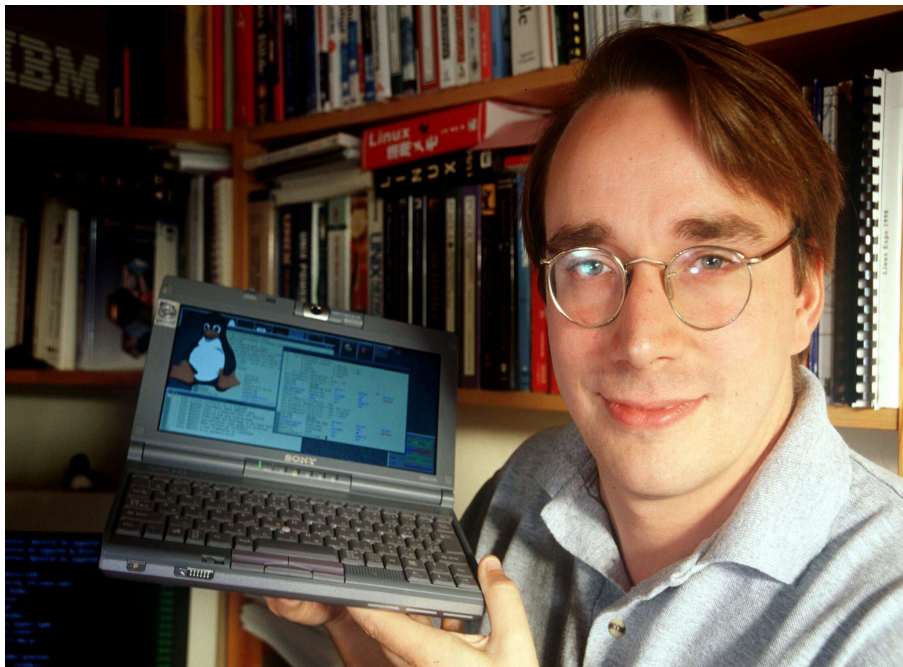


Curso básico de Git

by: cimadev community

Un poquito de historia

- Git fue creado por Linus Torvalds.
- Creado como solución, para mantener linux después de separarse de la empresa de bitkeeper en el 2005.



Un poquito de historia

— — —

Enfoque en:

- Velocidad
 - Trabajo fluido
- Diseño sencillo
 - Código robusto con herramientas necesarias.
- Fuerte apoyo en el desarrollo no lineal
- Complementamente distribuido
 - Copias personales de proyectos
- Capaz de manejar grandes proyectos.
 - Linux, Django, Alibaba, Laravel.

Que es Git?

- Git es un sistema de control de versiones (git, mercurial)
- Un sistema que registra los cambios realizados sobre un archivo o conjunto de archivos a lo largo del tiempo.

Tipos de sistemas de control de versiones.

— — —

1. Local

- a. Vive en nuestro ordenador. Si le ocurre una catástrofe vamos a perder nuestro trabajo. (Ctrl-z)

2. Centralizado

- a. Depende de un servidor donde está almacenado el repositorio. Si se quema el servidor vamos a llorar.

3. Distribuido

- a. El caso de git. Cada participante del repositorio tiene una copia local y no afecta el trabajo del resto. NO perdemos trabajo si nuestro ordenador se danama, simplemente pedimos una copia a cualquier servidor donde esté almacenado.

Estados de Git

— — —

- Working directory
 - Donde trabajamos de manera local, pero para guardarlo hay que pasarlo al Staging Area
- Staging Area
 - Es el área de preparación, se almacenan justo antes de hacer commit
- Git repository
 - El repositorio donde almacenaremos los cambios del proyecto.

