



UNIVERSIDAD PRIVADA DE TACNA

# Git y GitHub

Curso: Programación II

Ing. Enrique Lanchipa Valencia



# Resumen

**Sistema Operativo de Disco (D.O.S)**



**Comandos básicos Windows**



**Comandos básicos Linux**



**GIT**



**GITHUB**



# Sistema Operativo de Disco



# Sistema Operativo de Disco D.O.S

MS-DOS significa Micro-Soft Disk Operating System (Sistema Operativo en Disco) .

Creado en 1981 por IBM computers. DOS fue el S.O. adoptado inicialmente por la mayoría de los computadores personales

Este sistema operativo no usaba ninguna interfaz gráfica ni ratón. No tenía tecnología Plug-and-play. Todo debía ser configurado manualmente.

No era ni multiusuario ni multitarea.

Por este motivo MS DOS ha sido sustituido por la familia de sistemas Windows de Microsoft



# Comandos básicos Windows

Comando	Sintaxis	Uso
<b>MD</b>	C:>MD [nombre del directorio]↵	Crear directorios
<b>CD</b>	C:>CD [nombre del directorio]↵	Cambiarse de directorio
<b>RD</b>	C:>RD [nombre del directorio]↵	Borrar directorio
<b>DIR</b>	C:\[nombre del directorio]>DIR↵ C:>\DIR [nombre del directorio]↵	Ver contenido del directorio
<b>COPY</b>	C:>COPY[nombre del directorio] [lugar]:↵	Copiar archivos
<b>DEL</b>	C:>DEL [nombre del directorio] [lugar]↵	Borrar archivos
<b>MOVE</b>	C:>MOVE [nombre del archivo o directorio]↵	Mover de ubicación
<b>HELP</b>	C:>HELP [comando]↵	Buscar ayuda
<b>EXIT</b>	C:>EXIT↵	Salir de DOS

# Comandos básicos Linux

Comando	Sintaxis	Uso
<b>MKDIR</b>	C:>MKDIR [nombre del directorio]↵	Crear directorios
<b>CD</b>	C:>CD [nombre del directorio]↵	Cambiarse de directorio
<b>RMDIR</b>	C:>RMDIR [nombre del directorio]↵	Borrar directorio
<b>LS</b>	C:\[nombre del directorio]>LS↵ C:>\LS [nombre del directorio]↵	Ver contenido del directorio
<b>CP</b>	C:>CP[nombre del directorio] [lugar]:↵	Copiar archivos
<b>DEL</b>	C:>DEL [nombre del directorio] [lugar]↵	Borrar archivos
<b>MV</b>	C:>MV [nombre de archivo o directorio]↵	Permite mover de ubicación o renombrar
<b>TOUCH</b>	C:>TOUCH [nombre y extensión de archivo]↵	Crear un archivo en blanco
<b>PWD</b>	C:>PWD↵	Muestra la ruta del directorio actual de trabajo
<b>HELP</b>	C:>HELP [comando]↵	Buscar ayuda
<b>EXIT</b>	C:>EXIT ↵	Salir de DOS



