



# Git

---

LUIS FERNANDO GOMEZ ALEJANDRE



Mi Experiencia con Git Hub

# ¿Qué es el control de versiones, y por qué debería importarte?

---



*El control de versiones es un sistema que registra los cambios realizados sobre un archivo o conjunto de archivos a lo largo del tiempo, de modo que puedas recuperar versiones específicas más adelante.*



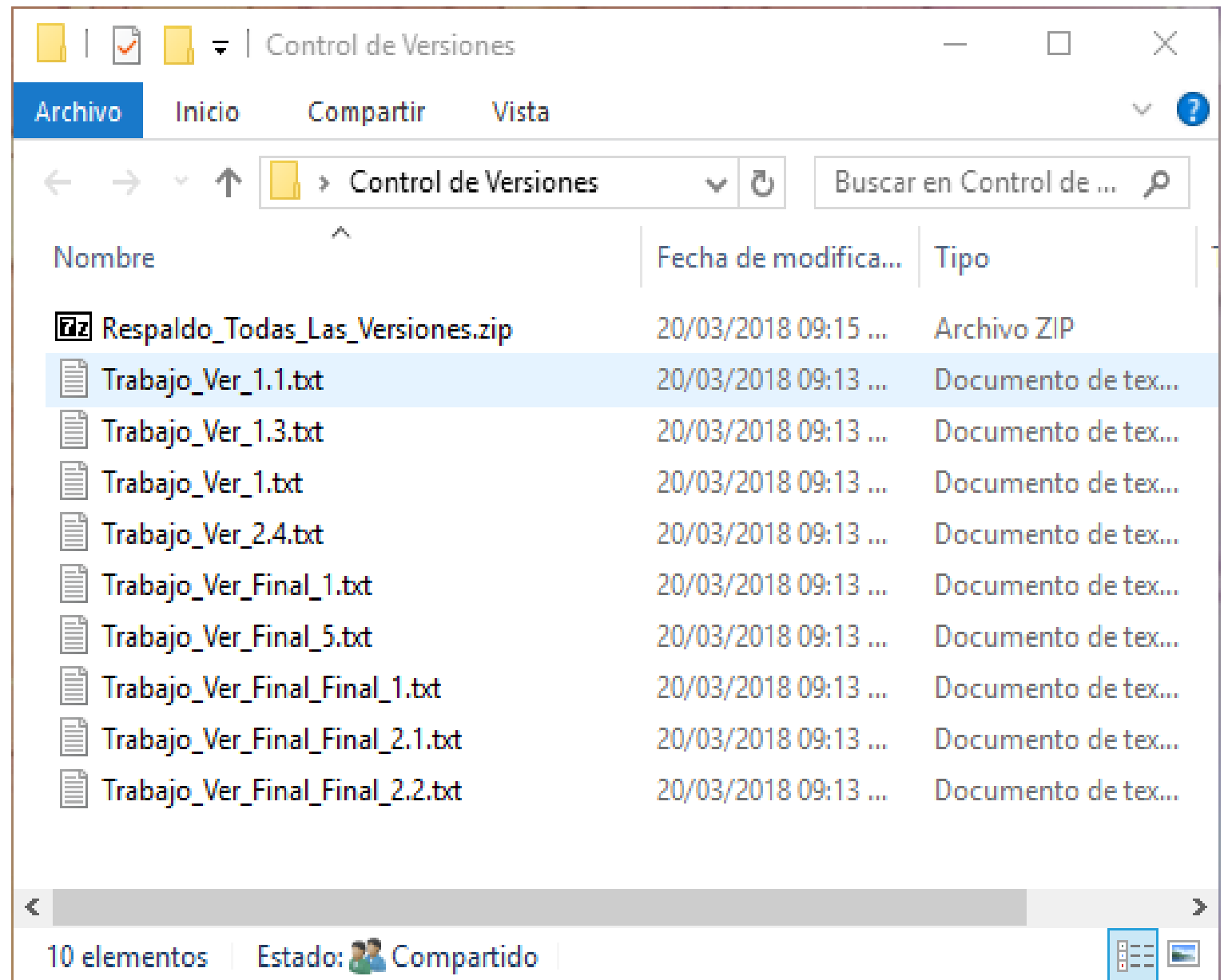
# ¿Qué puedes hacer con un VCS?

