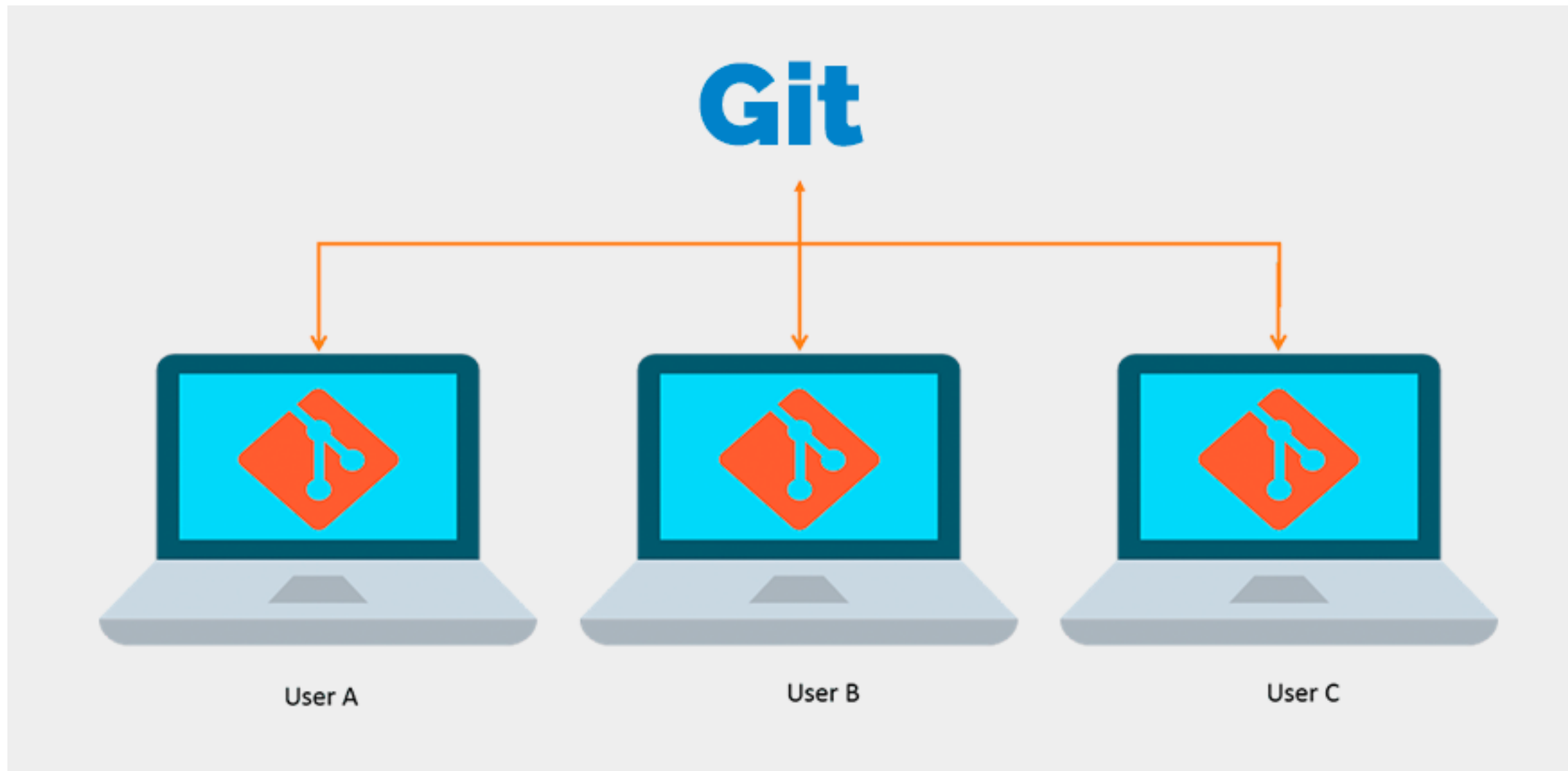




**git**

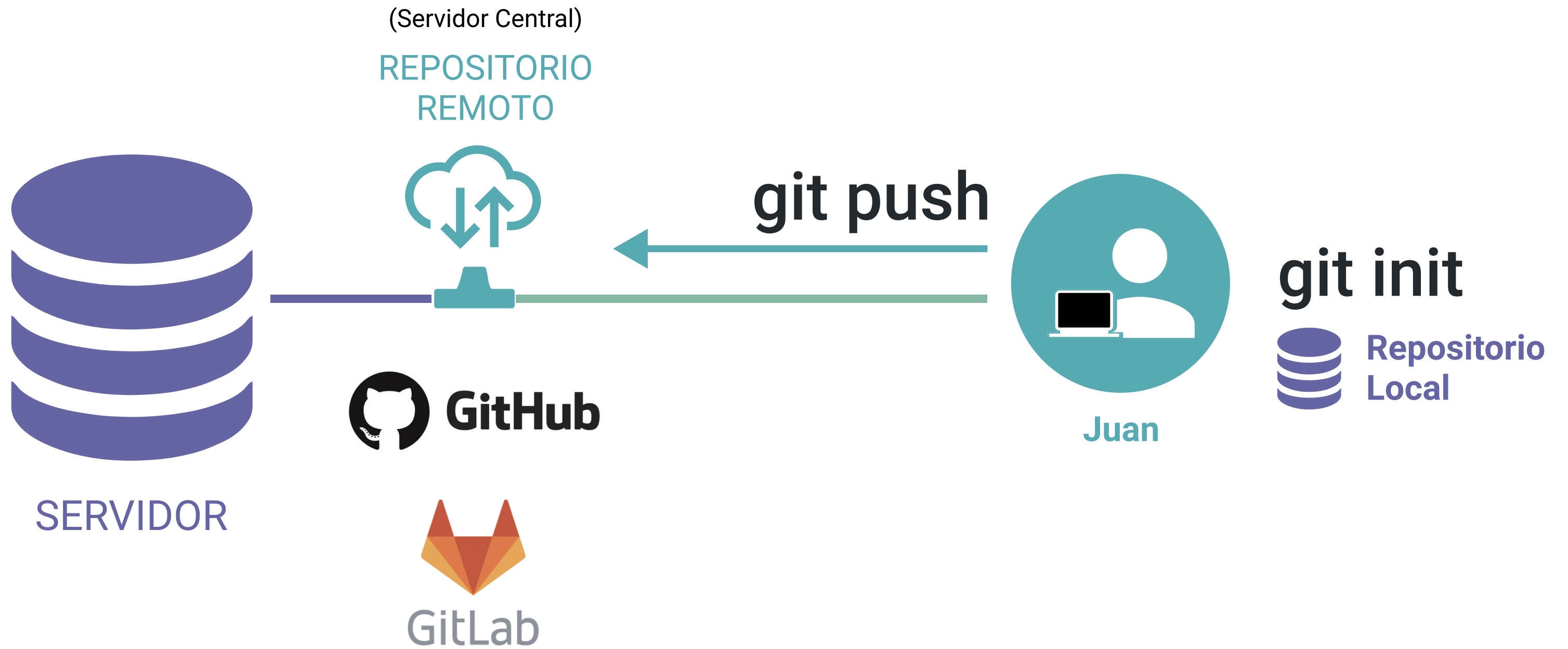
**Git es un sistema de control de  