# Control de Versiones



CControl de  
versiones ultimo



Cntrl revision2



Cntrl revision3



Control bueno



Control bueno  
corregido



Control de  
versiones 1



Control de  
versiones 1.2



Control de  
versiones Final



Control de  
versiones Final1



Control de  
versiones Final2



Control de  
versiones Revisado



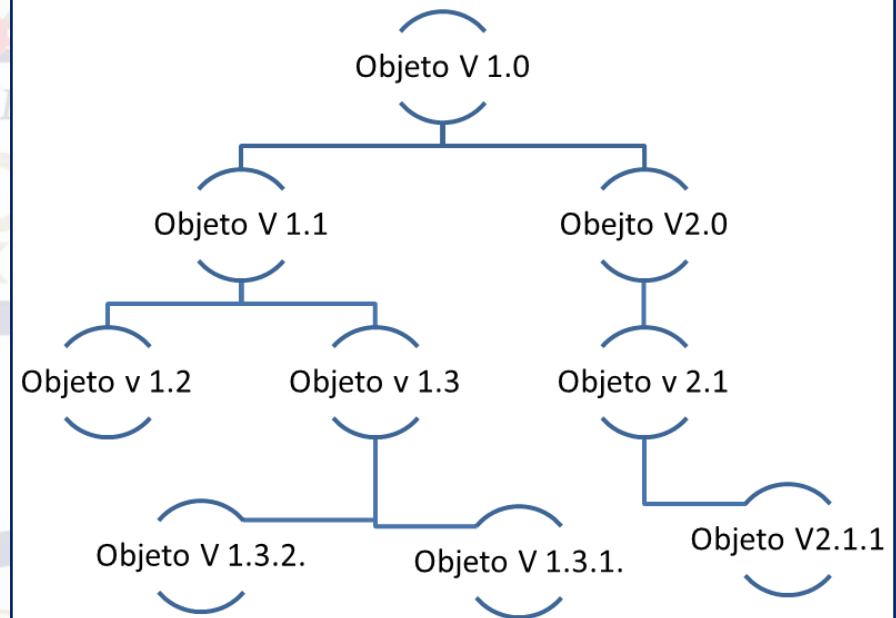
CONTROOOOL



Versiones listo



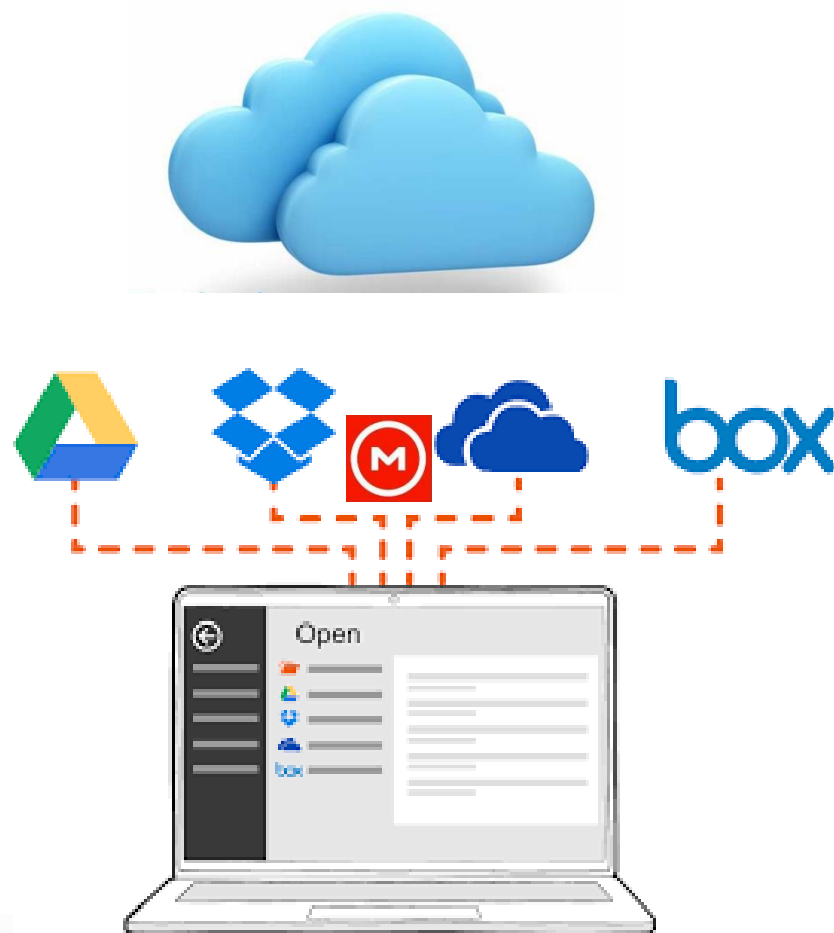
Versiones listo (2)





# ¿Es suficiente almacenamiento en la nube?

¿Cómo  
**compartimos**  
archivos en la  
**actualidad?**



# Git



Git es un sistema de control de versiones libre y de código abierto, su propósito es llevar registro de los cambios en archivos de la computadora y coordinar el trabajo que varias personas realizan sobre archivos compartidos para manejar desde pequeños hasta grandes proyectos con rapidez y eficiencia.

Git fue creado por Linus Torvalds en el 2005.



Conocimientos básicos previos:

1. Manejo del terminal o CMD (Linux / Windows)
2. Manejo de Sistema de archivos (directorios y archivos)
3. Aplicaciones Web (no indispensable)



## Beneficios del control de versiones



Compartir información siempre actualizada

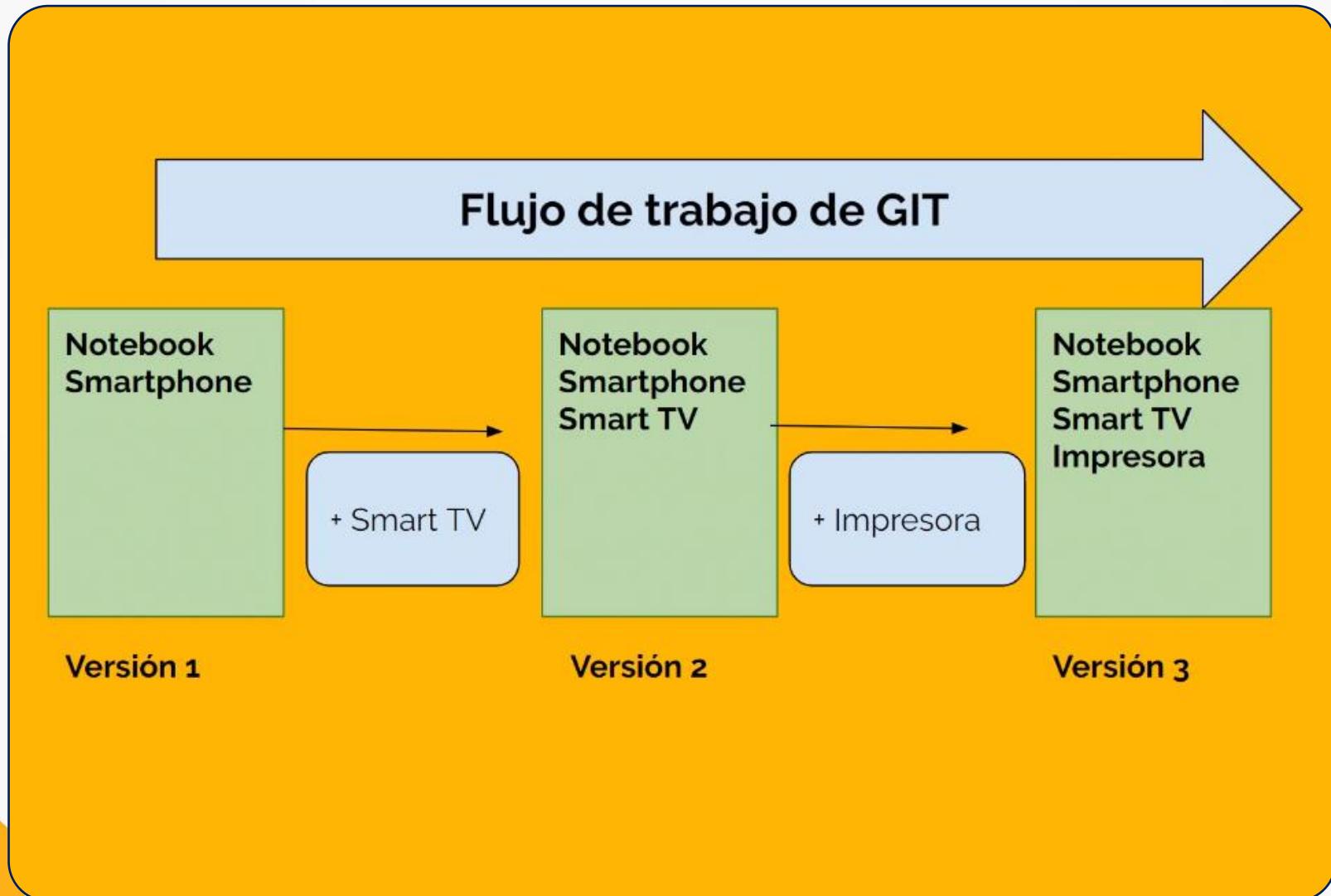


Evitar duplicación del contenido



Eliminar contenido no utilizado o antiguo

# Control de versiones



# Control de versiones

**Cambio 1**



Index.html

```
<DOCTYPE html>
```

**Cambio 2**



Index.html

```
<DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
</body>
```

```
</html>
```