---

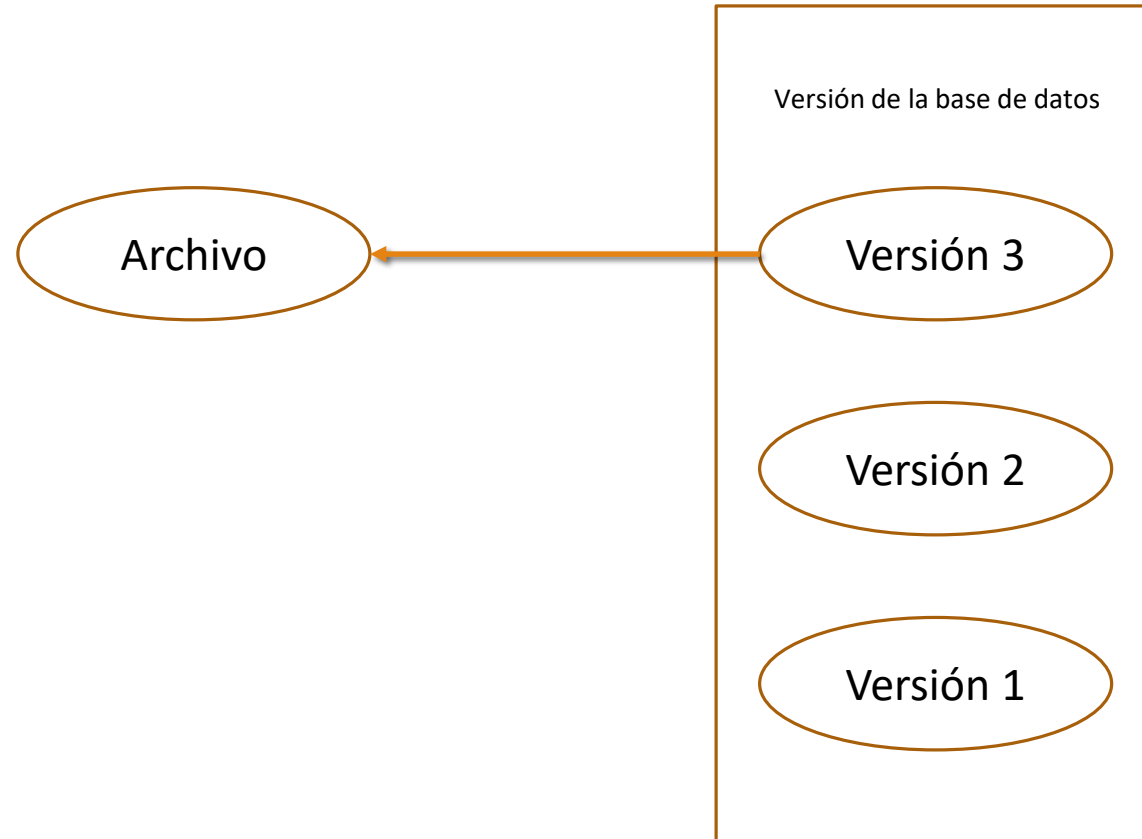
Te permite **revertir** archivos a un estado anterior, revertir el proyecto entero a un estado anterior, **comparar** cambios a lo largo del tiempo, ver quién **modificó por última vez** algo que puede estar causando un **problema**.

