



git

**GIT**

Taller Introductorio

Malcolm Davis Steele



## Malcolm Davis Steele

CE student at Tecnológico de Costa Rica, Vice Chair IEEE TEC, Computer Society. Entrepreneur, Hardware Hacker. @malkam03

# Presentación

## Taller Introductorio

---



# Agenda

Taller Introductorio

- Introducción
- Primeros Pasos
- Conceptos Básicos de Git
- Flujo de trabajo de Git
- Buenas Prácticas
- La Comunidad Git



# Introducción

Taller Introductorio



# git Contexto

- Git es un sistema de control de versiones libre diseñado para desarrollar todo tipo de proyectos, grandes y pequeños con velocidad y eficacia.
- Git cuenta con la mayor comunidad de desarrolladores.
- Repositorios en la nube.
- Lo usan los grandes.

Google

facebook

Microsoft

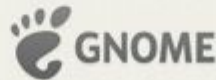
twitter

LinkedIn

NETFLIX



PostgreSQL





# Historia

- Mantenimiento del Kernel de Linux.
- Criterios de Diseño:
  - Velocidad
  - Diseño Simple
  - Soporte robusto para desarrollo en paralelo.
  - Entorno distribuido.
  - Capacidad para manipular grandes proyectos. (Linux Kernel)
- Creación de Git en 2005
- Soporte para proyectos de desarrollo no lineal.



# Primeros Pasos

Taller Introductorio

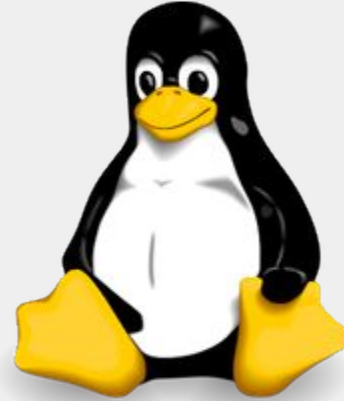
# Instalación



```
$ brew install git
```



```
Instalador .exe  
http://msysgit.github.com/
```



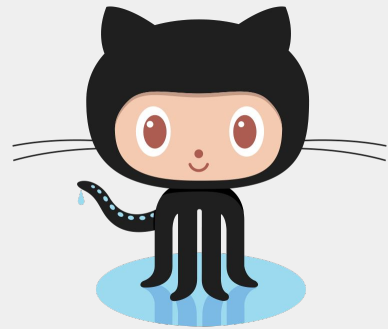
```
$ yum install git-core  
$ apt-get install git
```





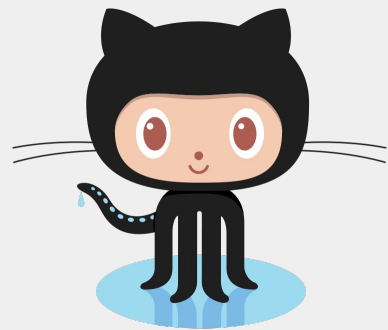
# GIT HUB

- Un usuario puede subir sus proyectos y darles una **licencia** que determinarán el uso que podremos hacer de los proyectos.
- Se pueden obtener **copias** de proyecto para seguir trabajándolo y/o modificarlo a su gusto.
- Cuenta con una característica llamada “**Branch**” (copia del proyecto) con la intención de trabajar sobre el mismo, sin afectar al original. Puede haber más de un Branch.
- Tiene diferentes plantillas de licencias como: **GPL**, **MIT** o **APACHE**.



# GIT HUB

- Cuenta con gráficas que muestran los **aportes** hechas por los usuarios, así como los miembros que trabajan sobre el
- Tiene características de **RED SOCIAL**, se tiene un perfil en el que aparecen sus contribuciones y los usuarios pueden seguir entre sí.
- Cada proyecto puede tener su propia **WIKI** con manuales e información referente a este.
- Una **organización** en GITHUB consiste en un repositorio o una serie de repositorios donde más de una persona son dueños del proyecto.