**Cambio 3**



Index.html

```
<DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
...
```

```
</head>
```

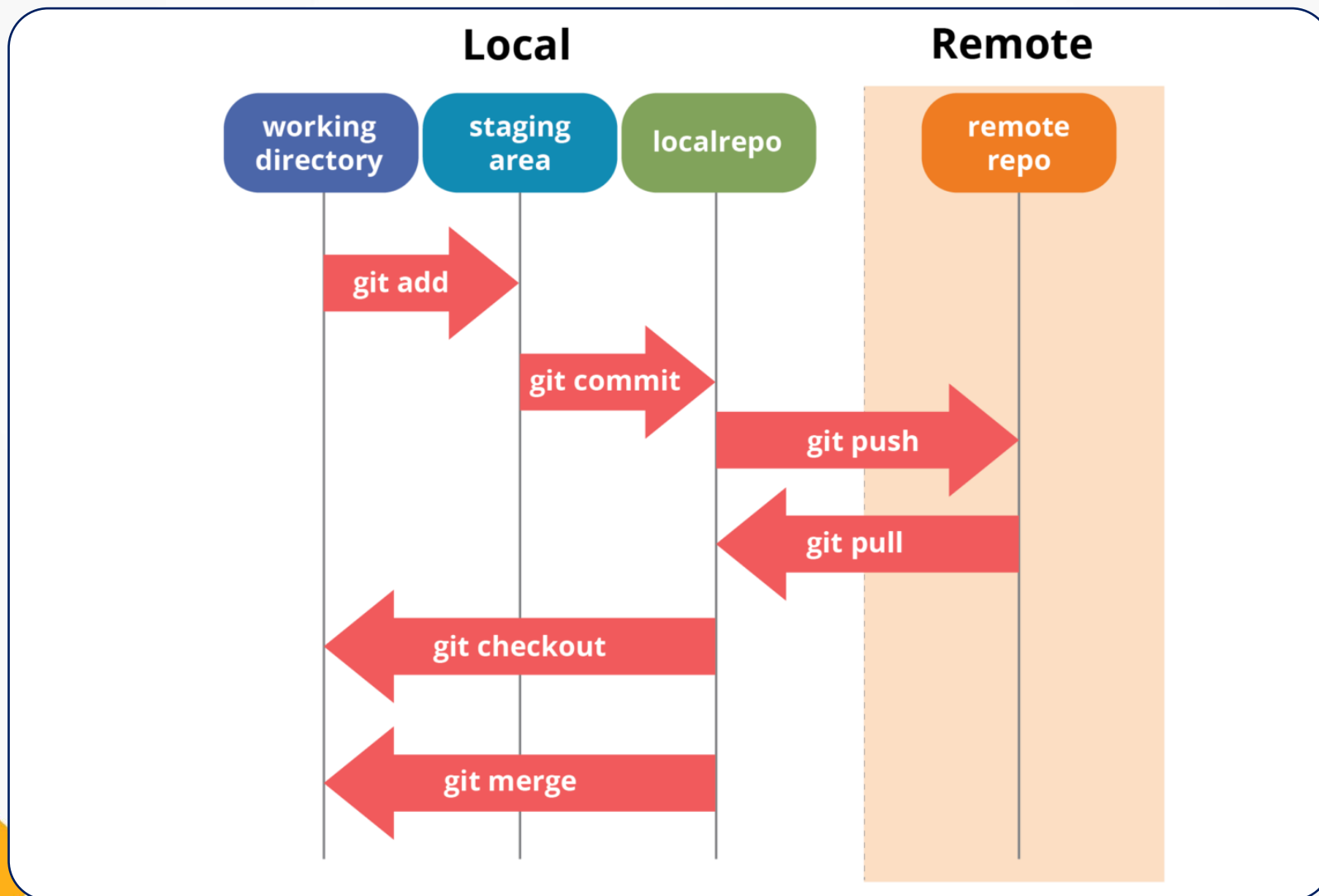
```
<body>
```

```
...
```

```
</body>
```

```
</html>
```

# Zonas de trabajo de GIT



# Conceptos importantes

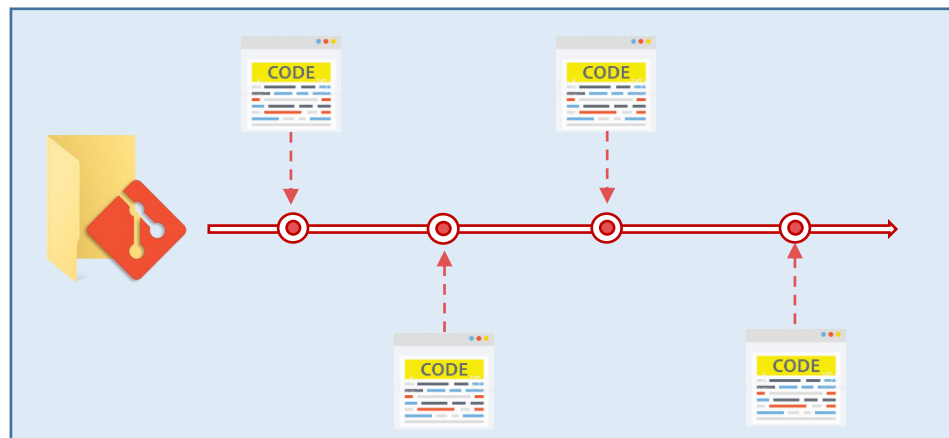
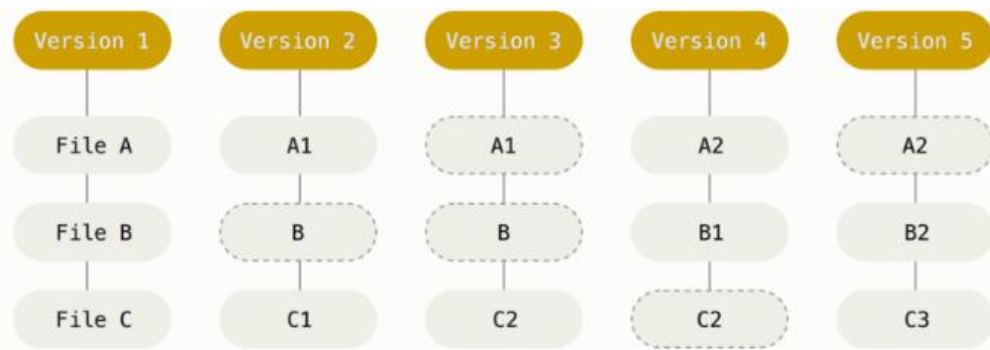
## Repositorio