# ¿Coincidencia?

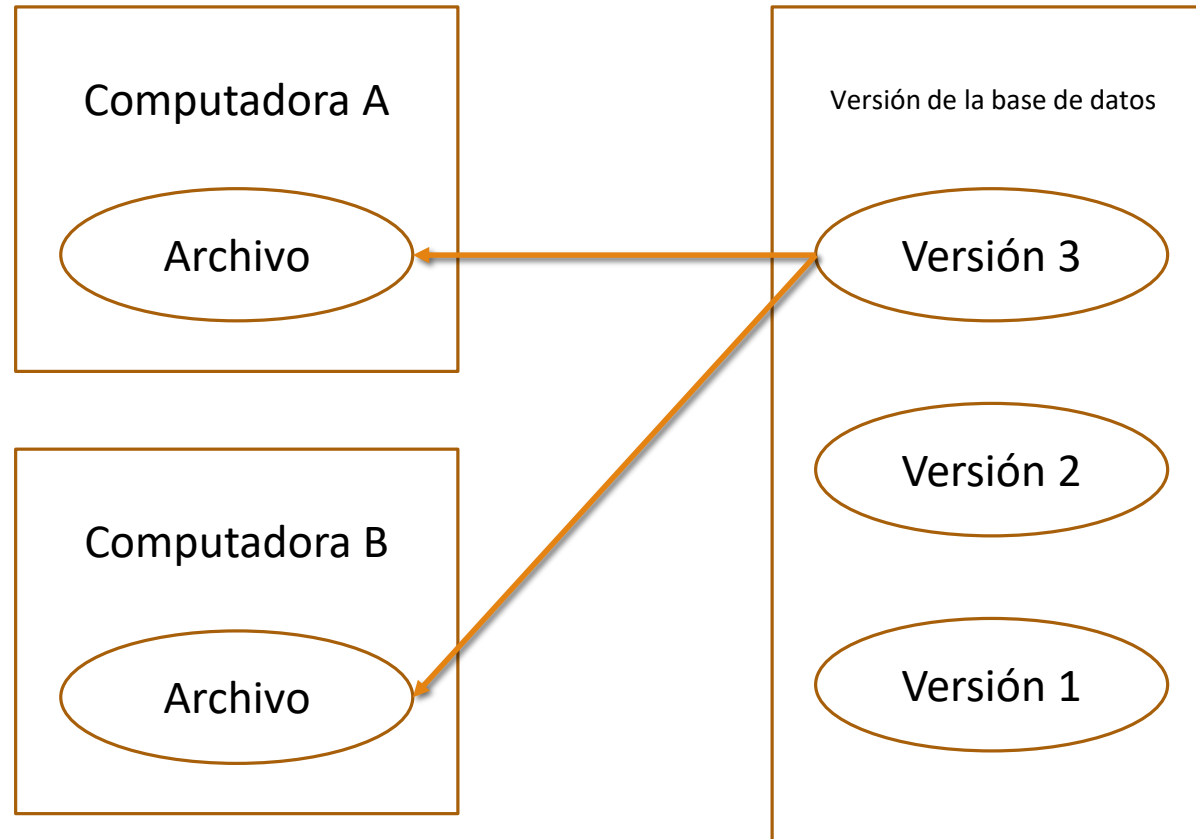
Cualquier parecido con la realidad es por que no conoce que es un control de versiones 😊

