

Git y GitHub

DEV.F
DESARROLLAMOS(PERSONAS);

dev



Git es un Sistema de Control de Versiones

Sistema de control de versiones (VCS)

Un **sistema de control de versiones (VCS - Version control system)** es aquel que nos permite llevar un historial y control de cambios a medida que las personas y los equipos colaboran en proyectos juntos.

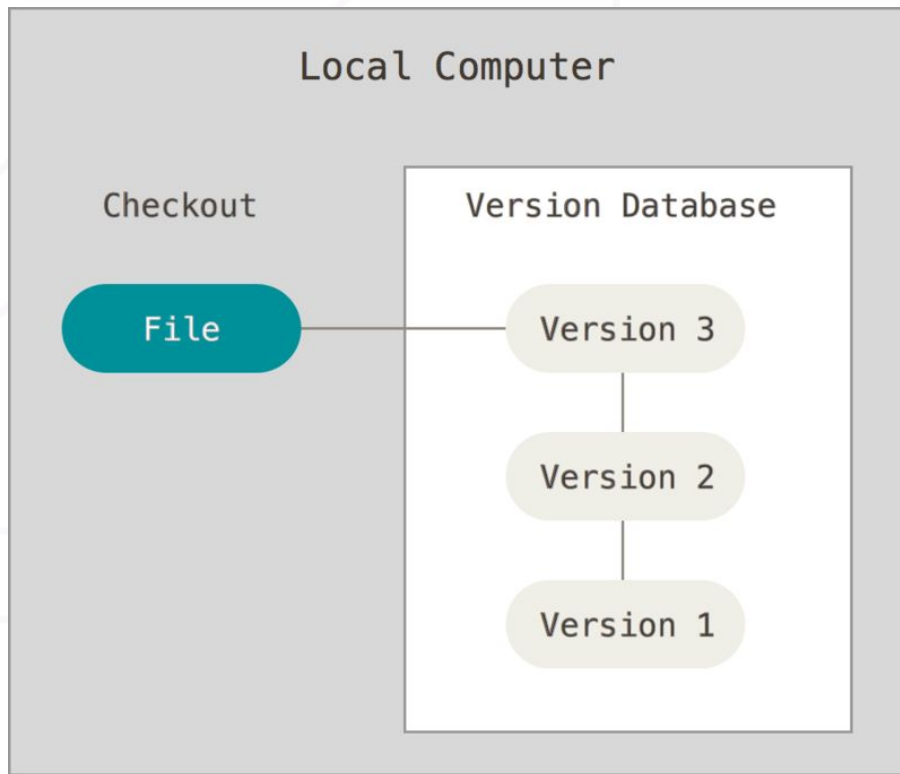
¿Que cambios se hicieron?

¿Quién hizo los cambios?

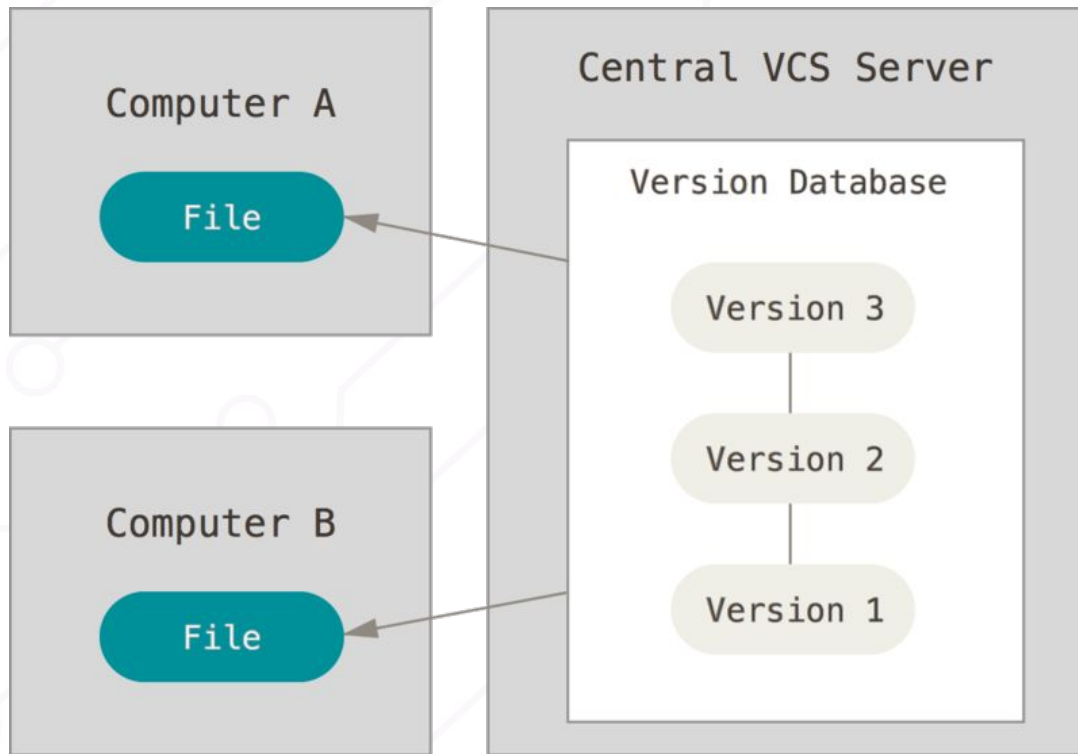
¿Cuando se hicieron los cambios?