Esto todo proyecto que ha sido iniciado en GIT y cuenta con un historial



## Commit

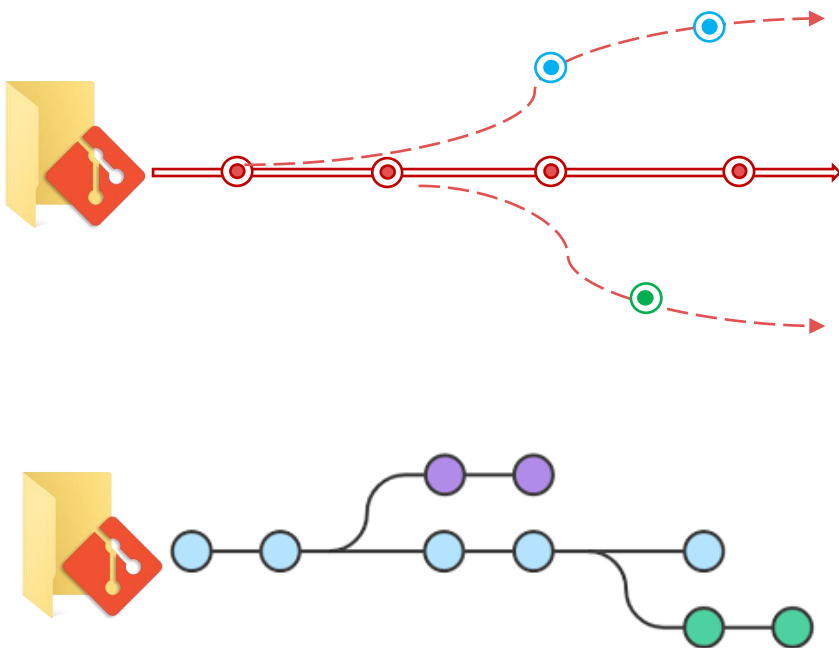
Es cada uno de los cambios registrados en el historial de GIT



# Conceptos importantes

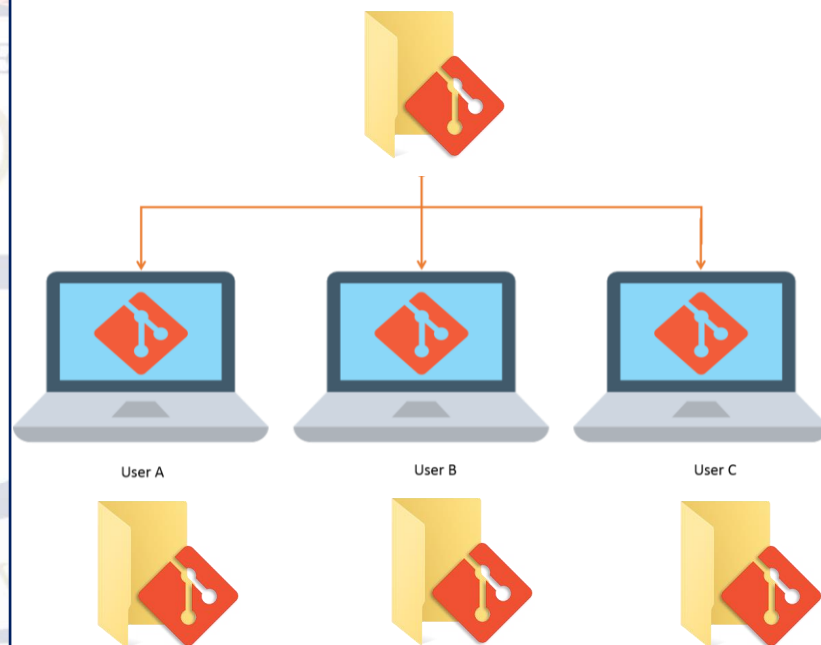
## Ramas (Branch)

Ramificaciones o bifurcaciones del proyecto que permite trabajar el proyecto de forma aislada