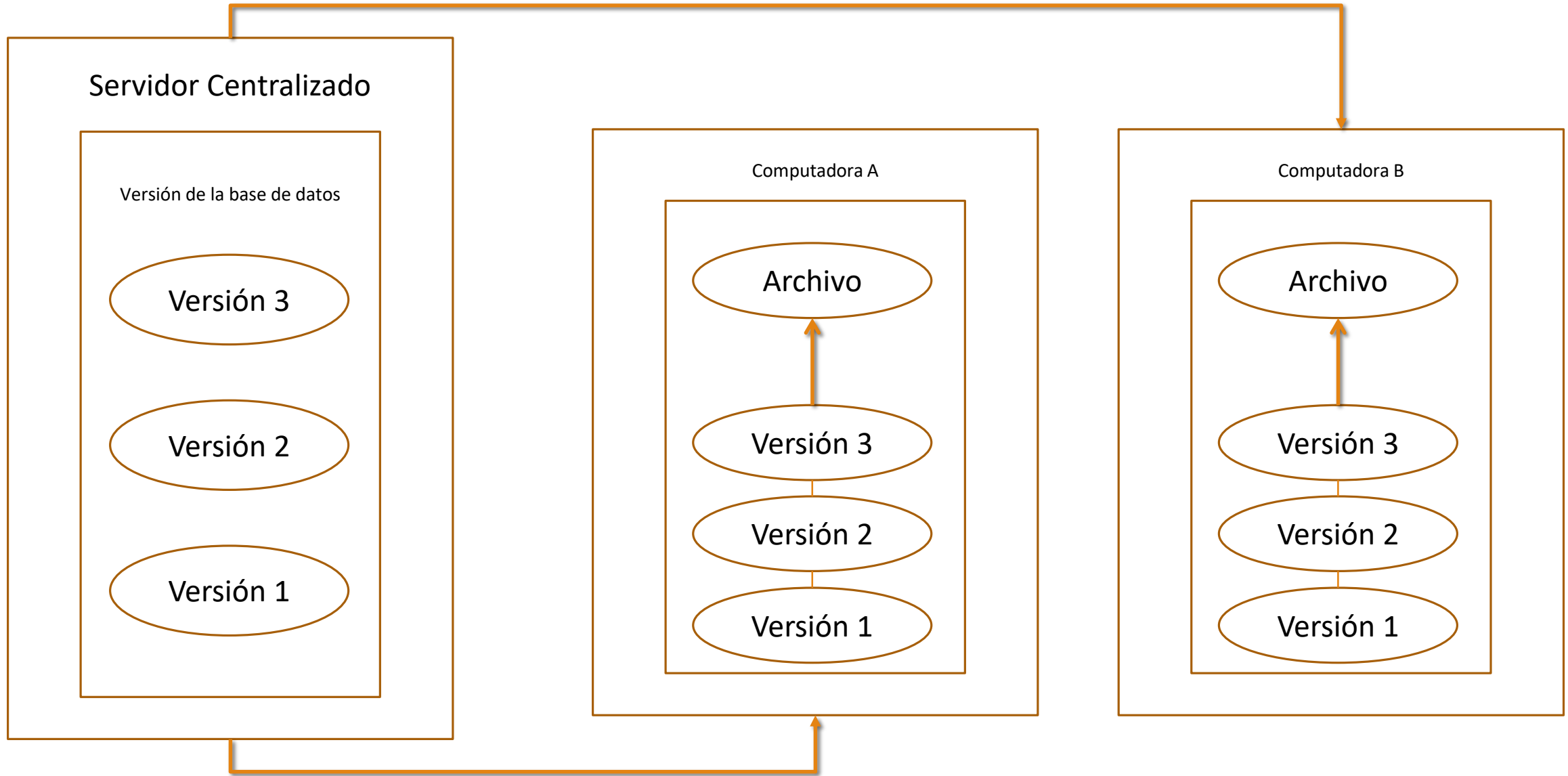


## Computadora Local



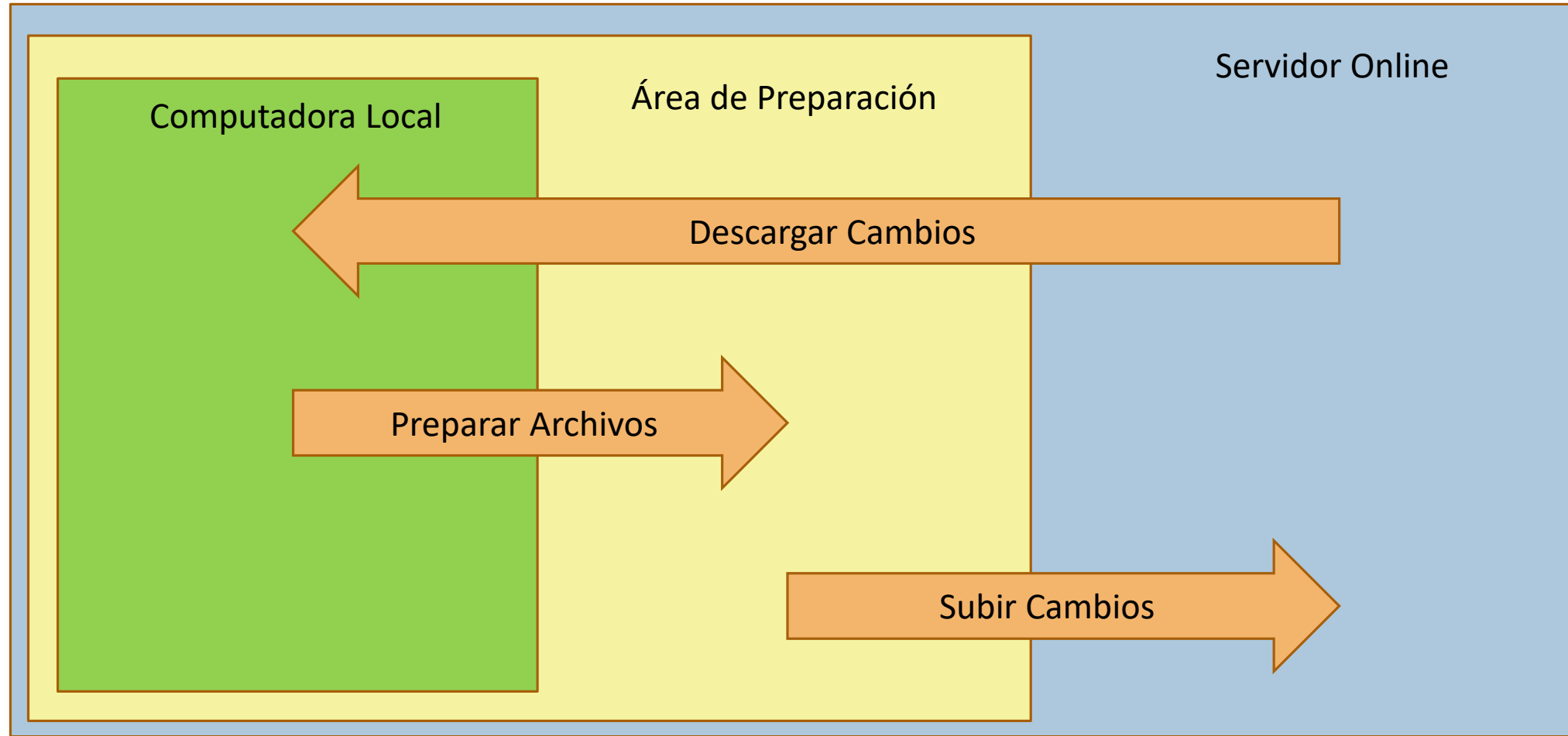
## Servidor Centralizado