¿Por qué fueron requeridos los cambios?

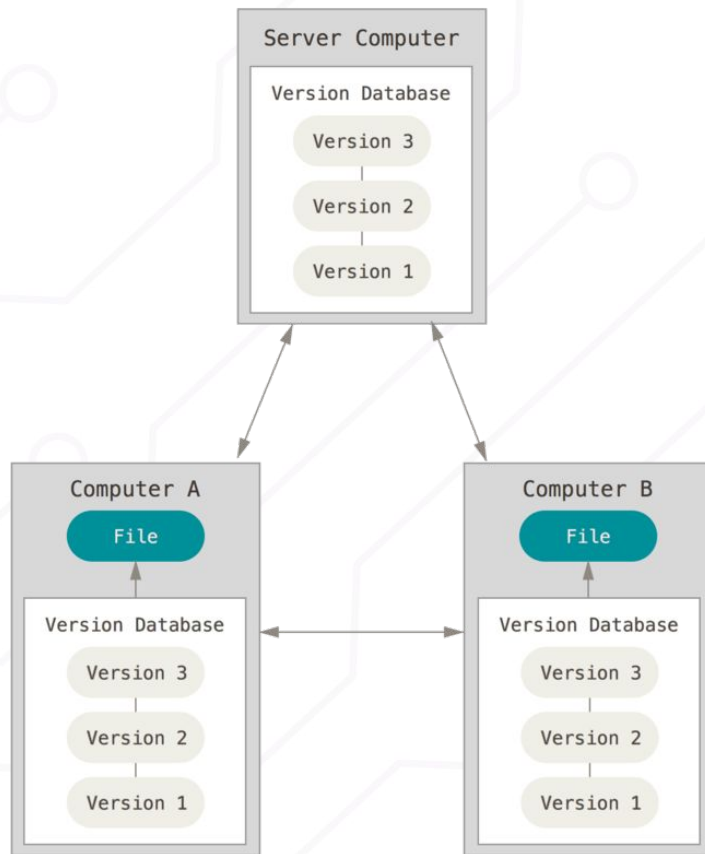
Tipos de VCS (Local)



Tipos de VCS (Centralizado)



Tipos de VCS (Distribuido)





Git es un (VCS) de tipo **distribuido** de código abierto y actualmente el más usado por los desarrolladores gracias a sus beneficios para individuos y equipos de trabajo como:

- Acceso detallado a la historia del proyecto
- Colaboración en cualquier momento y lugar

Su uso principal es mediante Interfaz de línea de comandos (**CLI - Command line interface**)

¿Qué es un repositorio?