Diferencia entre Git y GitHub

— — —

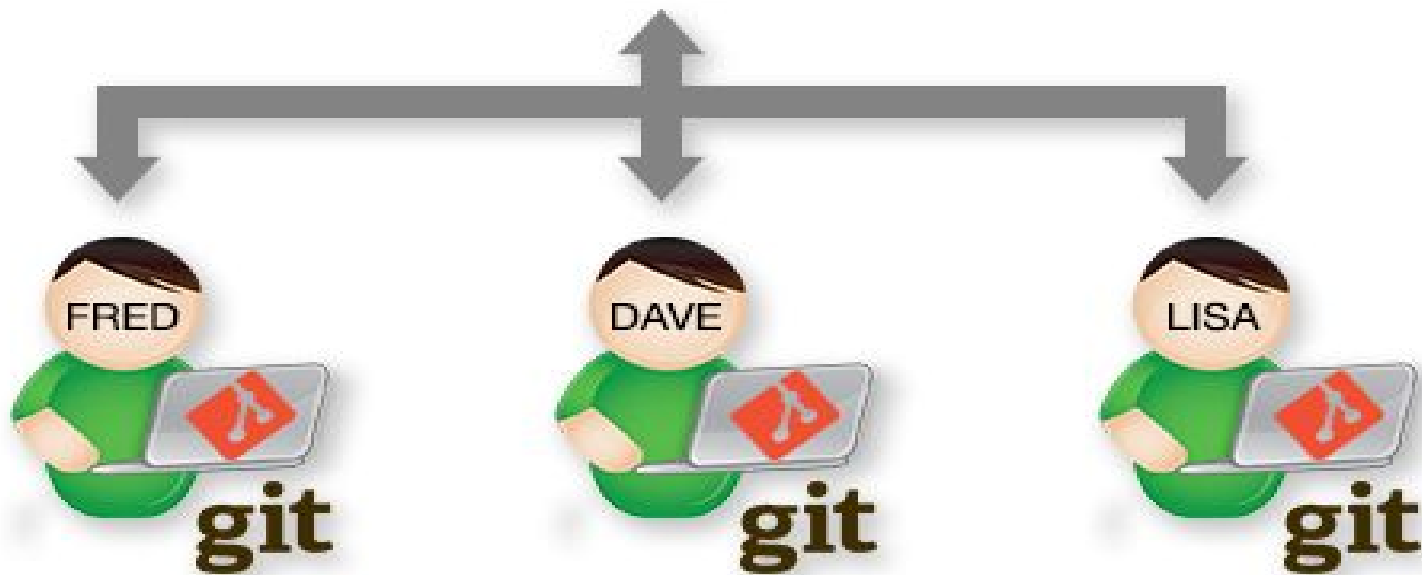
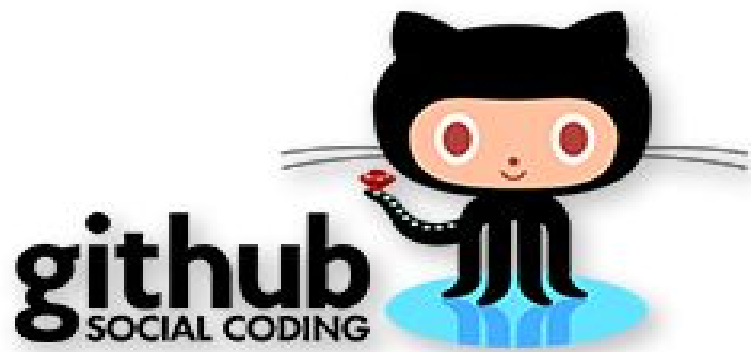
Git es el software que nos va ayudar versionar nuestro proyecto.

Github es el hosting que nos ayuda almacenar nuestro repositorio.

Plataforma puede distribuir un repositorio.

Tiene sus propias funciones, extiende los poderes de git.





Configuración de Git

Config

— — —

```
git config --global user.email  
cimaejemplo@gmail.com
```

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

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

Bonus el superlog!

```
git config --global alias.superlog  
"log --graph --abbrev-commit  
--decorate --date=relative  
--format=format:'%C(bold  
blue)%h%C(reset) - %C(bold  
green)(%ar)%C(reset)  
%C(white)%s%C(reset) %C(dim white)-  
%an%C(reset)%C(bold  
yellow)%d%C(reset)' --all"
```

Comandos Básicos

Comandos básicos

— — —

Untracked files: Son archivos que están en nuestro working directory, lo que aparezca en rojo es lo que se ha modificado y hay que pasarlo al staging.

git add nombreDeArchivo - Agrega el archivo al staging area

git rm -f nombreDeArchivo - elimina el archivo

git add -n nombreDeArchivo - No hace nada al archivo

git add -A - Agrega todos los archivos

git status - Ve el estado de tu working directory

Visualización de versiones

Comandos de versionamiento

— — —

git log - Muestra el historial de nuestro proyecto

git diff id1 id2 - Visualiza la comparación entre dos puntos diferentes en el historial de nuestro proyecto.

