



Control de versiones Manejo Básico

Esteban Saavedra Lopez

CEO AtixLibre

CEO Opentelematics

email: esteban.saavedra@atixlibre.org

estebansaavedra@gmail.com

<http://esteban.opentelematics.org>



Acerca del facilitador?



- Ingeniero de profesión
- Docente Universitario de Postgrado
- Investigador senior en innovación tecnológica.
- Presidente de la Fundación Atixlibre
- CEO Opentelematics Internacional
- Asesor de Tecnologías de la Información de Gobierno

Objetivo

- Realizar una presentación del manejo básico de un Sistema de Control de Versiones, considerando sus principios, sus características y sus tendencias

Contenido

- Git
- Instalación
- Configuración inicial
- Manejo básico

Git

- **Git** es un sistema de control de versiones distribuido, que permite la creación de una historia (bitácora) para una colección de archivos (código fuente de programas, documentación, etc)



Por que Git (1)

- **Adaptable**
 - > Orientado a la mayoría de los trabajos de desarrollo de software (local y/o remoto).
- **Amigable**
 - > Enfocado principalmente a la usabilidad y eficiencia.
- **Seguro**
 - > Mantiene características que proporcionan seguridad en todo el proceso de control de versiones.

Por que Git (2)

- **Rápido**
 - > Permite obtener rápidamente sus ventajas y características sin afectar al proceso de desarrollo.
- **Eficiente**
 - > Su almacenaje utiliza un formato altamente eficiente y limpio.
- **Ligero**
 - > No precisa tener un servidor dedicado

Por que Git (3)

- **Extensible**

- > Posee una arquitectura basada en plugins, aspecto que permite dotarle de mayores características específicas y acorde a cada necesidad.

- **Embebido**

- > Característica que le permite estar presente en una serie de aplicaciones y servicios libres y/o comerciales.

- **Libre**

- > Es software libre.

Instalación

- Red Hat - CentOS

- > `yum install git`

- Debian - Ubuntu

- > `apt-get install git`

Configuración inicial (1)

- Git trae una herramienta llamada `git config` que te permite obtener y establecer variables de configuración que controlan el aspecto y funcionamiento de Git

Configuración inicial (2)

- Estas variables pueden almacenarse en tres sitios distintos:
 - > Archivo `/etc/gitconfig`
 - Contiene valores para todos los usuarios del sistema y todos sus repositorios. Si pasas la opción `--system` a `git config`, lee y escribe específicamente en este archivo.
 - > Archivo `~/.gitconfig` o `~/.config/git/config`
 - Este archivo es específico a tu usuario. Puedes hacer que Git lea y escriba específicamente en este archivo pasando la opción `--global`.
 - > Archivo `config` en el directorio de Git (`.git/config`) del repositorio que estés utilizando actualmente:
 - Este archivo es específico al repositorio actual.

Identidad del usuario

- Asignación del nombre

- > `git config --global user.name "Esteban Saavedra"`

- Asignación del email

- > `git config --global user.email
esteban.saavedra@atixlibre.org`

Resaltado de colores

- `git config --global color.status auto`
- `git config --global color.branch auto`
- `git config --global color.interactive auto`
- `git config --global color.diff auto`