## Clon

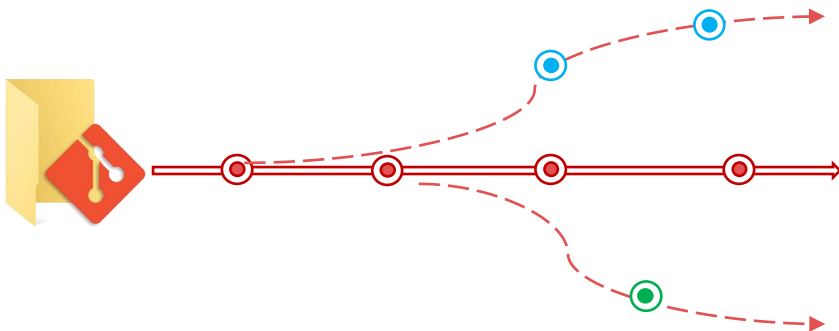
Es una copia exacta del proyecto



# Conceptos importantes

## Origin Master

Es la rama principal del proyecto

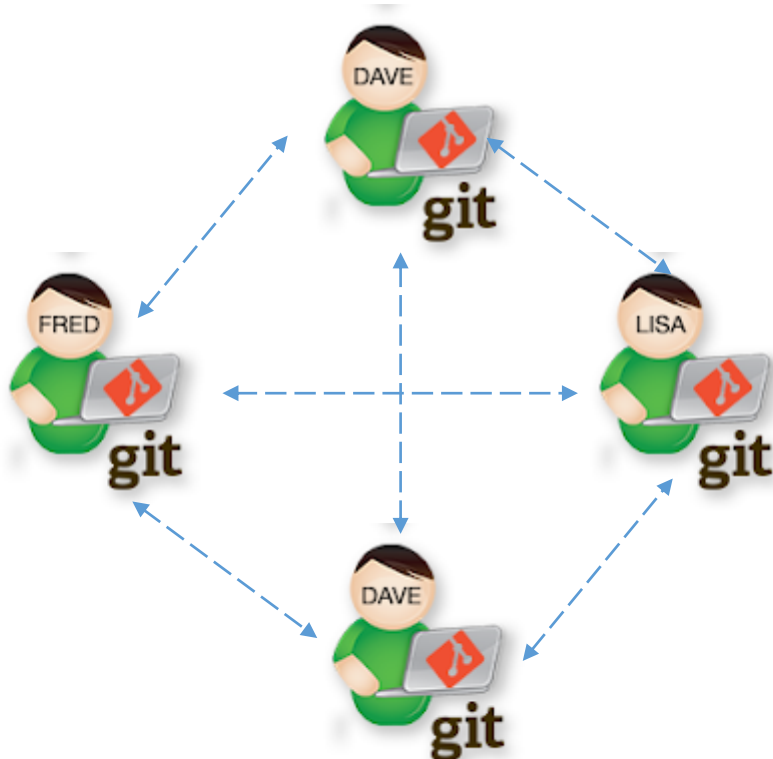




# Características

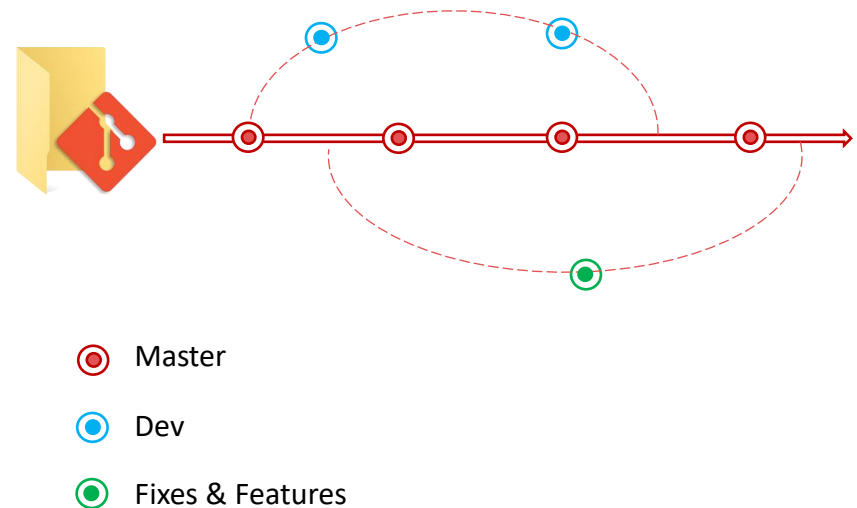
## Distribuido

No es necesario que el proyecto se encuentre centralizado y conectado a internet, ya que pueden trabajar localmente



## Ramas y fusiones (Branch y Merge)

Son ramificaciones que se crean para no comprometer la rama principal



# Características

## Integridad de Datos

Git garantiza que cada uno de los integrantes del proyecto tenga la misma información

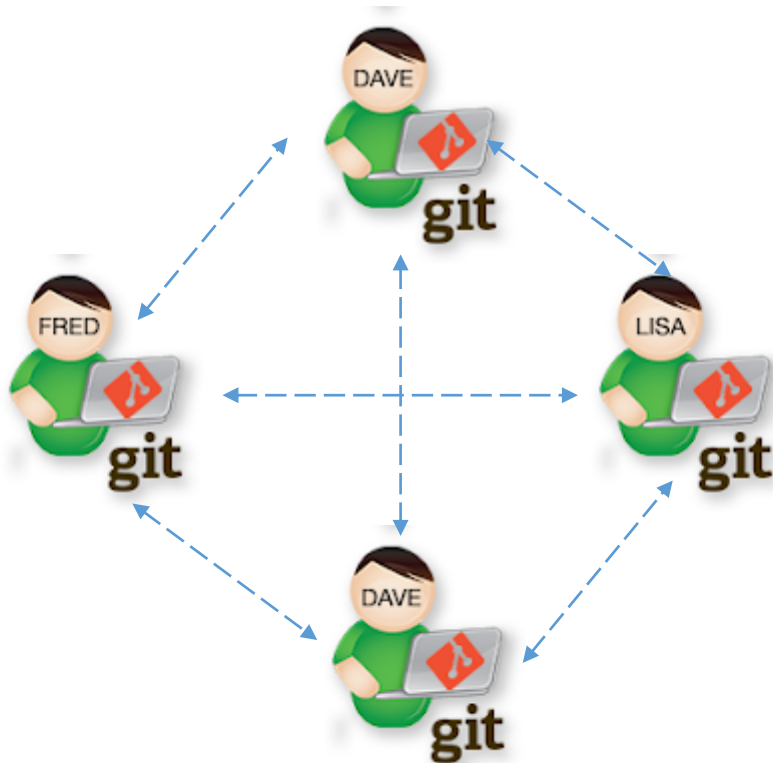
**La integridad de datos evita:**



# Características

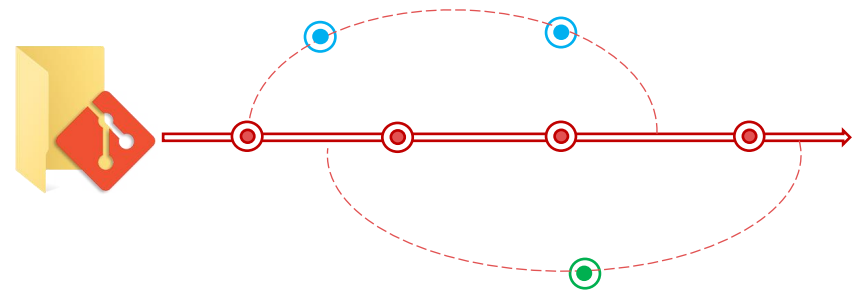
## Auditoria

Permite verificar el historial de cambios de código



## Ramas y fusiones (Branch y Merge)

Es una copia exacta del proyecto



# Comandos GIT



## Comandos básicos

```
// Conocer la versión de git instalada  
git version
```

js

```
// Ayuda sobre los comandos  
git help
```

js

```
// Iniciar un nuevo repositorio  
// Crear la carpeta oculta .git  
git init
```

js

```
// Ver que archivos no han sido registrados  
git status
```

js

```
// Agregar todos los archivos para que esté pendiente de los cambios  
git add .
```

