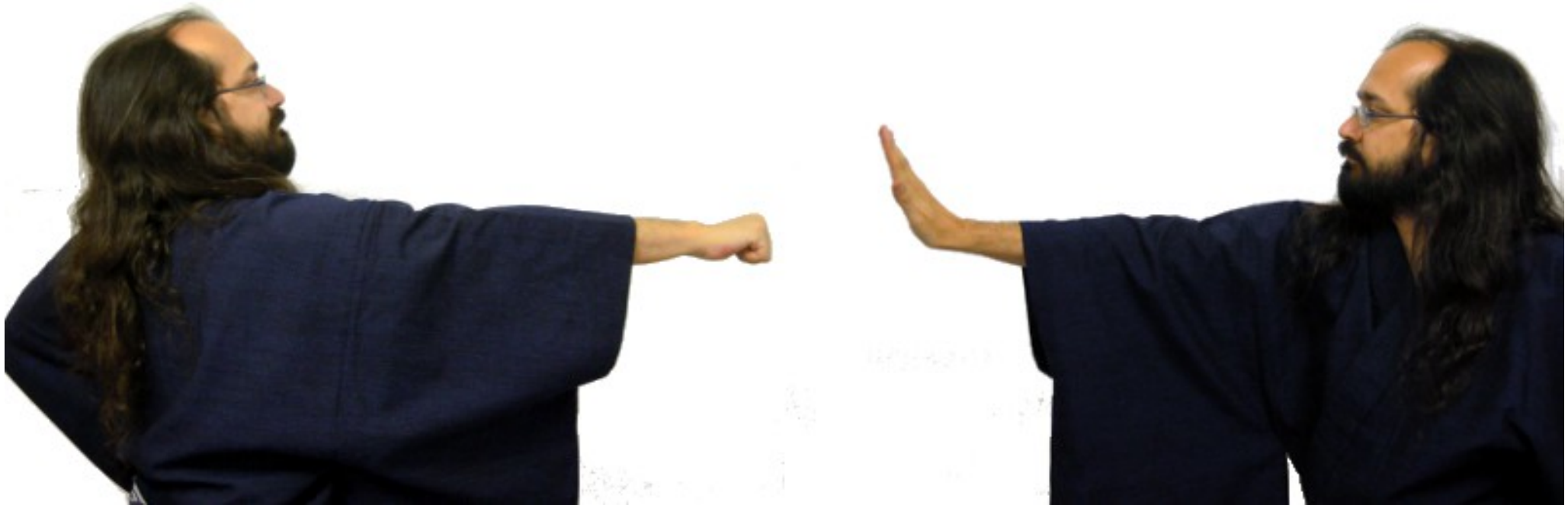
La vía exterior (Claves ssh)

ssh-keygen

ssh-copy-id Dirección-Del-Servidor

git como arte marcial

(O el “Dar cera – Pulir cera” de git)



git como arte marcial

Inicia un repositorio

```
git init
```


git como arte marcial

¡Ayuda!

git help COMANDO

git help init

git como arte marcial

Añade un origen remoto:

```
git remote add origin URL
```

git como arte marcial

O, directamente, clona un repositorio existente:

```
git clone URL
```

```
git clone git@github.com:psicobyte/PiMondrian.git
```

git como arte marcial

Añade ficheros al índice

```
git add Nombre-del-Fichero
```

```
git add Nombre-del-Directorio
```

```
git add .
```