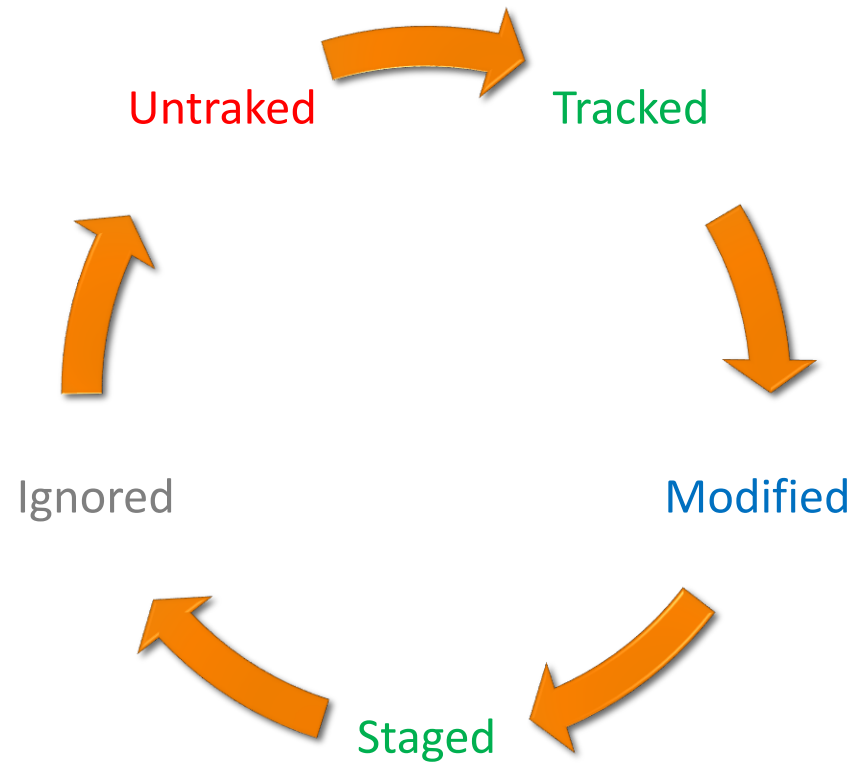
# Repositorio





# Estado de los Archivos

---





# Glosario

---

Branch (Rama)

