

# Git

M.C. Carlos Rojas Sánchez<sup>1</sup>

<sup>1</sup>Licenciatura en Informática  
Universidad del Mar :: Puerto Escodido

∴ Tools ∴



# git

# 1 Introducción

# ¿Qué es Git?

- SW desarrollado por Linus Torvalds
- Un versatil **CONTROLADOR DE VERSIONES DISTRIBUIDO**

# ¿Por qué usar algo como Git?

Caso Cuando usted tiene un archivo, y desea “guardarlo como”, o guardar varias versiones.

Solución Con Git, usted no tiene que hacerlo. Git mantiene "copias instantáneas" de cada punto en el tiempo, por lo que nunca se puede perder o sobrescribir.

El ambiente de Git es la línea de  
comandos  
¿Hay una interfaz?

# Github

- Es una plataforma de desarrollo colaborativo para alojar proyectos utilizando el sistema de control de versiones Git.
- Utiliza el framework Ruby on Rails por GitHub, Inc. (anteriormente conocida como Logical Awesome).



Entonce cuál utilizar Git o Github





# Git y Github se complementan

GitHub hace Git más fácil de usar, de dos maneras.

- Proporciona una interfaz visual.
- Características de redes sociales.



# Git y Github se complementan

- Puede navegar por los proyectos de otros usuarios GitHub, e incluso descargar copias para alterar y aprender.
- Otros usuarios pueden hacer lo mismo con los proyectos públicos, e incluso detectar errores y sugerir correcciones. De cualquier manera, no se pierden datos debido a Git guarda una "instantánea" de cada cambio.



# Conceptos

Repositorio: Es un directorio o espacio de almacenamiento. A veces los usuarios GitHub acortan esto a "repo". Puede ser local, o puede ser un espacio de almacenamiento en GitHub o otro host en línea. Se puede guardar archivos de código, archivos de texto, archivos de imágenes, lo que sea, dentro de un repositorio.



# Crea un repositorio nuevo

- Crea un directorio nuevo, ábrelo y ejecuta
  - `git init`
- Ver el estado
  - `git status`<sup>1</sup>

---

<sup>1</sup>este comando se utiliza para monitorear

# Ayuda en Git

- Para abrir los 21 comandos git más comunes
  - git help

# Especificando usuario

## ■ Local

- `git config - -local user.name` **usuario\_nombre**
- `git config - -local user.email` **"you@example.com"**

## ■ Remoto

- `git config - -global user.name` **usuario\_nombre**
- `git config - -global user.email` **"you@example.com"**

## ■ Comprobar

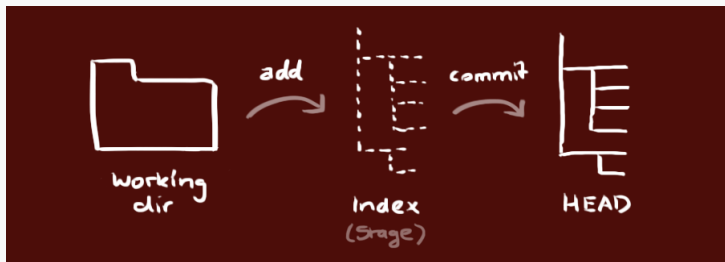
- `git config - -list`

# Trabajando con repositorios existentes

- Crea una copia local del repositorio
  - `git clone /path/to/repository`
- Si utilizas un servidor remoto, ejecuta
  - `git clone username@host:/path/to/repository`



# Flujo de trabajo



- El primero es tu **Directorio de trabajo** que contiene los archivos, el segundo es el **Index** que actúa como una zona intermedia, y el último es el **HEAD** que apunta al último commit realizado.



## add y commit

**add** Esto no agrega nuevos archivos a su repositorio. En cambio, trae nuevos archivos a la atención de Git. Después de agregar archivos, que están incluidos en "instantáneas" de Git del repositorio.

- `git add <filename>`
- `git add .`

**commit** Comando más importante de Git. Después de realizar cualquier tipo de cambio, que de entrada esto con el fin de tomar una "instantánea" del repositorio. Por lo general, va

- `git commit -m "Message here."`<sup>2</sup>

---

<sup>2</sup>El `-m` indica que es un mensaje.