versiones distribuido**

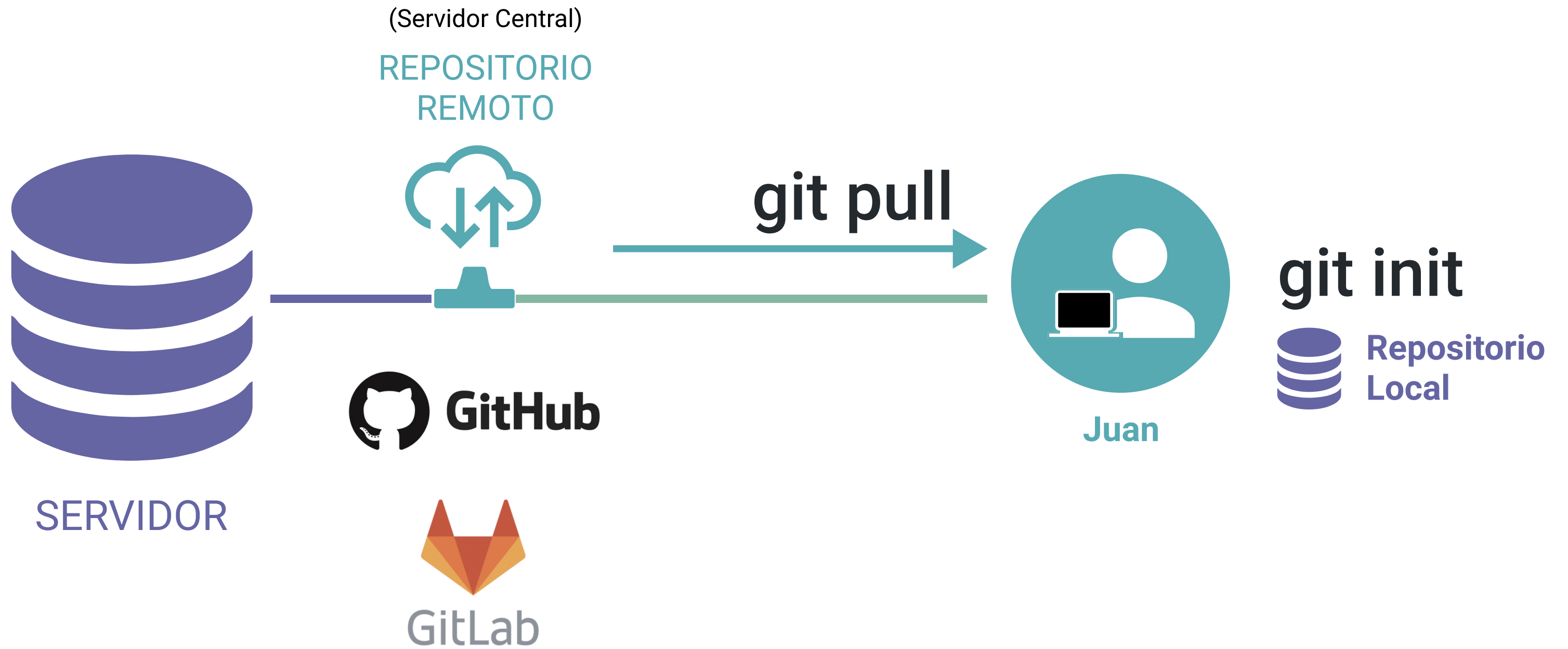
# Git es un sistema de control de versiones **distribuido**