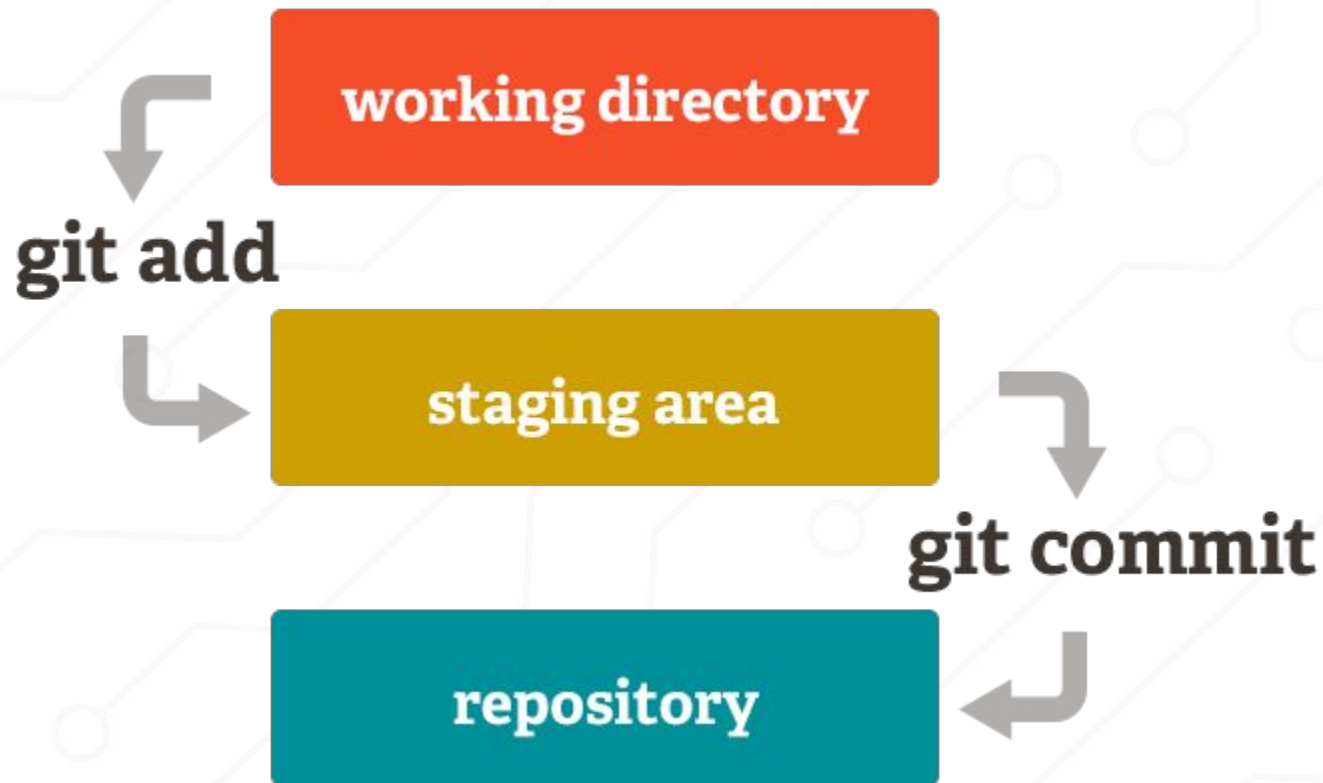


Un **repositorio** es un espacio de almacenamiento donde se organiza, mantiene y difunde información.

En el caso de **Git** el **repositorio** viene siendo el proyecto donde estará la colección de archivos y carpetas junto al historial de cambios.



Estados de Git





Pasos para tomar una Foto en la Playa

Paso 1: Decidir ir a la playa y estar ahí.

Paso 2: Con una cámara, realizar un encuadre de los elementos que queremos que aparezca en la foto.

Paso 3: Presionar el botón de tomar fotografía.



