

De la mano con git

Un taller básico

Agenda

Trabajar sin un VCS

¿Que es git?

¿Pros?

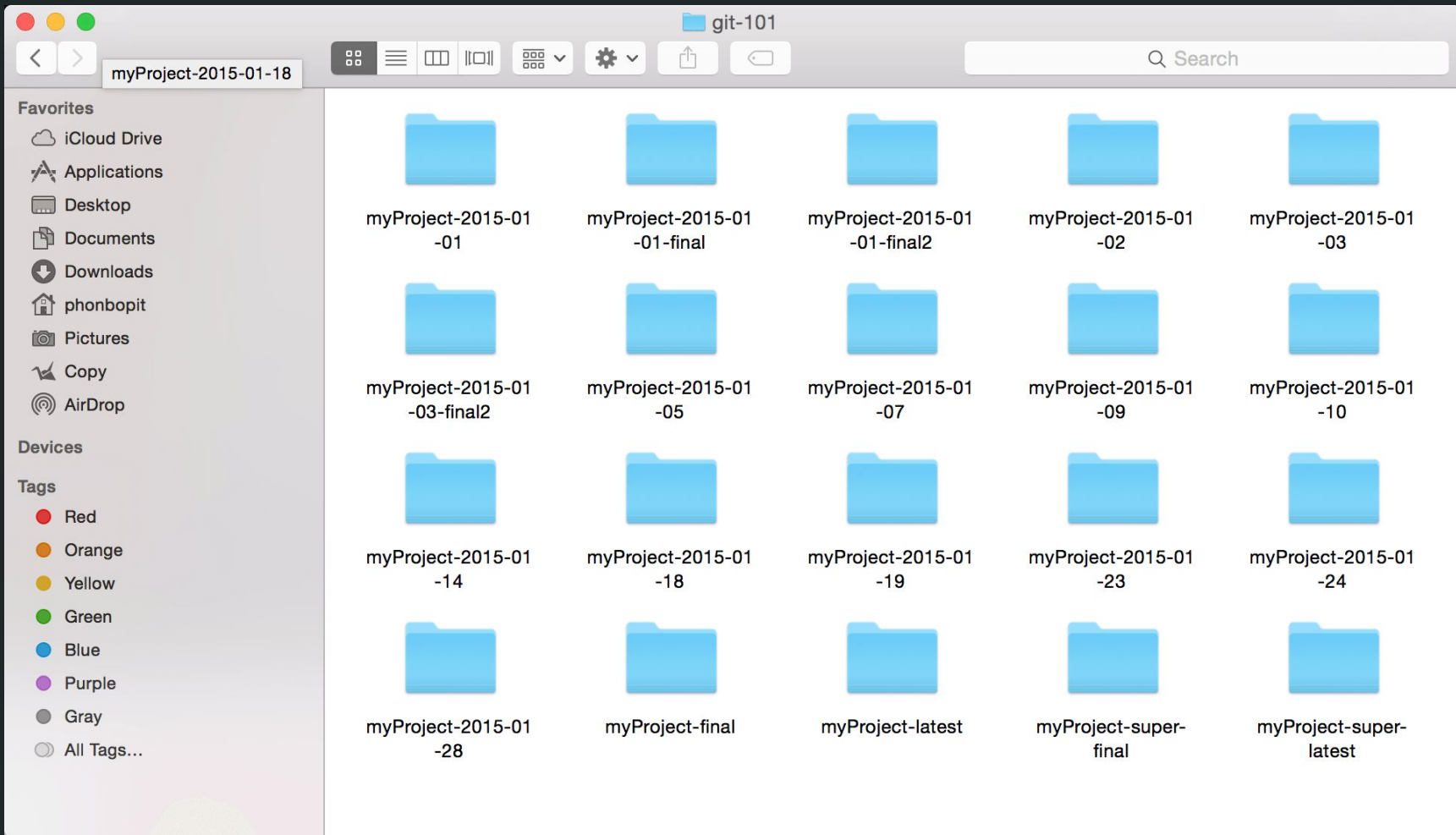
Instalación

Configuración

Comandos básicos

Github

Herramientas



Trabajar **sin** un VCS

- El control sobre los archivos de un proyecto es nulo.
- Modificar sin la posibilidad de retroceder a los cambios anteriores.
- Eliminar archivos sin la posibilidad de volverlos a recuperar.
- Impone presión innecesaria en un equipo de trabajo.

¿Que es **git**?

Sistema de control de versiones distribuido (**D**VCS), que como los otros VCS está orientado a la gestión de los diversos cambios de un elemento de un producto o configuración del mismo.

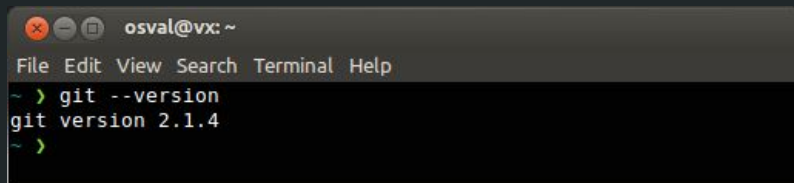
Ventajas

- Las modificaciones en un proyecto son controlables.
- Se puede tener un histórico del trabajo realizado y posibilidad de revertir dichos cambios.
- La comunicación con el equipo de trabajo es clara.
- Se conoce el estado actual del proyecto.
- Recibir feedback sobre cambios específicos

Instalación

En ambientes GNU/Linux:

- Derivados de Debian
\$ sudo apt-get install git
- Fedora
\$ sudo yum install git