Listar configuración

- `git config --list`

Estructura de trabajo en el SCV

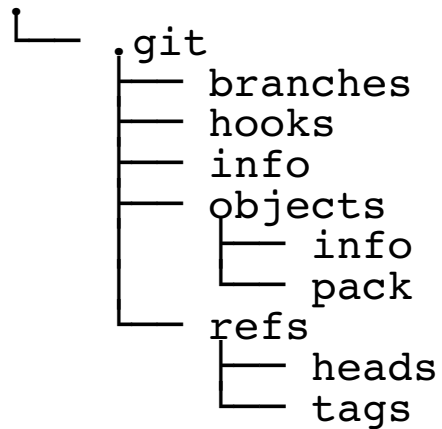
working directory

staging area

repository

inicializar el proyecto

- `git init`



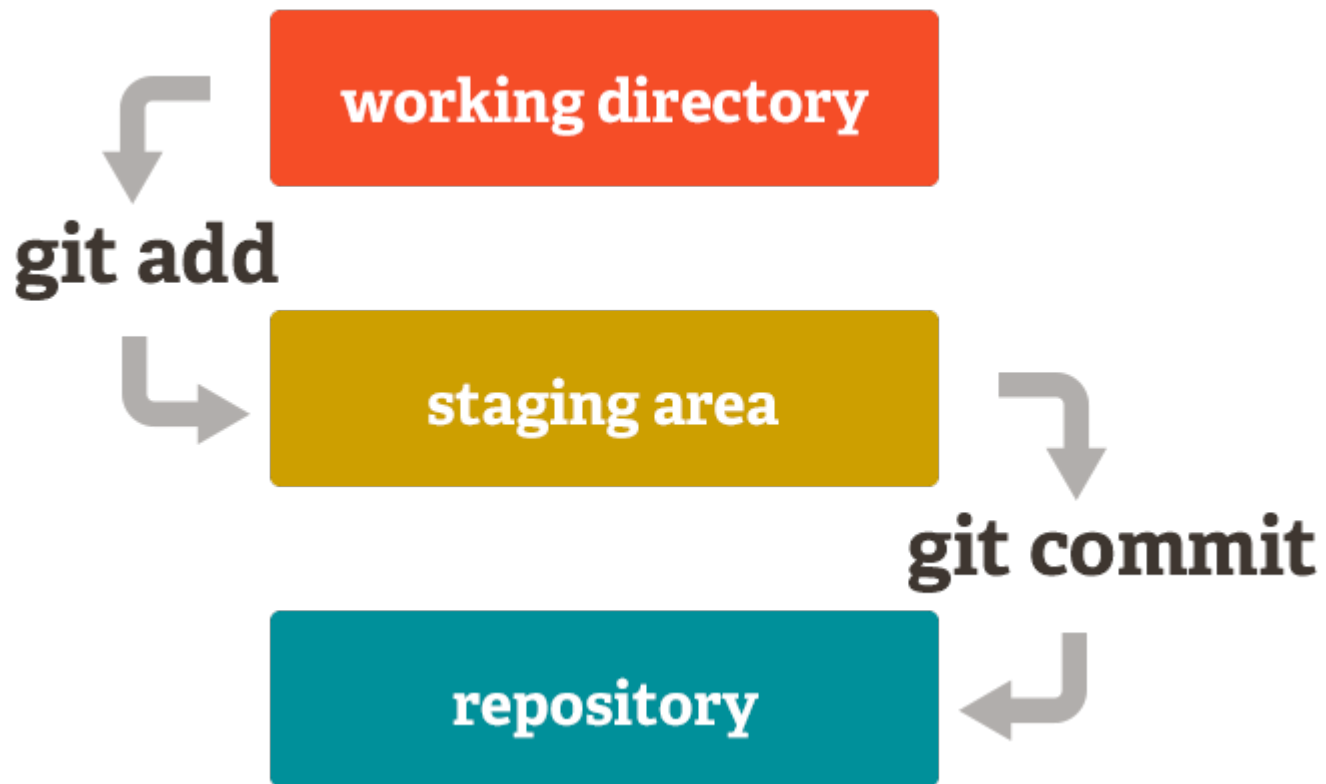
Ver el estado

- `git status`

Añadir archivos al staging

- Añadir un solo archivo
 - > `git add archivo`
- Añadir todos los archivos
 - > `git add -A`
 - > `git add .`

Movimiento y registro de cambios



Guardar en el repositorio local

- Por defecto

- > `git commit`

- Con mensaje incluido

- > `git commit -m "Mensaje del cambio realizado"`

Ver el historial de cambios

- Historial de los commits realizados

```
> git log
```

- Historial resumido

```
> git log --oneline
```

- Historial y ubicación del HEAD

```
> git log --oneline --decorate
```

Moverse a través del historial

- Movernos entre las distintas modificaciones
 - > `git checkout eb4a479d831669f484339a74b65bd9a49be7b080`
- Movernos al ultimo commit del repositorio
 - > `git checkout master`

Eliminar commits

- Eliminar commits

- > `git reset --soft`
`c912989a31432467cc99588e90063d6ded7a30c6`

- Eliminar commits del repositorio y del staging

- > `git reset --hard`
`c912989a31432467cc99588e90063d6ded7a30c6`

Modificar commits

- Modificar el ultimo commit

> `git commit --amend`

Deshacer commits

- Deshacer una modificación desde el staging
 - > `git reset HEAD archivo`
 - > `git checkout -- archivo`

Revertir commits

- Revertir el ultimo commit
 - > `git revert HEAD`
- Revertir sin hacer commit (lo lleva al staging)
 - > `git revert --no-commit HEAD`
- Revertir varios commit a la vez
 - > `git revert --no-commit HEAD`
 - > `git revert --no-commit HEAD~1`
 - > `git revert --continue`

Diferencias entre commits

- Ver diferencias entre commits
 - > `git diff a6524wc54 b763253d33`
- Ver diferencias haciendo uso de HEAD
 - > `git diff HEAD~1 HEAD`
 - > `git diff HEAD~2`

Preguntas - Consultas





Control de versiones Manejo Básico

Esteban Saavedra Lopez

CEO AtixLibre

CEO Opentelematics

email: esteban.saavedra@atixlibre.org

estebansaavedra@gmail.com

<http://esteban.opentelematics.org>