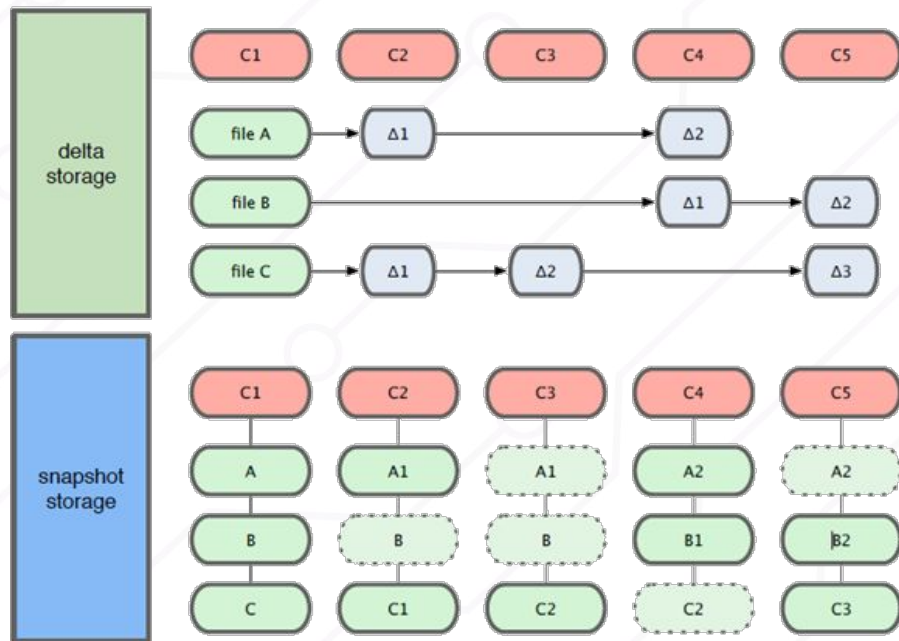
Pasos para tomar una Foto en la Playa

Paso 1: Decidir ir a la playa y estar ahí
working directory (git init / git clone)

Paso 2: Con una cámara, realizar un encuadre de los elementos que queremos que aparezca en la foto.
staging area (git add)

Paso 3: Presionar el botón de tomar fotografía.
repository (git commit)

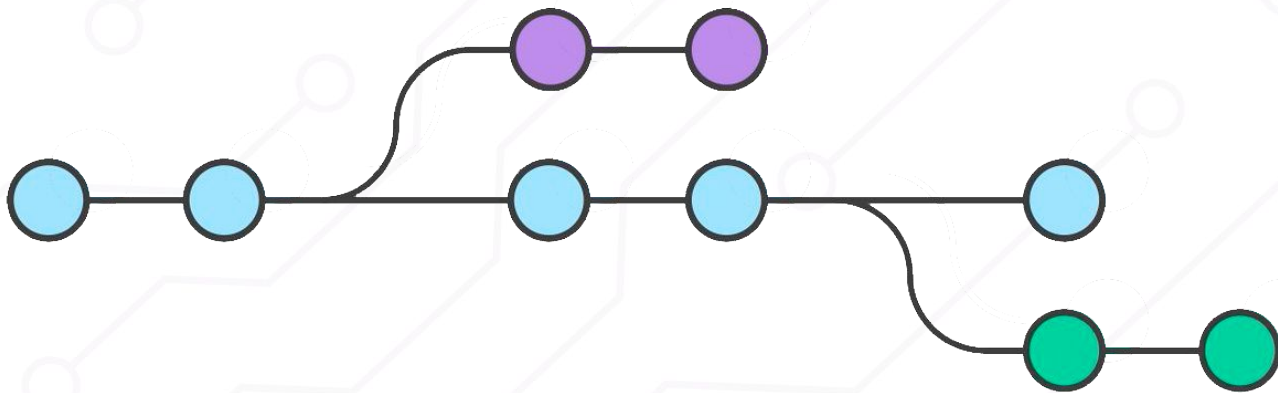
Instantáneas (Snapshots)



Un **Snapshot** es un estado de 'algo' en el tiempo, **Git** maneja **Snapshots** no cambios.

Ramas (Branch)

Las **ramas (Branch)** son bifurcaciones o variantes de un repositorio, estas pueden contener diferentes archivos y carpetas o tener todo igual excepto por algunas líneas de código.



Instalación y manejo de Git

DEV.F
DESARROLLAMOS(PERSONAS);