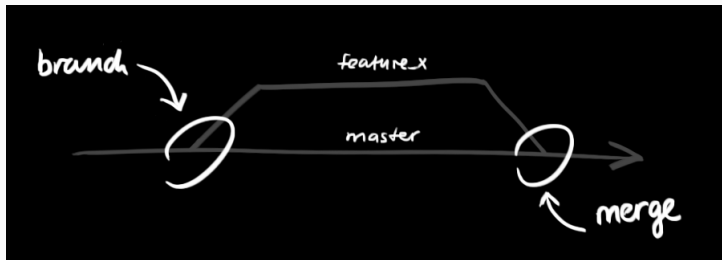
## Envío de cambios

- Tus cambios están ahora en el HEAD de tu copia local. Para enviar estos cambios a tu repositorio remoto ejecuta
  - `git push origin master`<sup>3</sup>
- Si no has clonado un repositorio ya existente y quieres conectar tu repositorio local a un repositorio remoto, usa
  - `git remote add origin <server>`

---

<sup>3</sup>Reemplaza master por la rama a la que quieres enviar tus cambios. 🔍 ↻

# Ramas



- Son utilizadas para desarrollar funcionalidades aisladas
- La rama master es la rama "por defecto" al crear un repositorio.
- Es posible crear nuevas ramas durante el desarrollo y posteriormente fusionarlas a la rama principal.

# Trabajando con ramas

- Crear una nueva rama llamada "feature\_x"
  - `git branch feature_x`
- Crea una nueva rama llamada "feature\_x" y cámbiate a ella usando
  - `git checkout -b feature_x`
- Volver a la rama principal
  - `git checkout master`
- Borra una rama
  - `git branch -d feature_x`



## Envío de cambios en ramas

- Una rama nueva no estará disponible para los demás a menos que subas (push) la rama a tu repositorio remoto
  - `git push origin <branch>`

# Actualizar y Fusionar

- Para actualizar tu repositorio local al commit más nuevo
  - `git pull`
- Para fusionar otra rama a tu rama activa (por ejemplo master), utiliza
  - `git merge <branch>`
- Antes de fusionar los cambios, puedes revisarlos usando
  - `git diff <source_branch> <target_branch>`



# Etiquetas

- Puedes crear una nueva etiqueta llamada 1.0.0 ejecutando
  - `git tag 1.0.0 1b2e1d63ff`
- 1b2e1d63ff se refiere a los 10 caracteres del commit id al cual quieres referirte con tu etiqueta. Puedes obtener el commit id con
  - `git log`
- también puedes usar menos caracteres que el commit id, pero debe ser un valor único.

## Reemplaza cambios locales

- En caso de que hagas algo mal (lo que seguramente nunca suceda) puedes reemplazar cambios locales usando el comando
  - `git checkout - <filename>`
- Este comando reemplaza los cambios en tu directorio de trabajo con el último contenido de HEAD. Los cambios que ya han sido agregados al Index, así como también los nuevos archivos, se mantendrán sin cambio.
- Por otro lado, si quieres deshacer todos los cambios locales y commits, puedes traer la última versión del servidor y apuntar a tu copia local principal de esta forma
  - `git fetch origin`
  - `git reset --hard origin/master`





## Datos útiles

- Interfaz gráfica por defecto
  - `gitk`
- Colores especiales para la consola
  - `git config color.ui true`
- Mostrar sólo una línea por cada commit en la traza
  - `git config format.pretty oneline`
- Agregar archivos de forma interactiva
  - `git add -i`



# Bibliografía I



git - la guía sencilla: una guía sencilla para comenzar con git.  
sin complicaciones ;).

*<https://rogerdudler.github.io/git-guide/index.es.html>*

por Roger Dudler (traducido por @lfbarragan y  
@adrimatellanes).



GitHub para los principiantes

*<http://www.impoweru.com/github-para-los-principiantes-no-te-asustes-comienza>*

2016