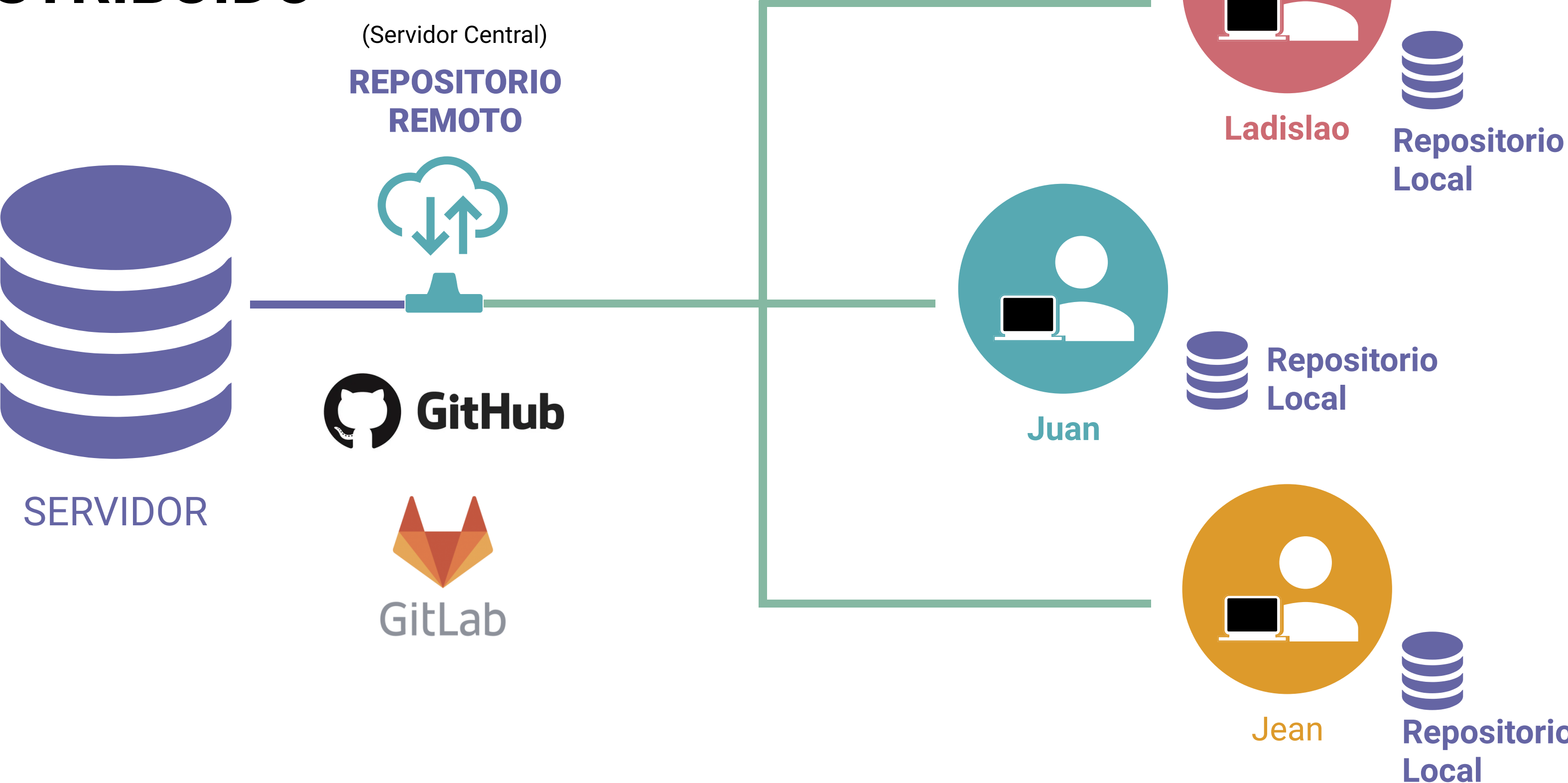
# Otras ventajas de trabajar con GIT

- Completamente distribuido
- Capaz de manejar grandes proyectos
- Es muy rápido
- Diseño sencillo
- Rastrea todos los cambios
- Acceso sin internet
- Open source
- Hace copias de seguridad
- Mantiene una historia
- Ver los cambios
- Experimentos
- Trabajo colaborativo





# SISTEMA DE CONTROL DE VERSIONES DISTRIBUIDO



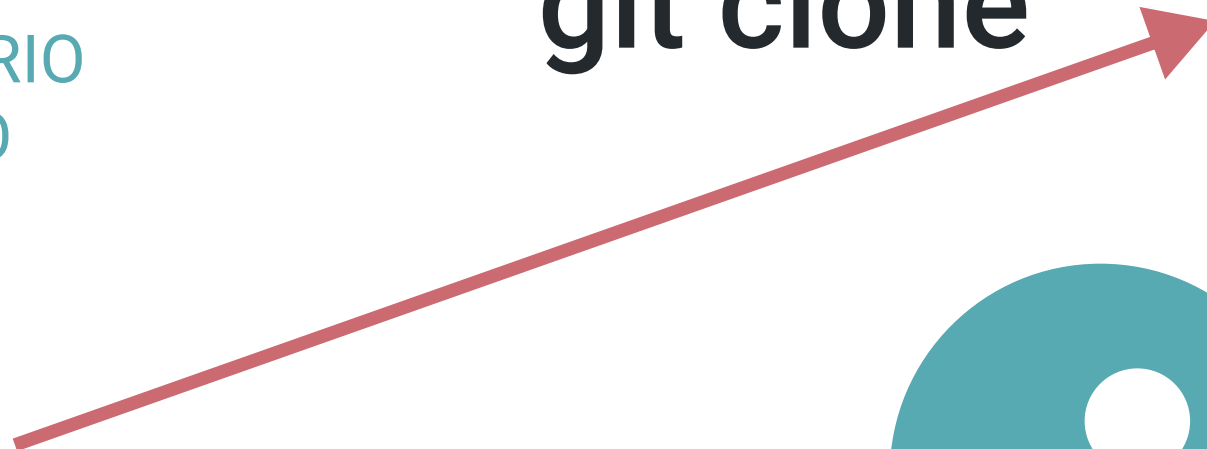


SERVIDOR

(Servidor Central)  
REPOSITORIO  
REMOTO



git clone



Juan



Ladislao



Jean





# ¿Qué es un repositorio?

Un repositorio de Git es un almacenamiento virtual de tu proyecto. Te permite guardar versiones del código a las que puedes acceder cuando lo necesites.

- Repositorio local
- Repositorio remoto

# ¿Qué es un clone?

Es una copia del repositorio central, el comando de clonación es la manera más común de obtener una copia de desarrollo local.

# ¿Qué hace el comando **git init**?

El comando `git init` crea un nuevo repositorio de Git. Puede utilizarse para convertir un proyecto existente y sin versión en un repositorio de Git o inicializar un nuevo repositorio vacío.

Es el primer comando que se ejecuta en los proyectos nuevos.

# ¿Qué hace el comando **push/pull**?

**git push:** El comando `git push` se usa para cargar contenido del repositorio local a un repositorio remoto.

**git pull:** El comando `git pull` se emplea para extraer y descargar contenido desde un repositorio remoto y actualizar al instante el repositorio local para reflejar ese contenido.

# Instalación

# Instalación

## Windows

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

## Mac

`brew install git`

## Linux Ubuntu

`apt-get install git`

<https://www.atlassian.com/git/tutorials/install-git>

Git is a [free and open source](#) distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

Git is [easy to learn](#) and has a [tiny footprint with lightning fast performance](#). It outclasses SCM tools like Subversion, CVS, Perforce, and ClearCase with features like [cheap local branching](#), convenient [staging areas](#), and [multiple workflows](#).



### About

The advantages of Git compared to other source control systems.



### Documentation

Command reference pages, Pro Git book content, videos and other material.



### Downloads

GUI clients and binary releases for all major platforms.



### Community

Get involved! Bug reporting, mailing list, chat, development and more.



**Pro Git** by Scott Chacon and Ben Straub is available to [read online for free](#). Dead tree versions are available on [Amazon.com](#).