# Creación de cuenta

*“GitHub es el mayor proveedor de alojamiento de repositorios Git, y es el punto de encuentro para que millones de desarrolladores colaboren en el desarrollo de sus proyectos.”*

- Ingreso al sitio: <https://github.com>
- Creación de perfil y verificación mediante email.
- Configuración de cuentas de correo.



# Dentro de GitHub

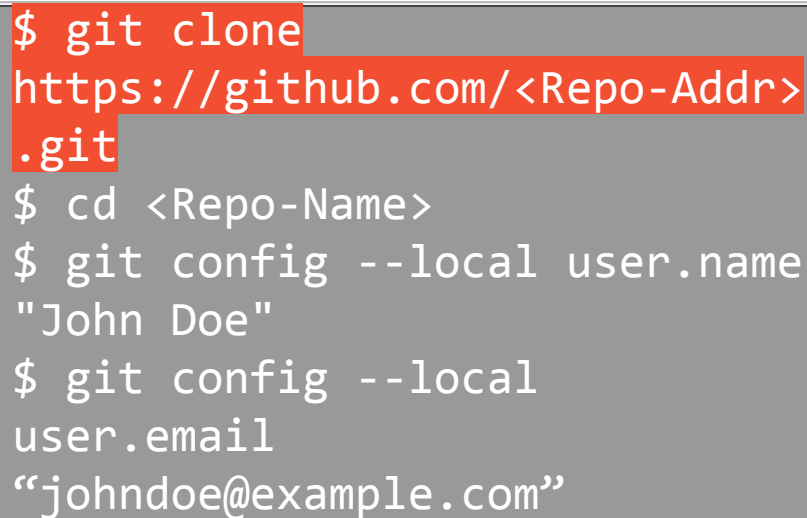
- Switch dashboard context.
- Repositories you contribute to.
- Your repositories...
- Create new...
- Notifications...
- Profile...



# Importancia de GitHub.

- Pull request
- Marketplace
- Gist
- Git Jobs
- Git Pages
- Comunidad Git

# Habilitar el entorno

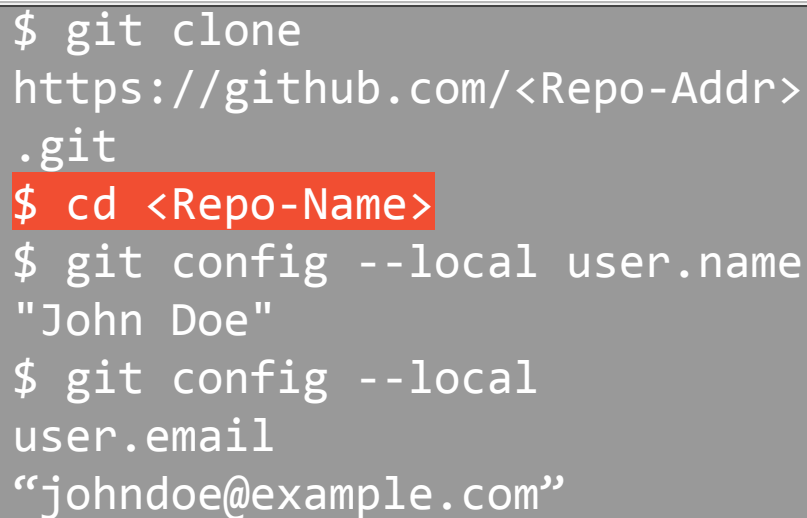


```
$ git clone https://github.com/<Repo-Addr>
.git
$ cd <Repo-Name>
$ git config --local user.name "John Doe"
$ git config --local user.email "johndoe@example.com"
```

## git clone

Este comando descarga el repositorio por primera vez para poder trabajar en ellos.