---

Repository (Repositorio)

---

Clone (Clonar)

---

Fetch (Revisar)

---

Pull(Incorporar)

---

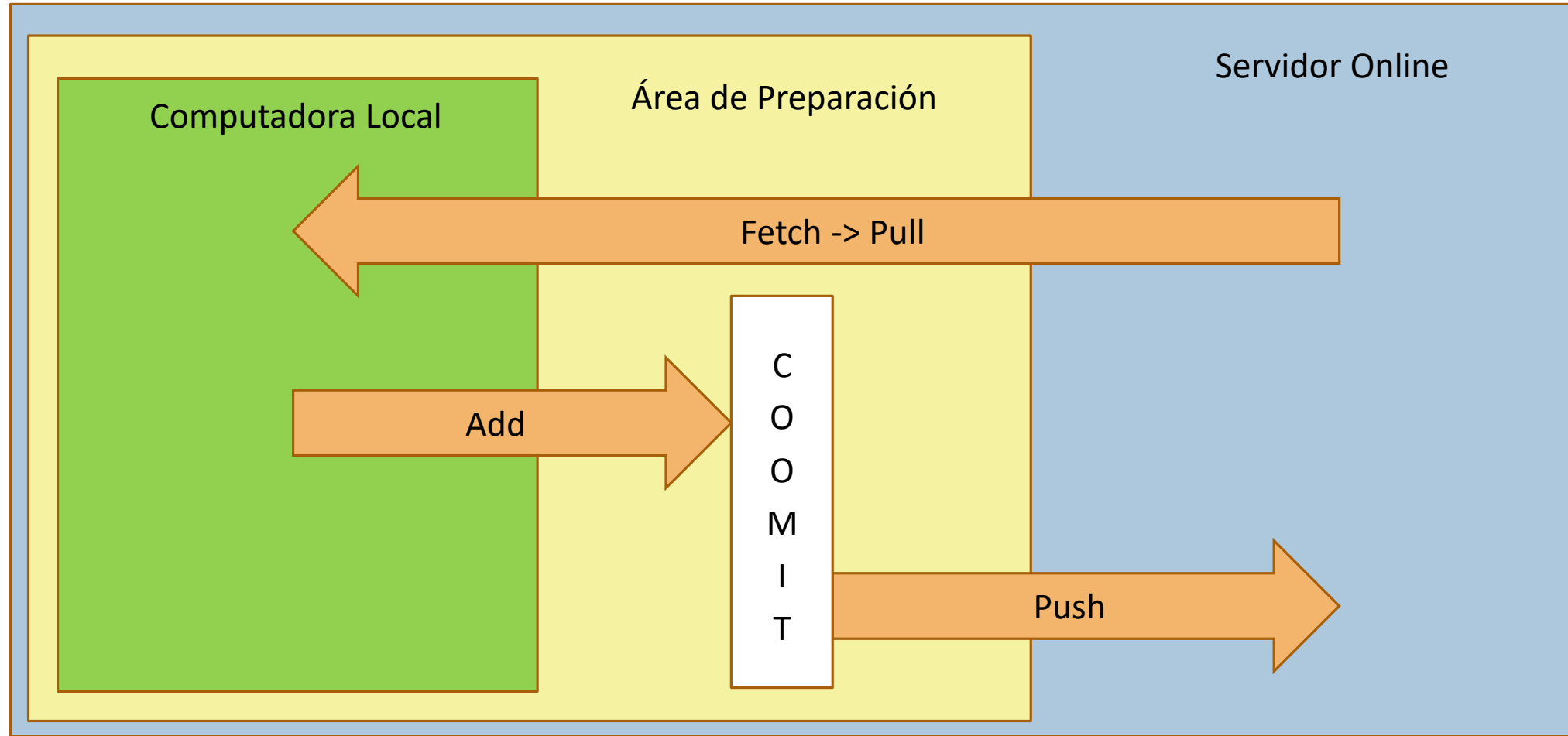
Push (Descargar)

---

Reset (Reiniciar)