Mac GUIs



Tarballs



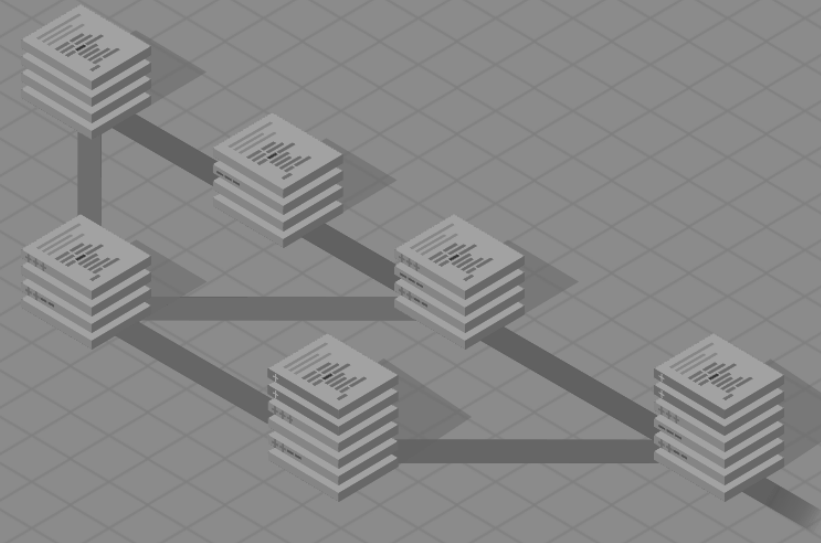
Windows Build



Source Code

Git is a **free and open source** distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

Git is **easy to learn** and has a **tiny footprint** with **lightning fast performance**. It outclasses SCM tools like Subversion, CVS, Perforce, and ClearCase with features like **cheap local branching**, convenient staging areas, and **multiple workflows**.



### About

The advantages of Git compared to other source control systems.



### Documentation

Command reference pages, Pro Git book content, videos and other material.



### Downloads

GUI clients and binary releases for all major platforms.



### Community

Get involved! Bug reporting, mailing list, chat, development and more.



**Pro Git** by Scott Chacon and Ben Straub is available to read online for free. Dead tree versions are available on [Amazon.com](#).