Push Pull, Merge

Comandos básicos

— — —

git pull - Actualiza tu repositorio local, con los cambios dados en el remoto

git commit -m 'descripción de commit' - Agrega un nuevo commit a tu repositorio local.

git push - Sube los commits creados a tu git repository

git remote add origin url -

git push --set-upstream origin nombreDeBranch - Actualiza tu repositorio local, con los cambios dados en el remoto si no está establecida una conexión.

Viaje a líneas alternas

Comandos básicos

— — —

git branch - Identifica el branch actual.

git checkout -b nombreDeBranch - crea un branch nuevo.

git checkout idDeBranch - crea un branch temporal.

git reset --soft - Nos permite quitar los cambios de un commit específico. Deja los archivos en el staging area, listos para hacer un commit.

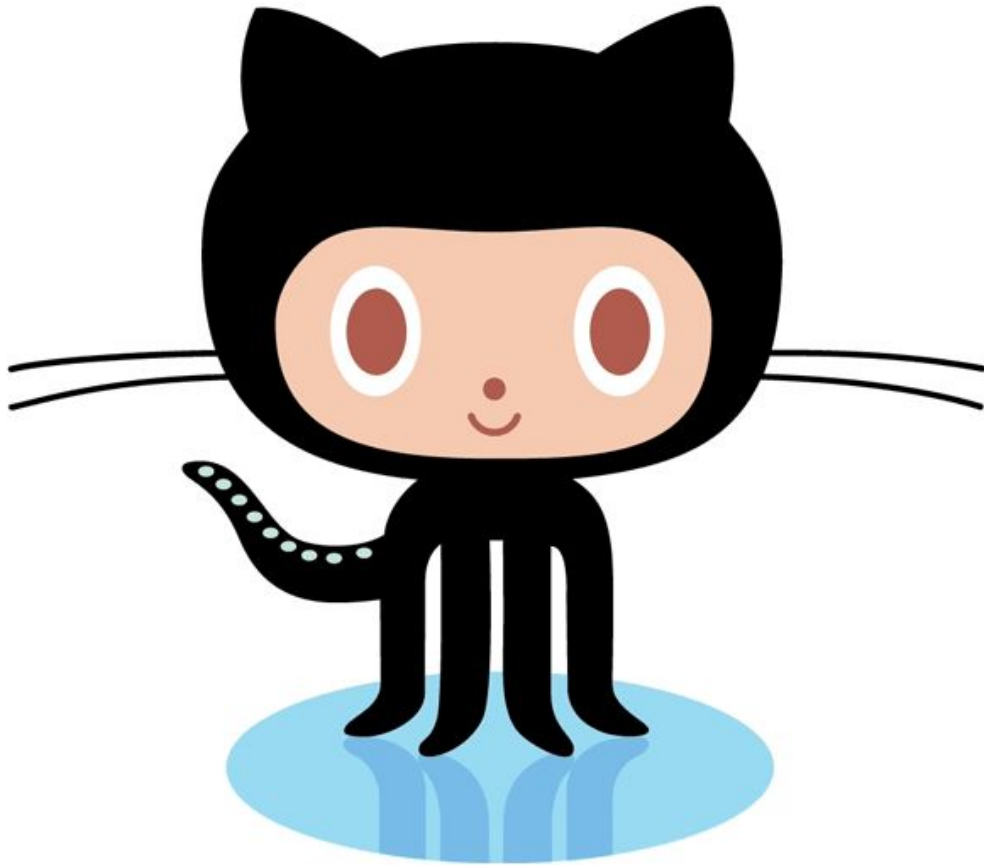
git stash - Elimina los cambios echos en el working directory

Temas Potenciales

Temas

— — —

1. Rebase
2. Reset (soft mixed hard)
3. Tag
4. Issues/Milestone
5. GitIgnore
6. Merge Issues
7. Cherry Picking Commits
8. Añadir llaves SSH



Gracias!

<https://github.com/saulflores95>
saulpolo95@gmail.com
flores.saul@uabc.edu.mx