

HUGE

Hello

PyLadies Colombia

Intro a Git & GitHub

Julio 18, 2018

Agenda.

1. Conceptos básicos
2. Que es Git
 1. Historia de Git
 2. Características de Git
 3. Palabras claves
 4. Comandos básicos
 5. Buenas prácticas
 6. Herramientas GUI

1

Conceptos básicos

Sistema de control de versiones (SCM)

Sistema para facilitar la administración de las distintas versiones de los datos.

Sistemas de control distribuido

Forma de control de todo el código base (incluido todo el historial), en donde todo es copiado en cada computador que haga una copia, para luego ser sincronizado cada vez que se hagan nuevos cambios.

Repositorio

Es un espacio centralizado donde se almacena, organiza, mantiene y difunde información digital

¿Qué es Git?

¿Qué es Git?



Es un software de control de versiones, pensado en la eficiencia y la confiabilidad del mantenimiento de archivos.

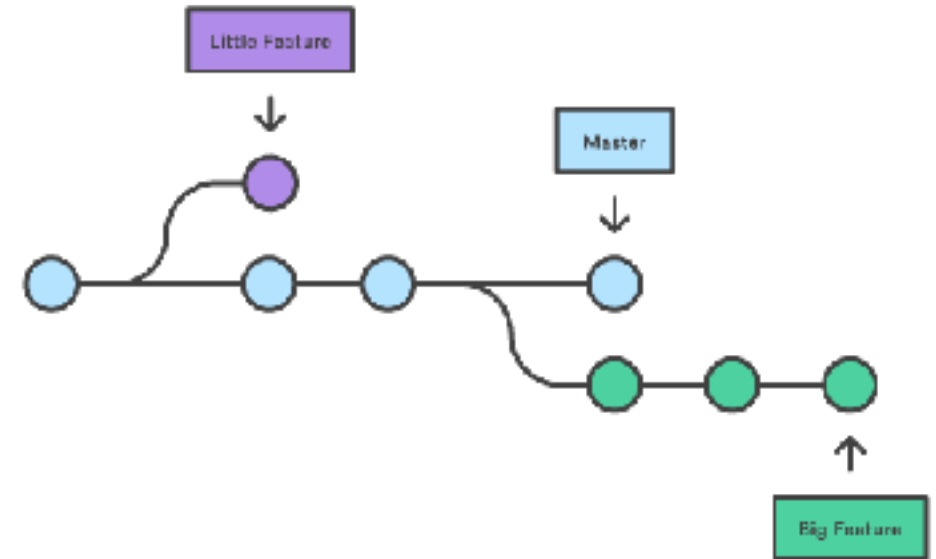
<https://git-scm.com/download/mac>

¿Historia de Git?

Fue creado por Linus Torvalds en 2005 para el desarrollo del Kernel de Linux, cuándo Bitkeeper (SCM propietario) decidió retirar la licencia de uso gratuito luego de que se violaran las condiciones de uso gratuito por parte del equipo de Linux.



Linus Torvalds



1. **Desarrollo no lineal**
2. **Sistema distribuido**
3. **Manejo eficiente en proyectos grandes**
4. **Historial completo de cambios**

Palabras claves



Repository



Clone



Branch



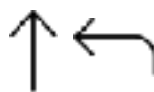
Commit



Remote



Push



Pull Request



Merge



Pull



Local

Comandos básicos

git pull

Descarga los cambios en el repositorio remoto y hace merge en el local.

git fetch

Descarga los cambios en el repositorio remoto.

git push origin <nombre_rama>

Sube la rama seleccionada al repositorio remoto.

git commit -am "<mensaje>"

Confirma los cambios realizados.

git merge <nombre_rama>

Unifica los cambios en la rama que me encuentre parado.

git status

Muestra el estado actual de la rama.

Comandos básicos

git add

Agrega el archivo al index de cambios.

git branch

Muestra la rama en la que estoy actualmente.

git reset - - hard HEAD

Elimina los cambios que aún no se hayan hecho commit.

git branch -b <nombre_rama_nueva>

Crea una nueva rama.

git revert <hash_commit>

Revierte el commit realizado.