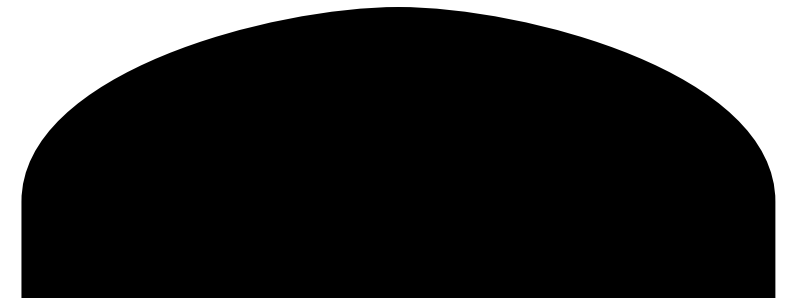


# Configuración

# ¿Cómo GIT rastrea nuestros cambios?

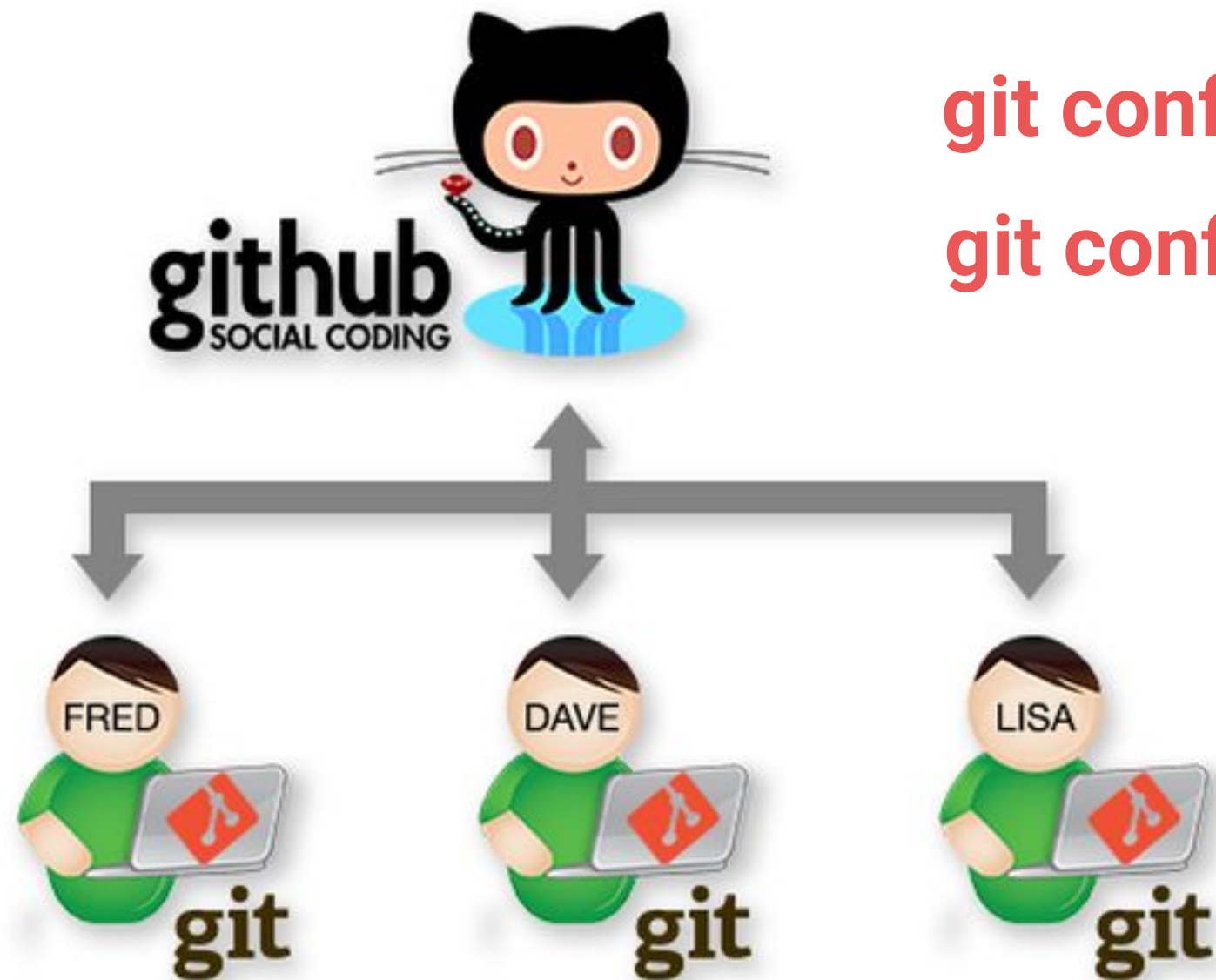


git





# Configuración



**git config --global** user.name "Juan Kuga"

**git config --global** user.email [juankp3@gmail.com](mailto:juankp3@gmail.com)