js

```
// Crear commit (fotografía del proyecto en ese momento)  
git commit -m "primer commit"
```

js

```
// Muestra la lista de commit del mas reciente al más antiguo  
git log
```

js

## Ramas o Branch

```
// Crea una nueva rama  
git branch nombreRama
```

js

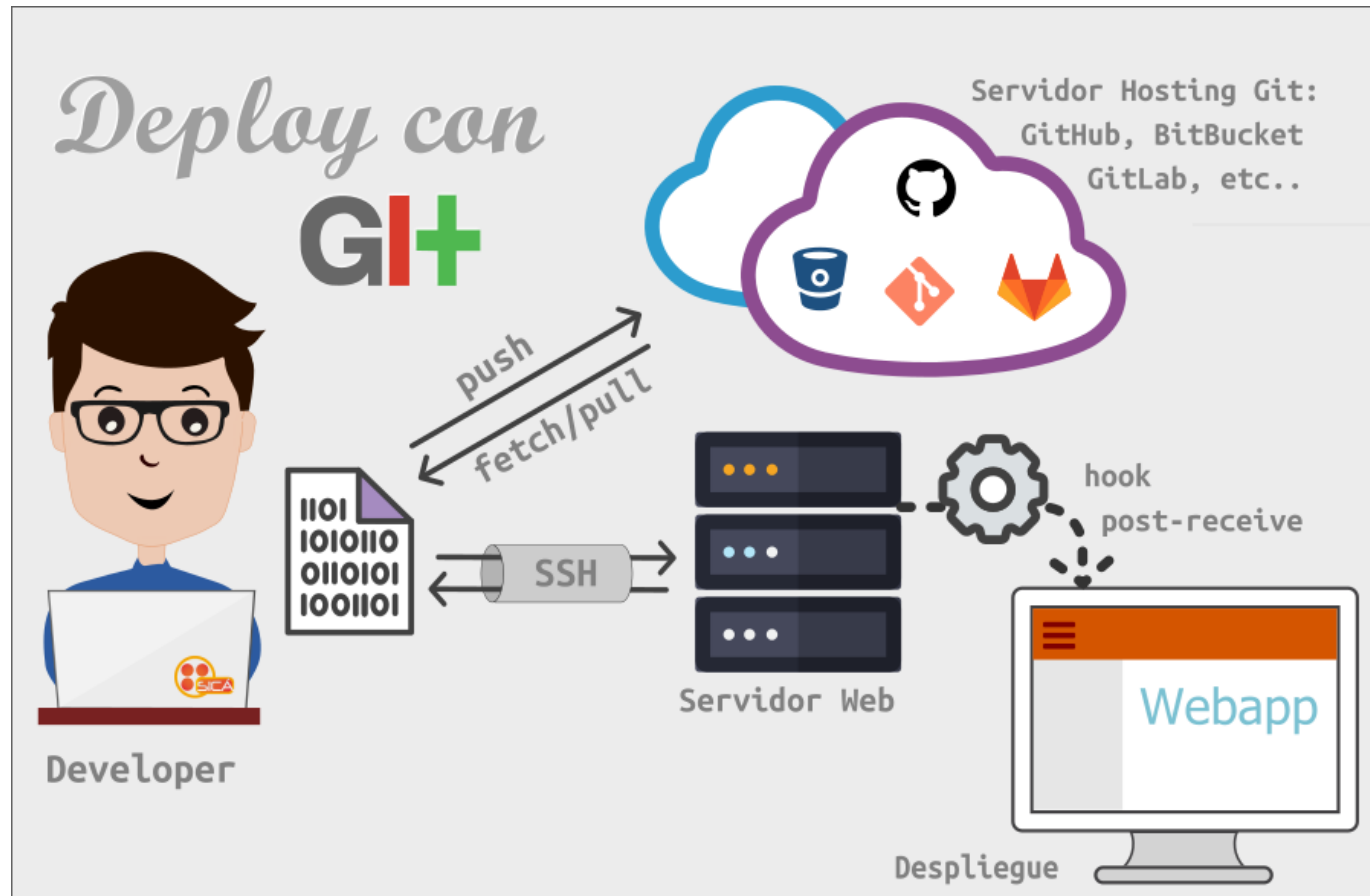
```
// Nos muestra en que rama estamos  
git branch
```

js

```
// Nos movemos a la nueva rama  
git checkout nombreRama
```

js


# Versionamiento en la nube



# GIT





Enlace de descarga <https://git-scm.com/>

 **git** --local-branching-on-the-cheap

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](#).




Latest source Release  
**2.27.0**  
[Release Notes \(2020-06-01\)](#)  
[Download 2.27.0 for Windows](#)