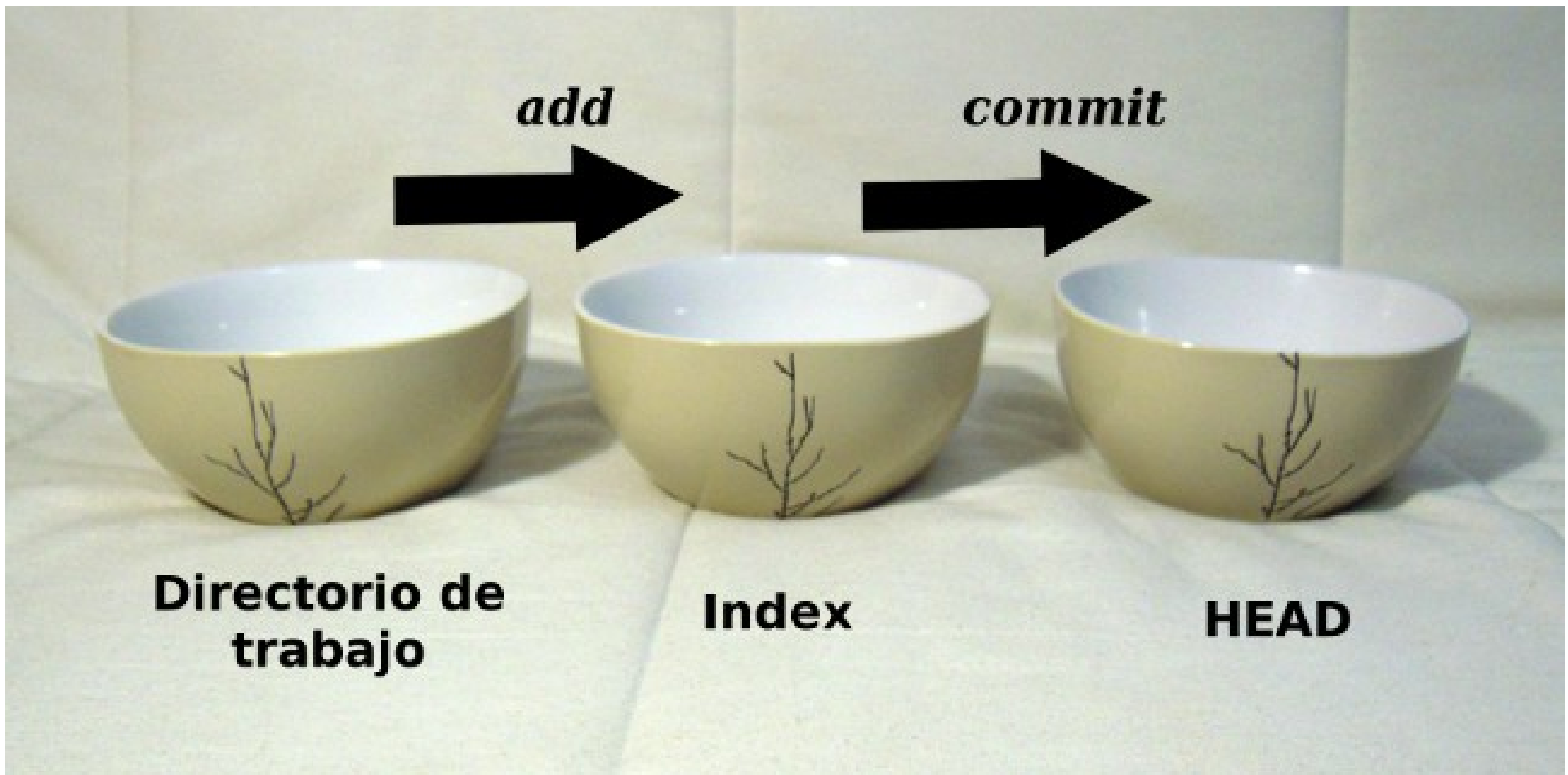
git como arte marcial

Pasa del índice al HEAD:

```
git commit
```

```
git commit -m "Comentario al commit"
```

¿Recuerdas los tres cuencos?



git como arte marcial

Borra:

```
git rm Nombre-del-Fichero
```

```
git rm --cached Nombre-del-Fichero
```

git como arte marcial

Mueve (o renombra):

```
git mv Nombre-del-Fichero Nuevo-Nombre
```


git como arte marcial

¿Ya he hecho commit? ¿En qué rama? ¿Me falta un add? ¿Qué está pasando?