# Habilitar el entorno

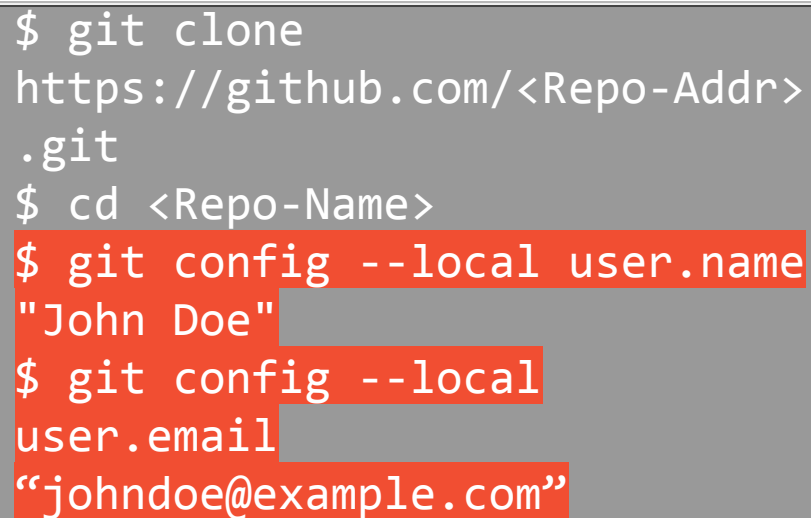


```
$ git clone  
https://github.com/<Repo-Addr>  
.git  
$ cd <Repo-Name>  
$ git config --local user.name  
"John Doe"  
$ git config --local  
user.email  
"johndoe@example.com"
```

**cd <directory>**

Este comando  
ingresa al  
directorio recién  
descargado para  
poder trabajar.

# Habilitar el entorno



```
$ git clone  
https://github.com/<Repo-Addr>  
.git  
$ cd <Repo-Name>  
$ git config --local user.name  
"John Doe"  
$ git config --local  
user.email  
"johndoe@example.com"
```

## **git config --local**

Este comando  
cambia los datos  
del archivo de  
configuración  
“.git/config”:  
user.name  
user.email