 **Windows GUIs**

 **Tarballs**

 **Mac Build**

 **Source Code**

# Herramientas GIT



## Terminal de consola

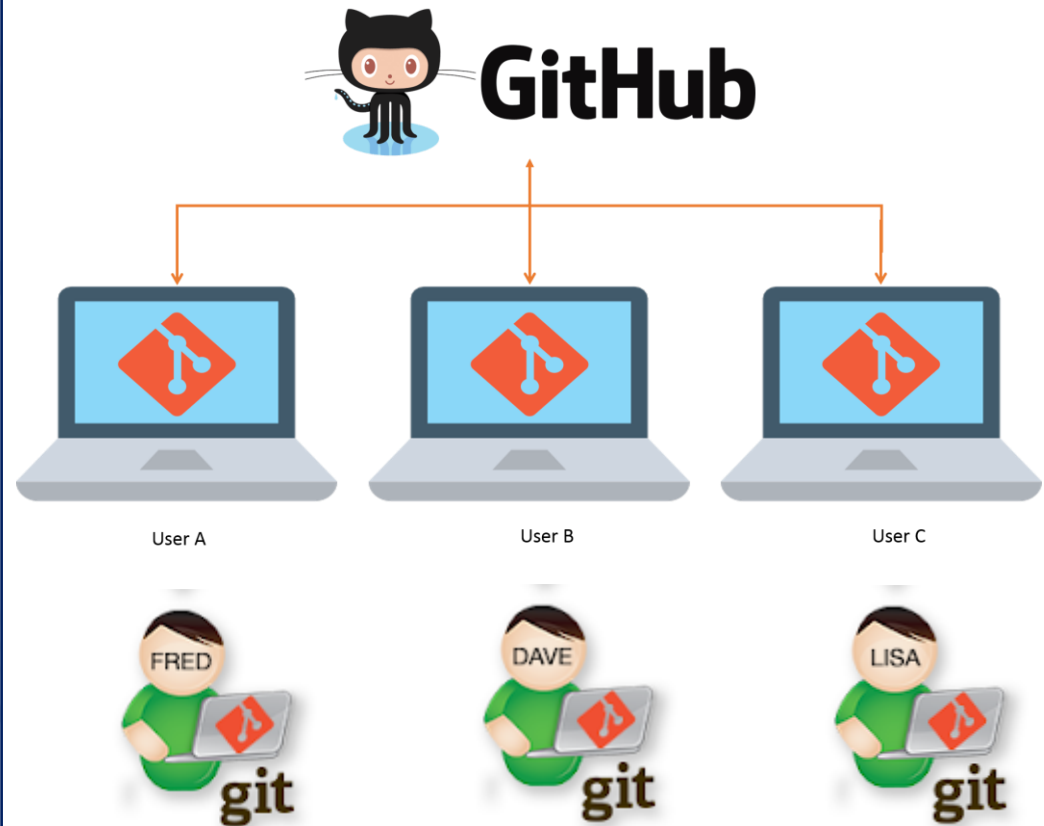
```
MINGW64:/c/Users/Enrique  
Enrique@ MINGW64 ~  
$ |
```