dev



git

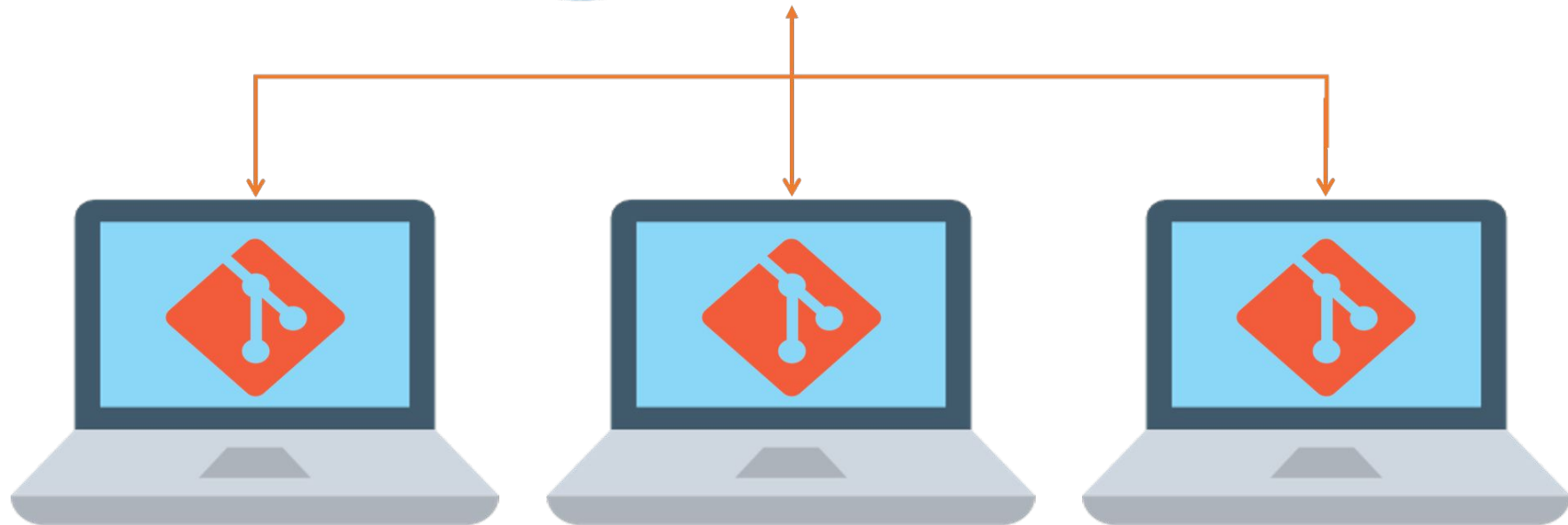


DEV.F





GitHub



User A

User B

User C



pokemon



Pull requests

Issues

Marketplace

Explore



Repositories

29K

Code

1M

Commits

410K

Issues

45K

Packages

0

Marketplace

0

Topics

128

Wikis

4K

Users

2K

Languages

JavaScript

6,935

Java

3,456

Python

2,436

HTML

2,223

C#


1,235

29,727 repository results

Sort: Best match ▾

AHAAAAAAA/PokemonGo-Map

★ 7.8k

 Live visualization of all the *pokemon* in your area... and more! (shutdown)

Updated on Jan 26, 2017

PokemonGoF/PokemonGo-Bot

● Python

★ 3.6k

The *Pokemon* Go Bot, baking with community.

pokemongo-bot

pokestops

pokemongof

MIT license

Updated on May 10

5 issues need help

Zarel/Pokemon-Showdown

● JavaScript

★ 2.2k

Pokémon battle simulator.

MIT license

Updated 10 hours ago

2 issues need help



GitHub

DEV.F

Configuración de Git

- Desde consola, se puede acceder a la configuración de **Git** con el comando:

```
git config
```

- Se recomienda establecer una identidad en **Git**, para ello se usan los comandos:

```
git config user.name
```

```
git config user.email
```

- Usando el flag “**--global**” podemos establecer la configuración de forma global y realizarla una sola vez.

```
git config --global user.name “devf”
```



Comandos básicos de Git

git init -> Inicializamos repositorio

git clone (url) -> Clonar repositorio existente

git status -> Nos muestra un desglose los archivos agregados y modificados

git add (nombre archivo) -> Agregamos archivos al staging area

git commit -m "Comentario" -> Agregamos comentario sobre el cambio hecho

git push -> Enviamos cambios a repositorio remoto

git checkout -b (nombre) -> Salimos de la rama actual creando una nueva

git checkout (nombre) -> Salimos de la rama actual a una existente

git merge (rama) -> Unimos cambios de una rama

git pull -> Obtenemos cambios más recientes de la rama



**I HAVE NO
IDEA WHAT
I'M DOING**