# Conceptos Básicos de Git

Taller Introductorio

Working  
Directory

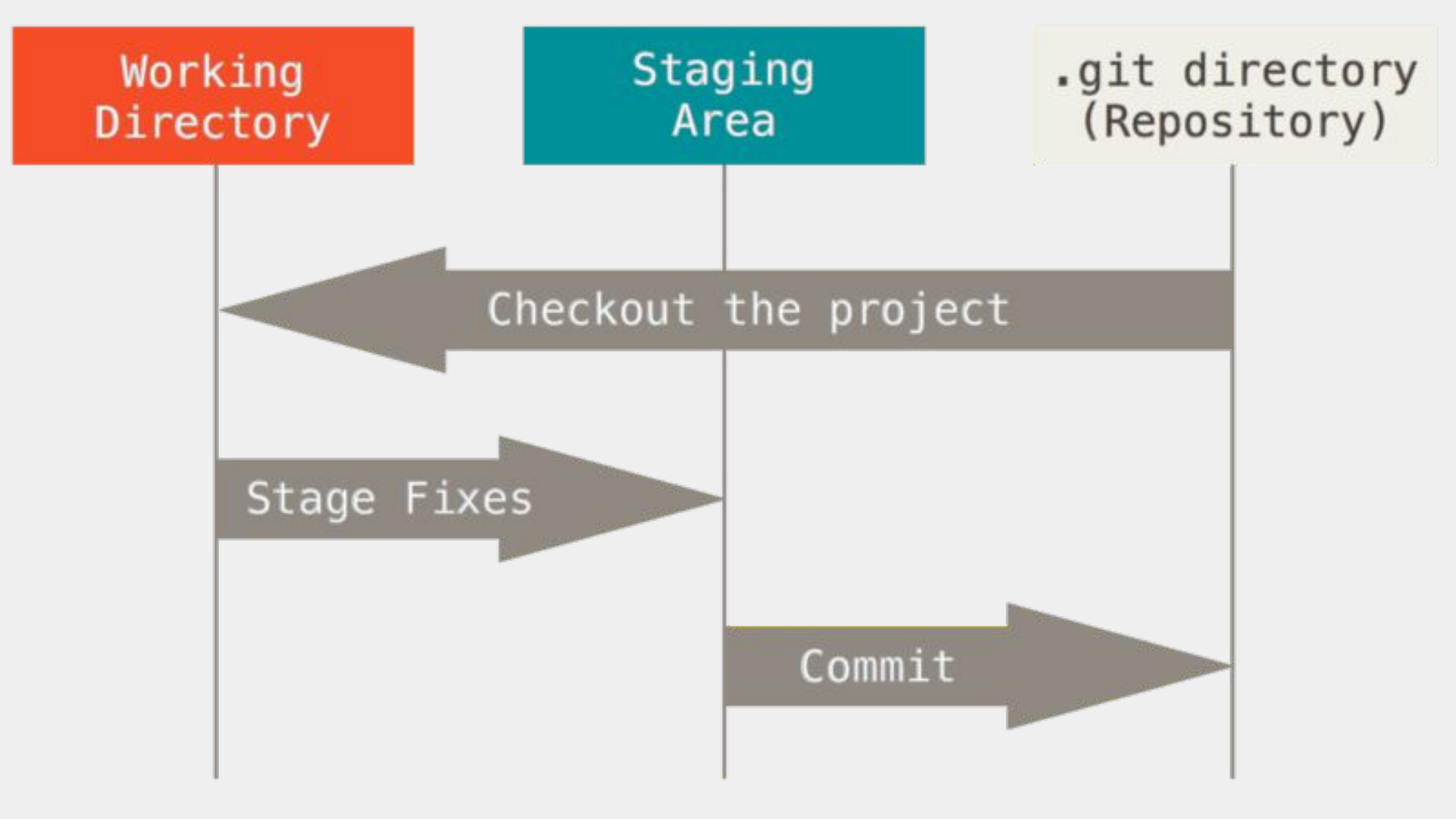
Staging  
Area

.git directory  
(Repository)

Checkout the project

Stage Fixes

Commit





# Ejercicio 1

***Objetivo:** Crear un repositorio local, respaldar cambios y control del histórico de los mismos. Por último agregar un repositorio remoto.*



# Ejercicio 2

*Objetivo: Actualizar el repositorio local y resolver conflictos de cambios hechos.*