A terminal window titled 'osval@vx: ~' with a menu bar (File, Edit, View, Search, Terminal, Help). The terminal shows the command 'git --version' being executed, resulting in the output 'git version 2.1.4'. The prompt '~ >' is visible before and after the command.

```
osval@vx: ~  
File Edit View Search Terminal Help  
~ > git --version  
git version 2.1.4  
~ >
```

En Windows:

- Ir a <http://git-scm.com/download/win>, bajar e instalar

Configuración

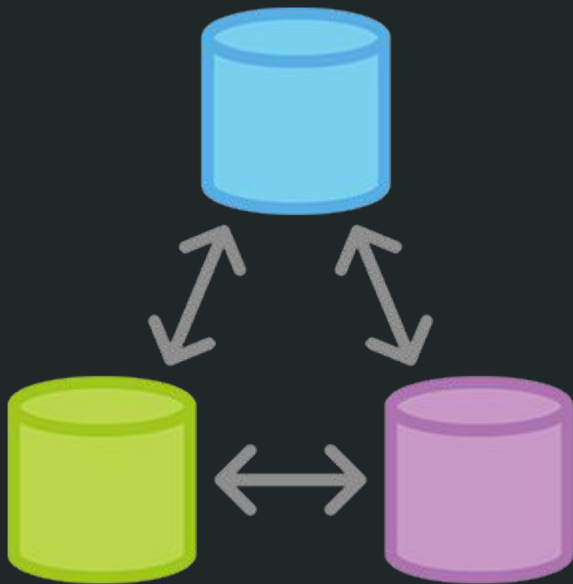
Establecer el nombre de nuestro usuario.

```
$ git config --global user.name "Osval Reyes"
```

Establecer el correo electrónico.

```
$ git config --global user.email osvaljr2@gmail.com
```

Comandos básicos



```
$ git init
```

Inicializa un repositorio, con este comando le decimos a Git que haga seguimiento de todo lo que hagamos en la carpeta actual.

```
$ git clone <url_repo>
```

Clona repositorios

```
$ git remote add origin <url_repo>
```

Este comando solo se usa solo cuando queremos añadir un nuevo repositorio, no se utiliza con frecuencia.

Comandos básicos

```
$ git status
```

Muestra el estado actual del repositorio local

```
$ git log
```

Muestra commits que se han realizado, tanto locales, como los del repositorio remoto.

```
$ git show a906cb2
```

Muestra los cambios asociados a un commit.

a906cb2a4a904a152e80877d4088654daad0c859

SHA-1

Comandos básicos

```
$ git diff
```

Muestra diferencias en archivos

```
$ git add
```

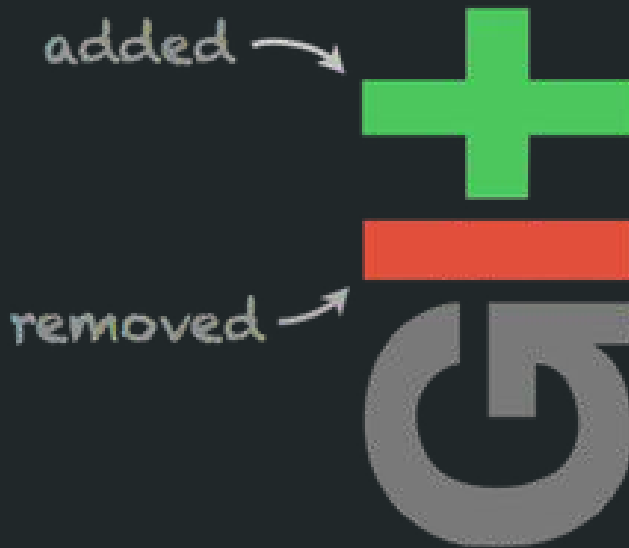
Añade archivos modificados. Recibe un parámetro que es el path del archivo a añadir.

```
$ git commit --message=""
```

Con éste comando se sella el sobre con el nombre del usuario, correo y mensaje principal de los cambios hechos.

```
$ git push -u origin  
nombre_branch
```

Sube a un repositorio remoto el branch



Comandos básicos

```
$ git diff
```

Muestra diferencias en archivos

```
$ git add
```

Añade archivos modificados. Recibe un parámetro que es el path del archivo a añadir.

```
$ git commit --message=""
```

Con éste comando se sella el sobre con el nombre del usuario, correo y mensaje principal de los cambios hechos.

```
$ git push -u origin  
nombre_branch
```

Sube a un repositorio remoto el branch



Comandos básicos



`$ git reset`

Revierte el efecto del comando `git add`.

`$ git checkout`

Sirve para revertir modificaciones de los archivos especificados, cambiar de rama de desarrollo, crear ramas de desarrollo.

`$ git merge`

Este comando se usa para mezclar los cambios que se han hecho de una rama a otra.

`$ git rm`

Elimina archivos del repositorio.

Github

Plataforma de desarrollo colaborativa de software para alojar proyectos basado en git, brinda herramientas que permite el trabajo en equipo y la gestión de repositorios. Es gratuito para proyectos abiertos.

Contiene muchas funcionalidades, entre ellas:

- **Fork**: clonar un repositorio de una cuenta a otra.
- **Pull Request**: proponer cambios ramas de desarrollo basado en una rama de desarrollo.
- **Comentar** en los cambios realizados.

Herramientas

- `git log --oneline --graph --decorate`
- SmartGit
- gitk

...mas en <https://git-scm.com/downloads/guis>