git status

git como arte marcial

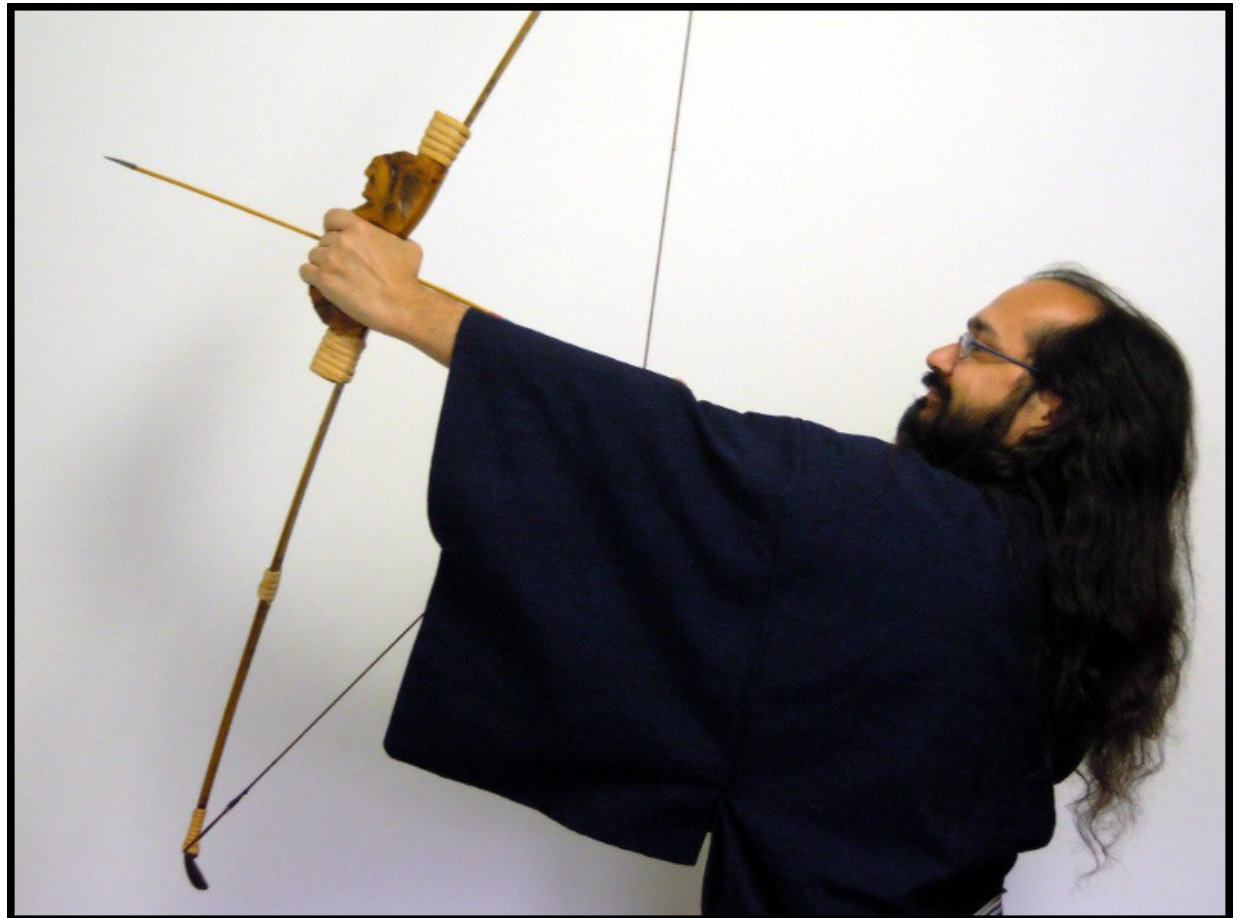
Conocer la historia:

`git log`

`git log --graph`

Más allá del horizonte

(sincronizando con repositorios remotos)



Más allá del horizonte

Envía cambios al servidor remoto

```
git push origin master
```

```
git push origin Nombre-De-La-Rama
```

```
git push
```

(recordamos)

Añade un origen remoto:

```
git remote add origin URL
```

O, directamente, clona un repositorio existente:

```
git clone URL
```

Más allá del horizonte

Recibe cambios desde el servidor remoto

```
git pull origin master
```

```
git pull origin Nombre-De-La-Rama
```

```
git pull
```

Más allá del horizonte

¡Conflicto entre servidores!

¿Qué hacemos?

git informa detalladamente del problema y debemos arreglarlo nosotros mismos y hacer un commit y un push.

La vía óctuple

(Usando ramas)



La vía óctuple

Crea una rama:

```
git branch Nombre-De-La-Rama
```

La vía óctuple

Nos movemos a una rama:

```
git checkout Nombre-De-La-Rama
```

Nos movemos a la rama principal:

```
git checkout master
```

La vía óctuple

Crea una rama y nos movemos a ella:

```
git checkout -b Nombre-De-La-Rama
```

La vía óctuple

¡Por el gran dragón! ¿En qué rama estoy?

git branch

La vía óctuple

Borra una rama:

```
git branch -d Nombre-De-La-Rama
```

La vía óctuple

Mezcla una rama con aquella en la que estés:

```
git merge Nombre-De-La-Otra-Rama
```

La vía óctuple

¡Conflicto entre ramas!

¿Qué hacemos?

git informa detalladamente del problema y debemos arreglarlo nosotros mismos y hacer un commit.

El templo shaolin de git



Enlaces y referencias

Git: <http://git-scm.com/>

GitHub: <http://github.com/>

TryGit: <http://try.github.io/>

OSL de la UGR: <http://osl.ugr.es/>

Muchas gracias

<http://www.psicobyte.com>

psicobyte@gmail.com

[@psicobyte_](#)

Esta presentación se puede descargar de:

<http://www.psicobyte.com/info/descargas>