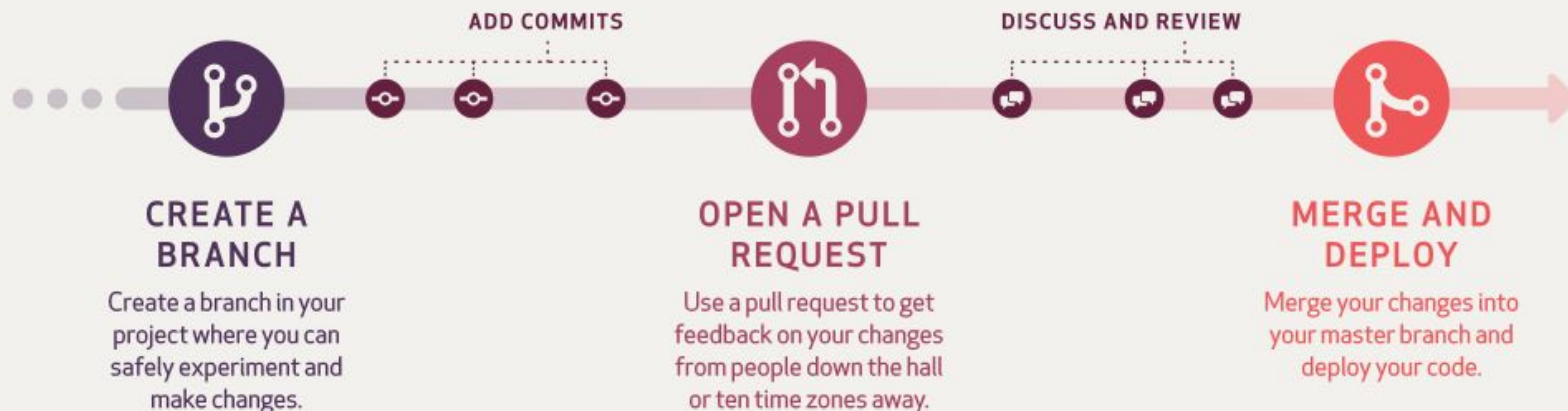


# Flujo de trabajo de Git

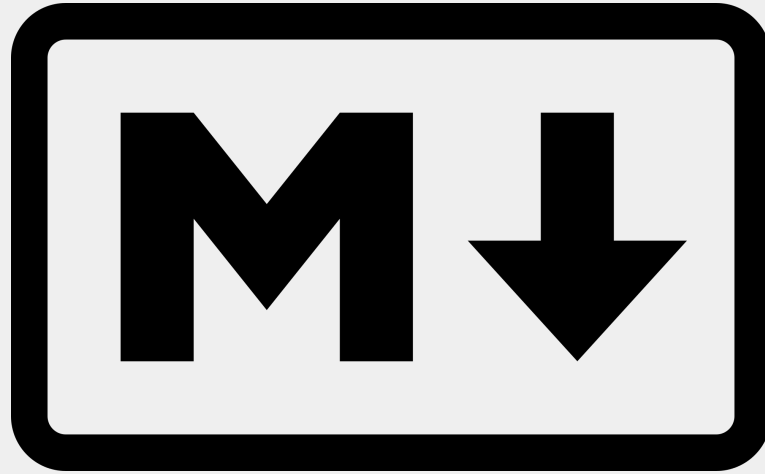
Taller Introductorio

# WORK FAST WORK SMART THE GITHUB FLOW

The GitHub Flow is a lightweight, branch-based workflow that's great for teams and projects with regular deployments. Find this and other guides at <http://guides.github.com/>.



# Markdown





# Ejercicio 3

**Objetivo:** Simulación de un ambiente común en el desarrollo en paralelo mediante git.  
*The Branch Strategy*





# Buenas Prácticas

Taller Introductorio

## Respaldo de cambios asociados. *“Commit related changes”*

---

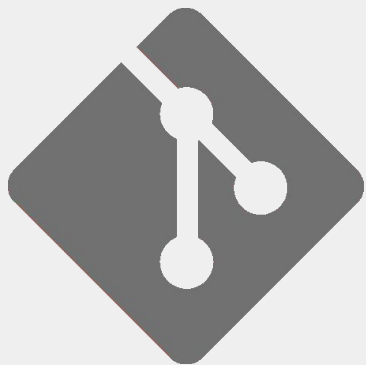
- Cambios granulados.
  - Semejanza entre el trabajo hecho.
  - Facilidad para hacer rollback.
  - No dejar todos los cambios para un solo commit.
  - Mayor velocidad para compartir.
  - Reducción de riesgo de conflictos.
  - Realizar pruebas antes de respaldar los cambios realizados.
-

## Estándar para los comentarios de los commits.

*“Commit related changes”*

- Do not assume the reviewer understands what the original problem was.
  - Do not assume the code is self-evident/self-documenting.
  - Describe why a change is being made.
  - Insert a single blank line after the first line.
  - The first line should be limited to 50 characters and should not end with a period.
-

# Felicidades!



git

**GIT**

Taller Introductorio

Malcolm Davis Steele  
Juan E. Navarro