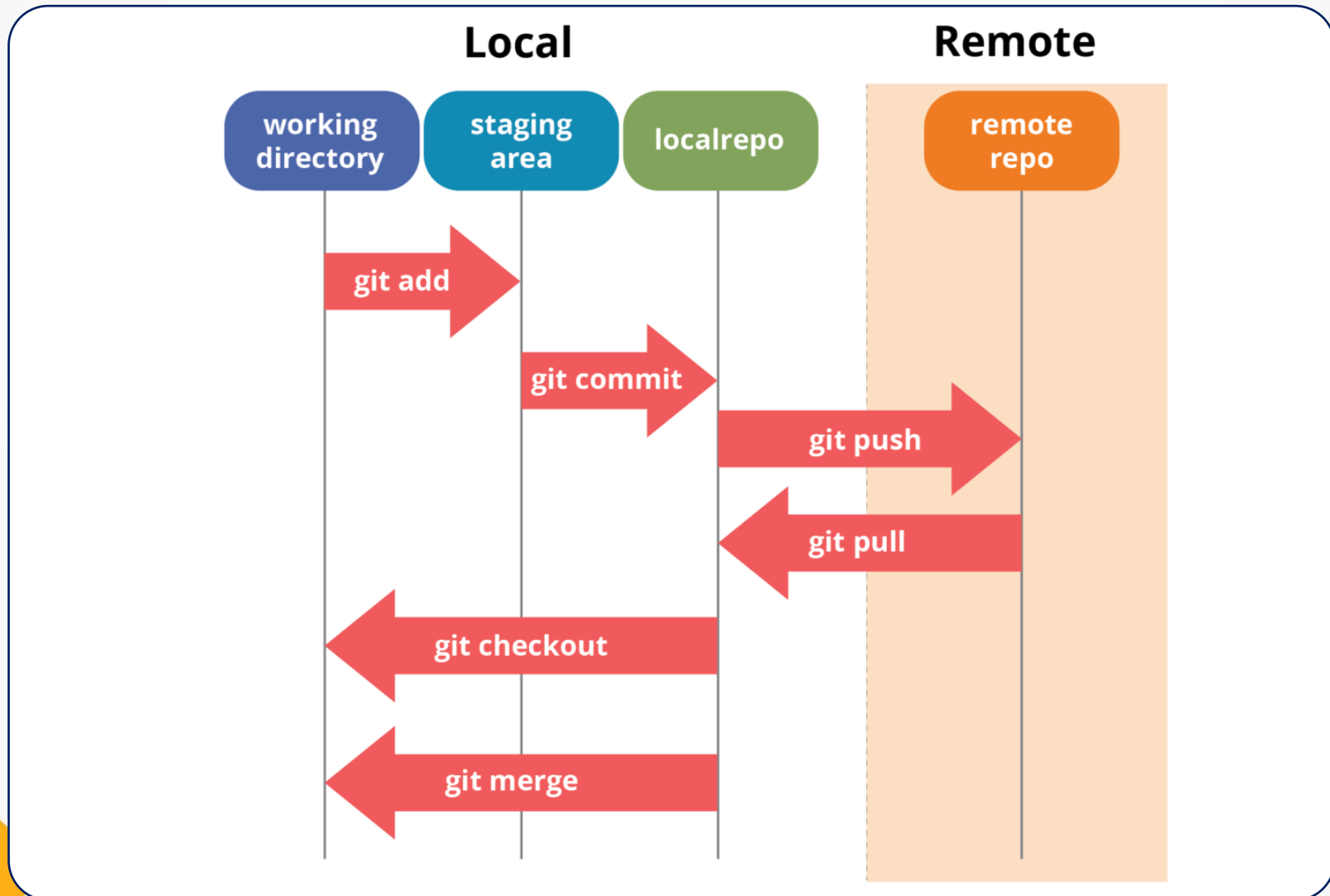


# GitHub

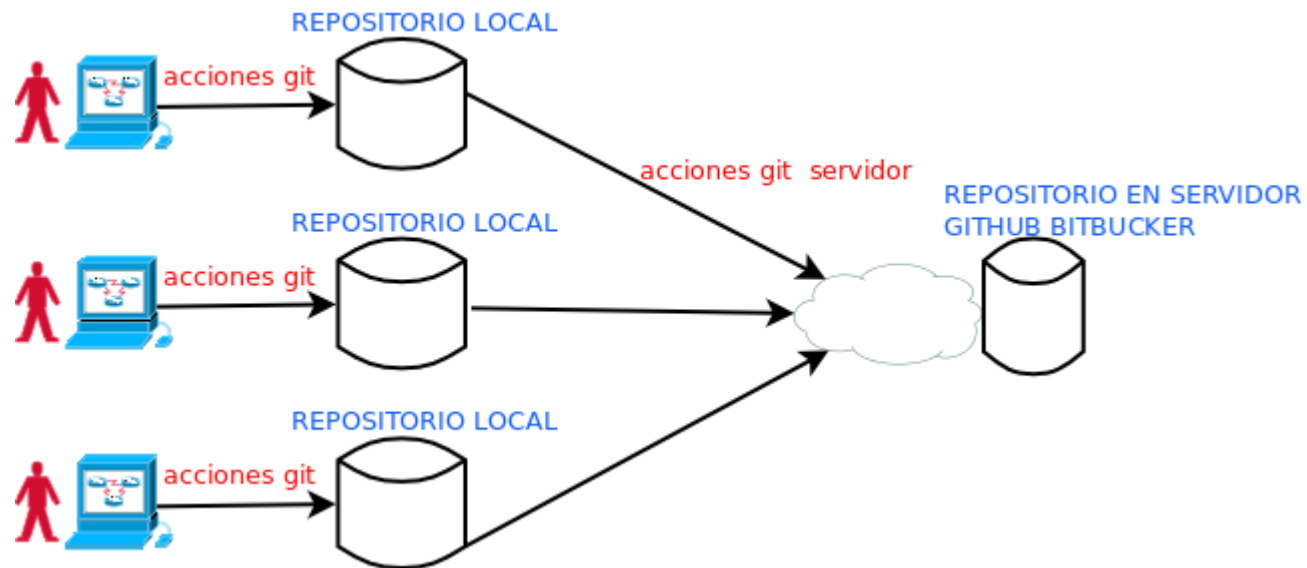
Es una plataforma de desarrollo colaborativo para alojar proyectos (en la nube) utilizando el sistema de control de versiones Git, Además cuenta con una herramienta muy útil que es GitHub Pages donde podemos publicar nuestros proyectos estáticos (HTML, CSS y JS) gratis.



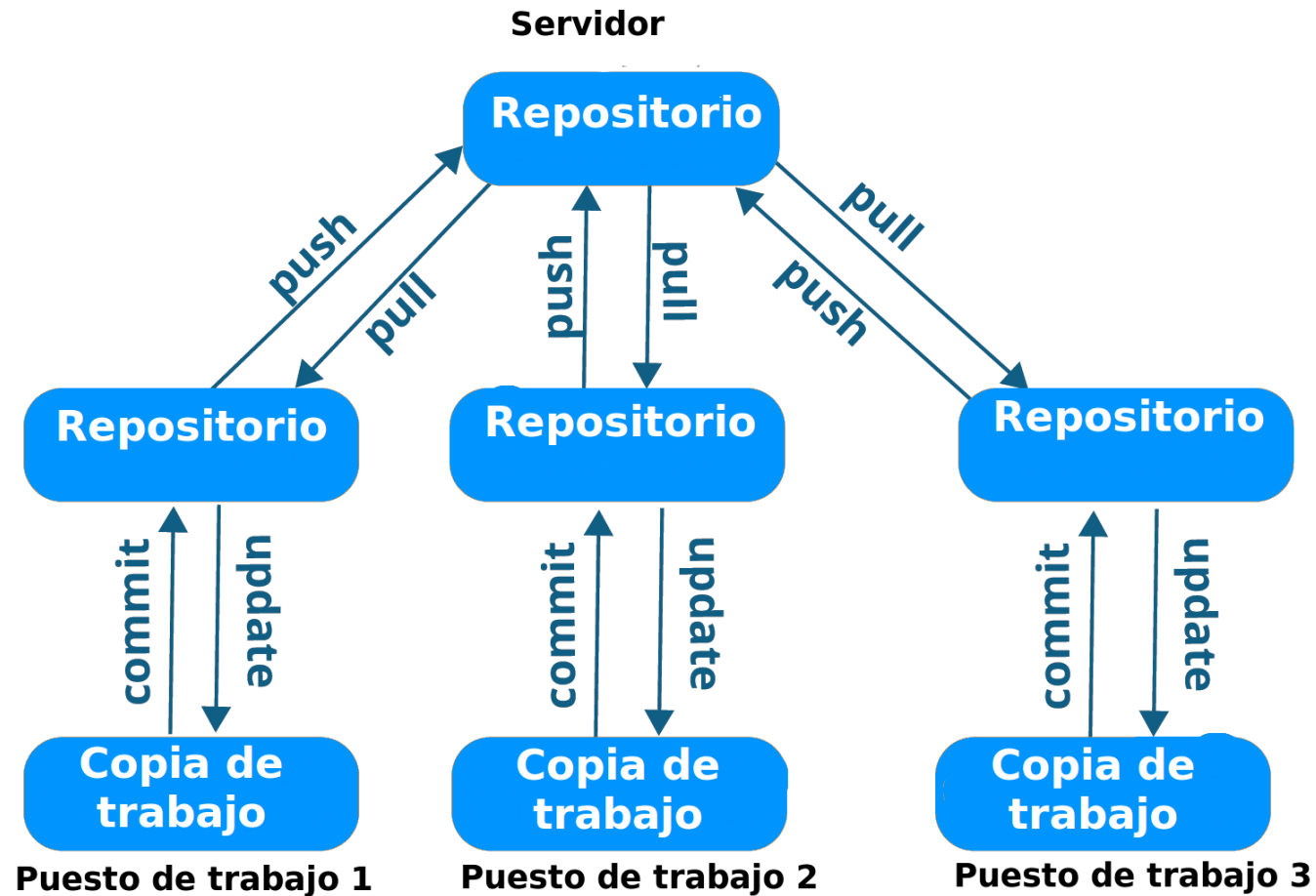
# Versionamiento en la nube



# GITHUB



# GITHUB



# GITHUB



Enlace de descarga <https://github.com/>

**Built for developers**

GitHub is a development platform inspired by the way you work. From open source to business, you can host and review code, manage projects, and build software alongside 50 million developers.

Username

Email

Password

Make sure it's at least 15 characters OR at least 8 characters including a number and a lowercase letter. [Learn more.](#)

**Sign up for GitHub**

By clicking "Sign up for GitHub", you agree to our [Terms of Service](#) and [Privacy Statement](#). We'll occasionally send you account related emails.

# GIT VS GITHUB



# GIT - GITHUB



# git

Repo Local



# GitHub

Repo Remoto

**Gracias**