# Comandos útiles en la terminal



# Comandos útiles en la terminal

cd

ls

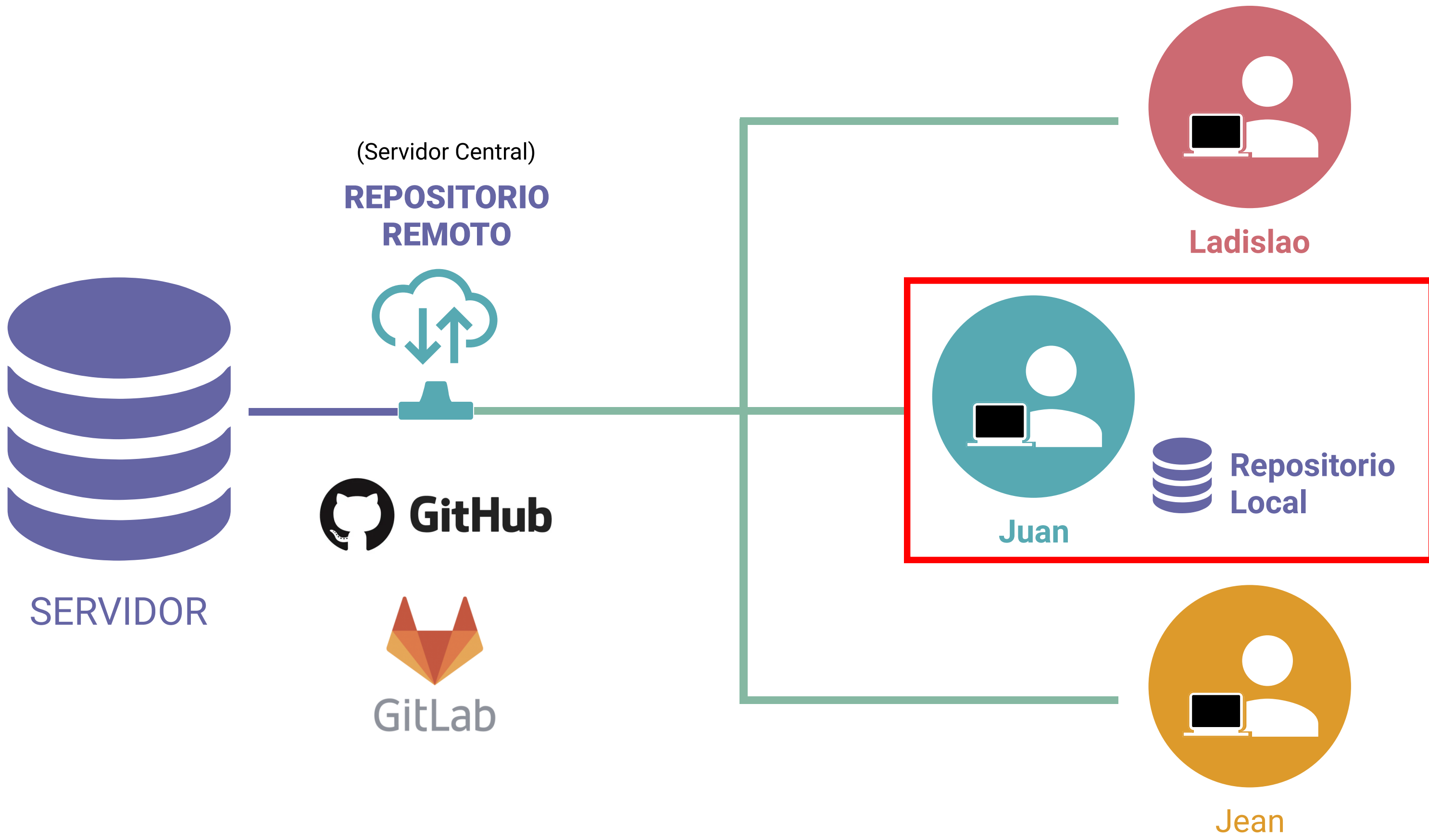
mkdir

touch

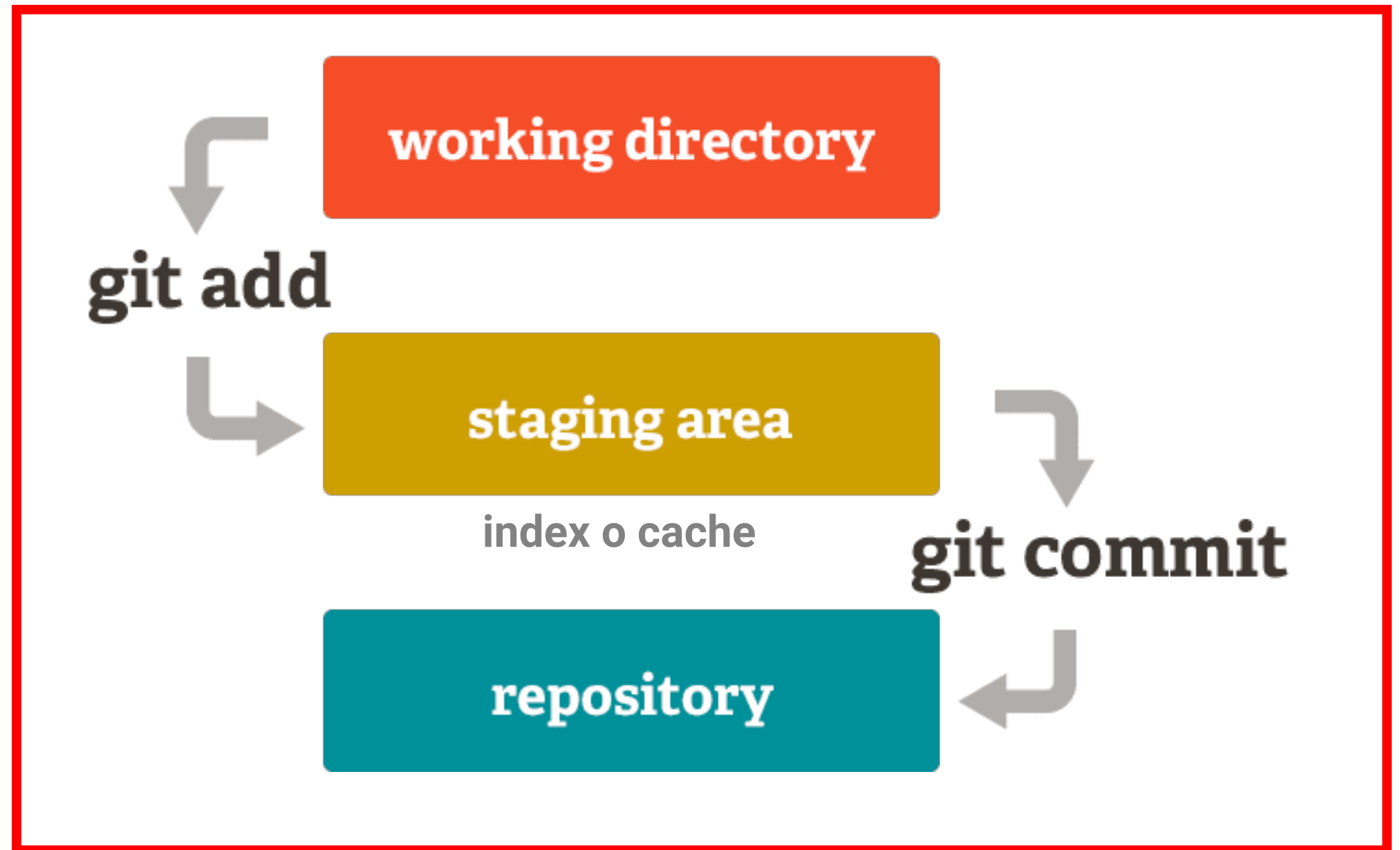
cat

vim

# Las 3 áreas de GIT



# Las 3 áreas de GIT



## Working directory

Es la carpeta de nuestro proyecto.