Comandos básicos

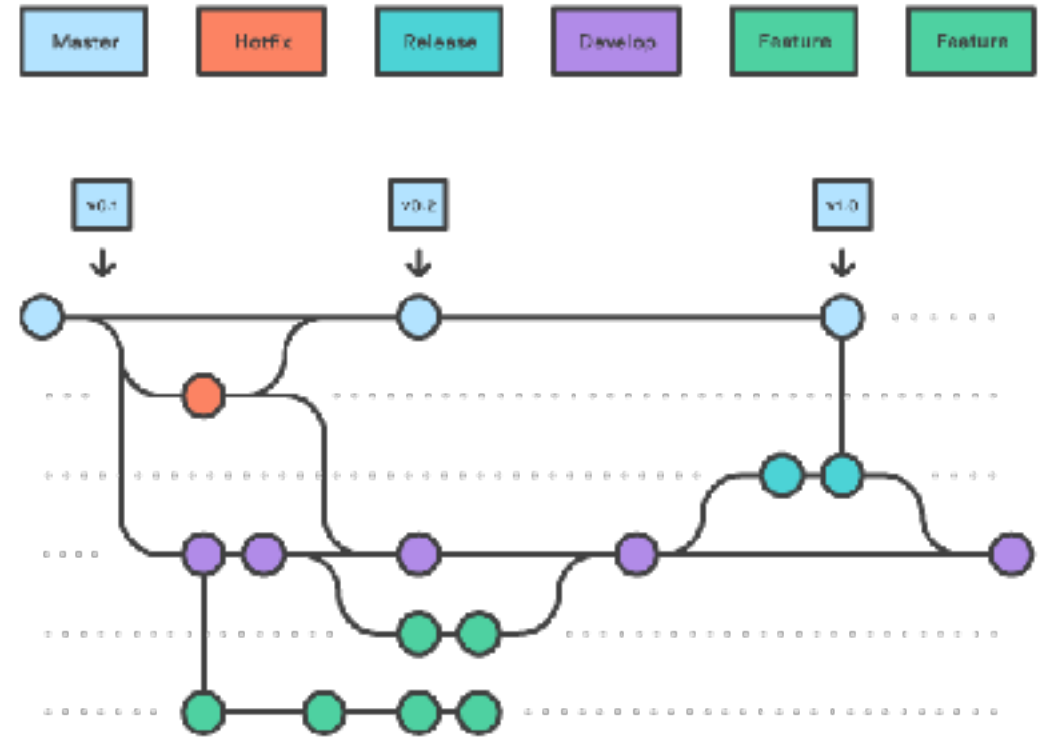
git diff

Muestra las diferencias del archivo seleccionado.

git stash

Guarda los cambios realizados en un estado provisional.

Buenas prácticas



1. Seguir workflow (GitFlow).

Master

Contiene toda la información de los lanzamientos oficiales

Develop

Sirve para como rama de integración de los nuevos “features”

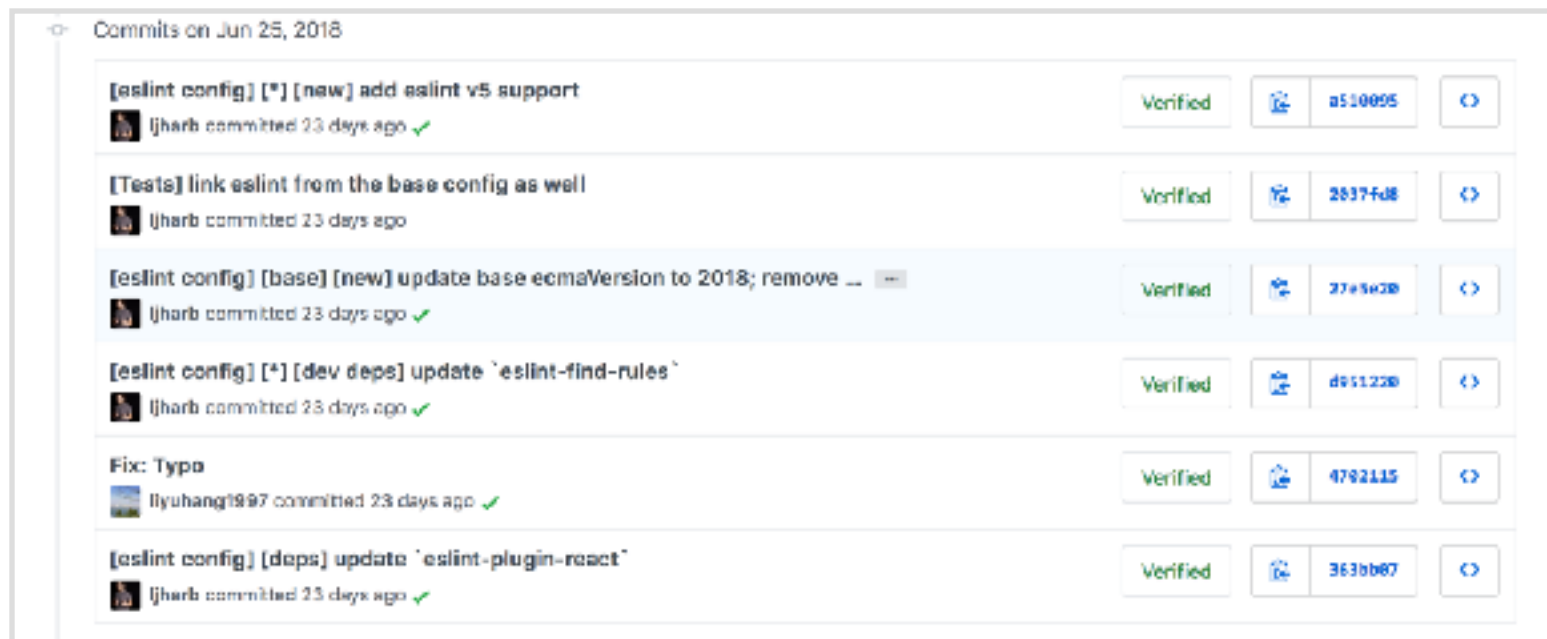
Hotfix

Es usado para ajustes rápidos en producción

Feature

Nuevos features para del producto

Buenas prácticas



1. Mensajes claros de commit.

Commits atómicos

Una tarea puntual debería reflejarse en un commit.

Plantilla de commits

Guía de como crear el mensaje del commit para facilitar la lectura y entendimiento

```
1  # gitignore
2
3  node_modules
4
5  # Only apps should have lockfiles
6  yarn.lock
7  package-lock.json
```

1. Crear archivo .gitignore

.gitignore

Sirve para definir que archivos ó carpetas se deben ignorar y no subir al repositorio.



Sourcetree



Github Desktop

HUGE

Done.

Intro a Git & GitHub

Julio 18, 2018