

# Git 101

Víctor Orozco

GuateJUG

October 2, 2015

# Víctor Orozco

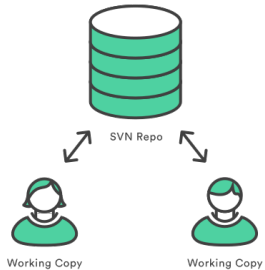
- ▶ Developer (JVM/Open Source Advocate)
- ▶ Ex-JUG Leader
- ▶ Consultor independiente (Nabenik)
- ▶ @tuxtor
- ▶ The J\*



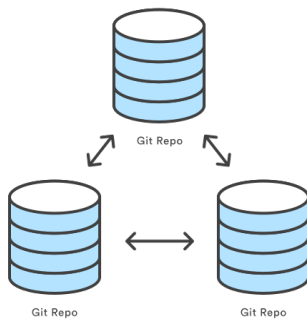
# Git (historia)

- ▶ DVCS
- ▶ Linus Torvalds (2005)
- ▶ Bitkeeper workflow
- ▶ "The stupid content tracker"

Central-Repo-to-Working-Copy Collaboration



## Repo-To-Repo Collaboration



# Git (características)

- ▶ Soporte robusto a metodologías no lineales
- ▶ Compatibilidad con protocolos estándar (HTTPS, SSH)
- ▶ Eficiencia con grandes volúmenes de datos
- ▶ Estándar criptográfico de nombrado
- ▶ Modular, toolkit, GPLv2

# Git (características)

- ▶ Soporte robusto a metodologías no lineales
- ▶ Compatibilidad con protocolos estándar (HTTPS, SSH)
- ▶ Eficiencia con grandes volúmenes de datos
- ▶ Estándar criptográfico de nombrado
- ▶ Modular, toolkit, GPLv2

## Setup



# Git workflow

Creamos un repositorio

```
git init
```

# Git workflow

Copiamos un repositorio local

```
git checkout /path/to/repository
```

Copiamos un repositorio remoto

```
git checkout username@host:/path/to/repository
```

# Git workflow

Consiste en tres arboles

- ▶ Working directory
- ▶ Staging
- ▶ HEAD



# Git workflow

Ejemplo add & commit

```
git add <filename>
```

```
git commit -m "Creando mi primer archivo git"
```

# Git config

```
git config --global user.name <name>  
git config --global user.email <email>
```

# Git workflow

Enviar a servidor remoto

- ▶ `git push <remote> <branch>`
- ▶ `git remote add <remote> <url>`

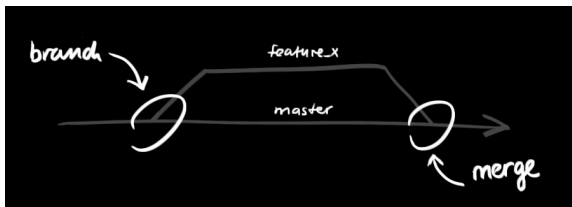
```
git remote add origin <server>  
git push origin master
```

# Git branching

- ▶ Isolar características entre si
- ▶ **master** es la rama predefinida

```
git checkout -b <branch>
```

```
git push origin <branch>
```



# Git merge

- ▶ Integrar características desde una rama
- ▶ Rama actual = rama donde se integrara el cambio

```
git checkout master  
git merge <branch>  
git branch -d <branch>
```



# Git update

```
git pull //obtiene cambios remotos  
git merge <branch>
```

# Git log

```
git log
git log --pretty=oneline
git log --graph --oneline --decorate --all
git log --name-status //archivos cambiados
```

# Git tagging

```
git tag 1.0.0 1b2e1d63ff
```

# Git replace

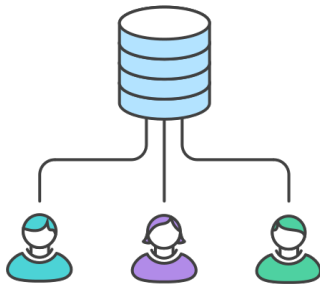
```
git checkout -- <filename>
```

Obtiene el ultimo HEAD, cambios en index y new file son conservados

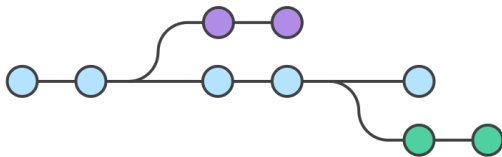
# Workflows

- ▶ Centralized
- ▶ Feature branch
- ▶ **Gitflow**
- ▶ **Forking**

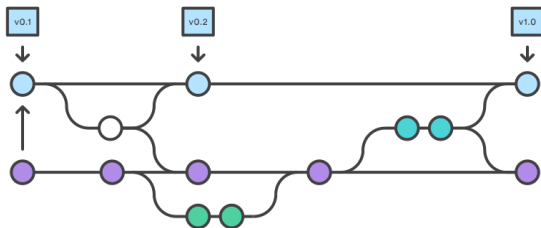
# Centralized



# Feature branch

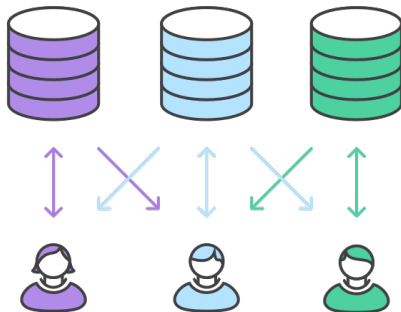


# Gitflow





# Forking



- ▶ Atlassian Git Tutorials
- ▶ Git School

- ▶ me@vorozco.com
- ▶ <http://vorozco.com>
- ▶ <http://github.com/tuxtor/slides>



This work is licensed under a Creative Commons Attribution-ShareAlike 3.0 Guatemala License.