---

# Repositorio Comandos





# Creando un repositorio

---



# Comandos for 1<sup>st</sup> Repo 😊

---

1. `curl -u "user:password" https://api.github.com/user/repos -d "{\"name\":\"repoName\"}"`
2. `echo "Contenido del README" >> README.md`
3. `git init`
4. `git add README.md`
5. `git commit -m "Comentario del primer commit"`
6. `git remote add origin https://github.com/user:repoName.git`
7. `git push -u origin master`

:repoName      Nombre del repositorio donde se va a guardar



# Mandar Invitación

---

```
curl -i -u "user:password" -X PUT -d "  
"https://api.github.com/repos/:owner/:repoName/collaborators/:profileToInvite"
```

:owner	Propietario del repositorio a invitar
:repoName	Nombre del repositorio al que se va a agregar
:profileToInvite	Nombre de usuario del colaborador

Regresa un ID de invitación...

```
C:\Users\DARKE\Desktop\Example>curl -i -u "DuckHunter213:[REDACTED]" -X PUT "https://api.github.com/repos/DuckHunter213/Example/collaborators/WoodPecker213"
HTTP/1.1 201 Created
Server: GitHub.com
Date: Wed, 25 Apr 2018 02:03:16 GMT
Content-Type: application/json; charset=utf-8
Content-Length: 6951
Status: 201 Created
X-RateLimit-Limit: 5000
X-RateLimit-Remaining: 4997
X-RateLimit-Reset: 1524625268
Cache-Control: private, max-age=60, s-maxage=60
Vary: Accept, Authorization, Cookie, X-GitHub-OTP
ETag: "f4dec948a344f86ce0f805751a8b1a60"
Location: https://api.github.com/user/repository_invitations/8883972
X-GitHub-Media-Type: github.v3; format=json
Access-Control-Expose-Headers: ETag, Link, Retry-After, X-GitHub-OTP, X-RateLimit-Limit, X-RateLimit-Remaining, X-RateLimit-Reset, X-OAuth-Scopes, X-Accepted-OAuth-Scopes, X-Poll-Interval
Access-Control-Allow-Origin: *
Strict-Transport-Security: max-age=31536000; includeSubdomains; preload
X-Frame-Options: deny
X-Content-Type-Options: nosniff
X-XSS-Protection: 1; mode=block
Referrer-Policy: origin-when-cross-origin, strict-origin-when-cross-origin
Content-Security-Policy: default-src 'none'
X-Runtime-rack: 0.113986
X-GitHub-Request-Id: E3F4:617E:4AEF58:B6CB65:5ADFE1E4

{
  "id": 8883972,
  "repository": {
    "id": 130931483,
    "name": "Example",
    "full_name": "DuckHunter213/Example",
    "owner": {
      "login": "DuckHunter213",
```





# Aceptar Invitación

---

```
curl -i -u "user:password" -X PATCH https://api.github.com/user/repository_invitations/:invID
```

:invID id de la invitación que tu compañero te mando



# Colaborar en el Proyecto

---

git clone https://github.com/:owner/:repoName

:owner            Propietario del repositorio

:repoName        Nombre del repositorio

\* Nota (No se necesita ser colaborador para poder clonar los proyectos).



# Agregar archivos en el repositorio

---

1. `echo "Hola mundo" >> holaMundo.txt`
  1. crear un archivo en la carpeta del repositorio
2. `git status` \*Solo para visualizar los documentos que cambiaron es opcional
3. `git add .`
  1. `git add :fullPathArchivo`
4. `git commit -m "Comentario del commit"`
5. `git push`



# Recibir los cambios

---

1. git push
  1. Si existen conflictos se necesita decidir por los archivos.
  2. Se puede ocupar una herramienta como un *git diff :fileName*
2. Se puede hacer un **MERGE** o un **FAST FORWARD**.
3. Evita hacer a toda costa las acciones **REBASE** Suelen ser **acciones destructivas** que terminan por borrar tus archivos del repositorio.



# Referencias

---

## Inicio de ramas en Git

- <https://www.youtube.com/watch?v=rmO7t35l1XI>

## Manual de Git

- <https://git-scm.com/book/es/v1/Empezando-Acerca-del-control-de-versiones>

## Ayuda de GitKraken

- <https://support.gitkraken.com/>

## API REST GIT

- <https://developer.github.com/v3/>