## Staging area

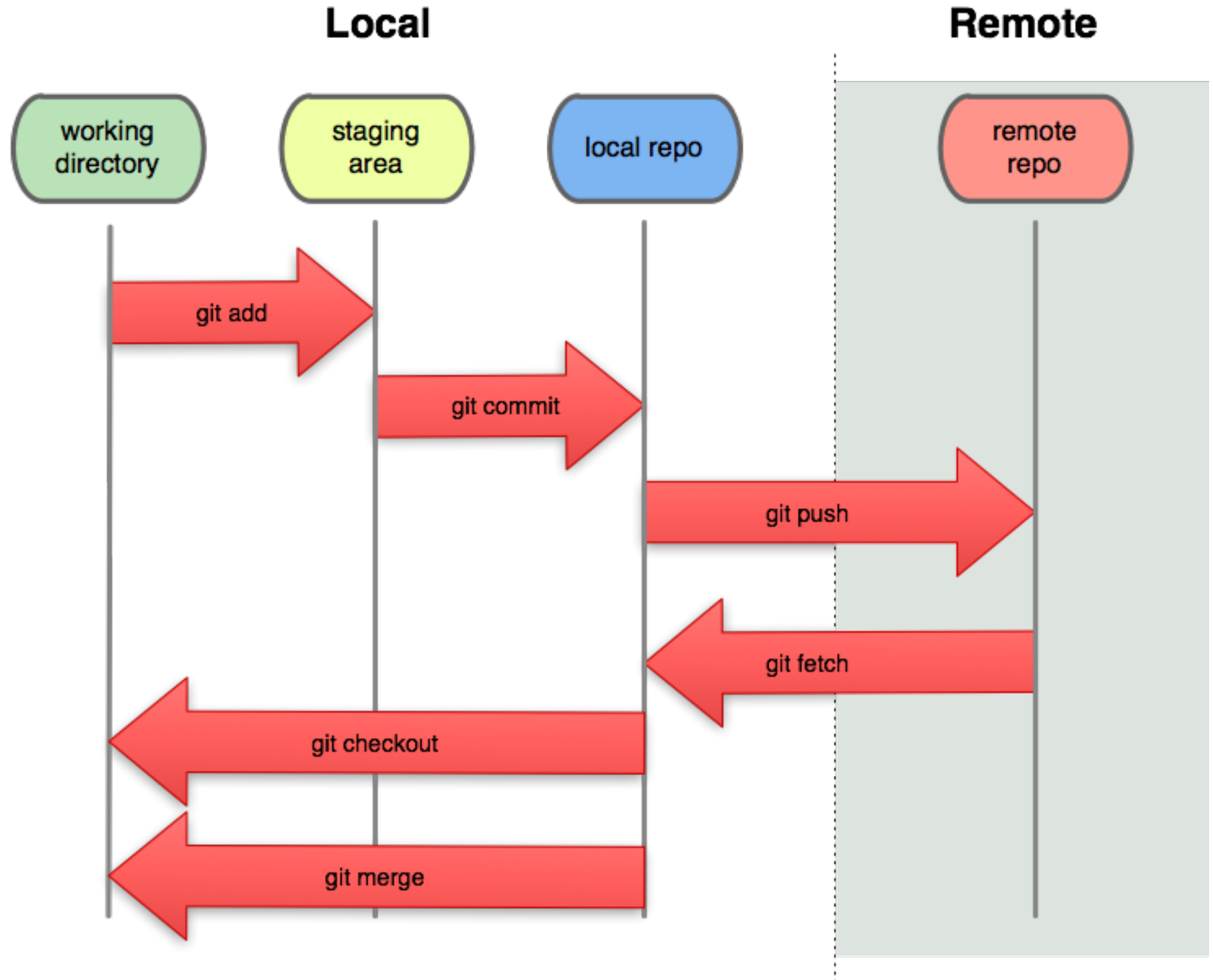
Donde pondremos los archivos a trackear.

## Repository

Donde se almacena toda la información de los cambios.



# Las 3 áreas de GIT



# Los 3 estados de archivos (cambios)



# Estados de archivos (cambios)

