

joselatrans / tallerGit Private

Branch: #1 tallerGit / tallerGitUNA.md

Find file Copy

joselatrans github

d3936de 5 days

1 contributor

184 lines (123 sloc) 4.87 KB



## ¿Qué es Git?

- Es un sistema de **control de versiones**
- Creado por Linus Torvalds durante la creación de Linux
- Software libre

## ¿Para qué?

- Trabajos colaborativos o que deben de ser revisados muchas veces
  - Tesis
  - Artículos
  - Proyectos finales de cursos
- Respalidar archivos importantes

## ¿Cómo usarlo?

- [Instalar](#) Git
- Abrir línea de comandos
  - También se puede acceder desde otros programas como RStudio, etc.

- Ir al directorio de trabajo
- Iniciar git, agregar cambios (add), anotar cambios (commit), agregar ramas, push, etc.

## Taller

1. Crear carpeta en el *Escritorio* **tallerGit**
2. Crear archivo **plantasCompartidas.txt**
  - Hacer lista de 4 plantas favoritas
  - Guardar archivo
3. Abrir línea de comandos de Git (Windows) (Linux / Mac abrir la terminal)
  - Ir a la carpeta:

```
"cd direccion/carpeta/"
```

- Escribir:

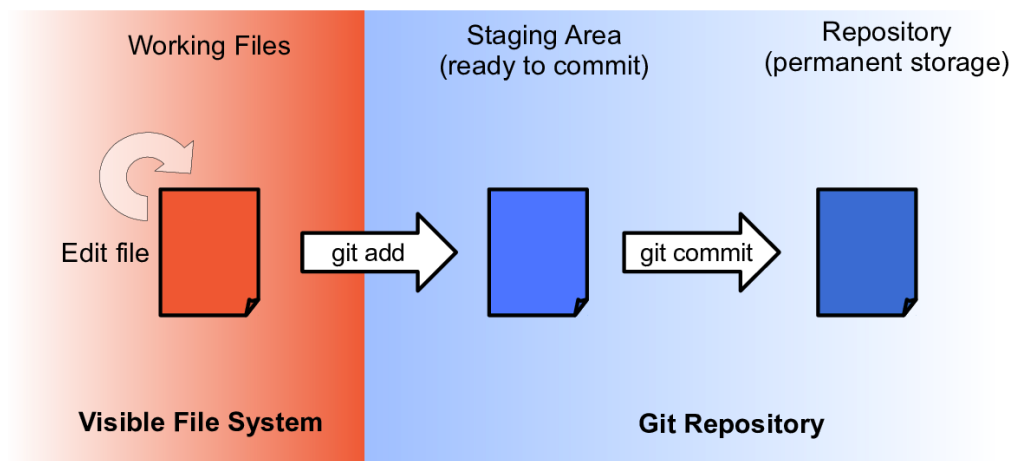
```
git status
git init
git status (de nuevo)
git add <archivo>
git commit -m "un comentario" **ATENCIÓN**
```

4. Editar documento
  - Ver diferencias entre los archivos

```
git diff (--staged)
```

- Commit

```
git add <archivo>
git commit -m "agregué plantas comestibles"
```



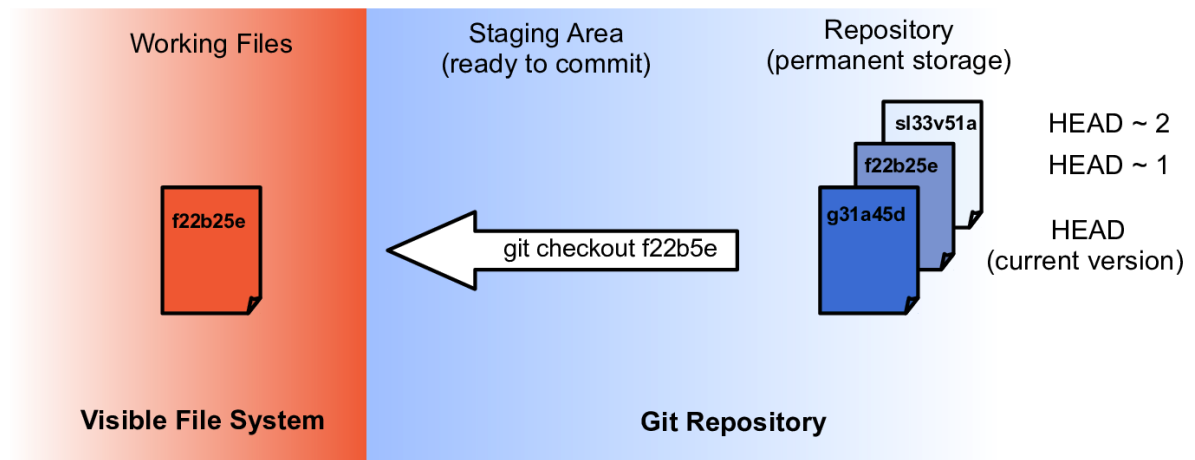
5. Agregar archivo a la carpeta
  - Agregar ítemes al archivo

```
git add <archivo>      ó      < . >
```

- Agreguen el archivo al git
- Revisar historial

```
git log
```

## HEAD



```
git diff HEAD~1 <archivo>
```

```
checkout
```

Checkout es un comando que nos permite volver a la versión que está bien, en caso de que hayamos guardado y "giteado" un error terrible.

## Ignorando cosas

En un git no siempre es conveniente subir todo lo que tenemos en la carpeta.

Git hace una copia de **todos** los archivos en esa carpeta, debemos de mantenerla liviana.

```
.gitignore
```

Es un archivo que sirve para decirle a Git que no queremos que copie

1. Cree algunos archivos
2. Cree un `.gitignore`
3. Ponga el nombre de los archivos en el `.gitignore`
4. Gitee la carpeta

¿Qué pasa si "giteamos" los archivos y luego los metemos en el `.gitignore`?

- Extra:

1. Cree archivos d1.dat, d2.dat, d3.dat
2. Metales en el .gitignore con <\*.dat>
3. Gitee 🙌

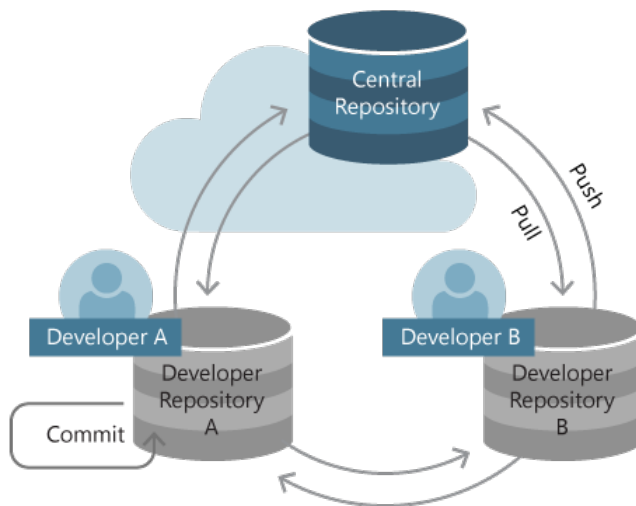
## Colaboración

### Repositorios

**GitHub** es más famoso, pero **Gitlab** es completamente open source y además permite hacer repositorios privados de forma gratuita, para Github hay que pagar o poner el correo institucional (2 años).

Sin embargo...

1. Haga un cuenta en **GitHub**.
  - o Si ya tiene logueese
2. Logueese en su cuenta

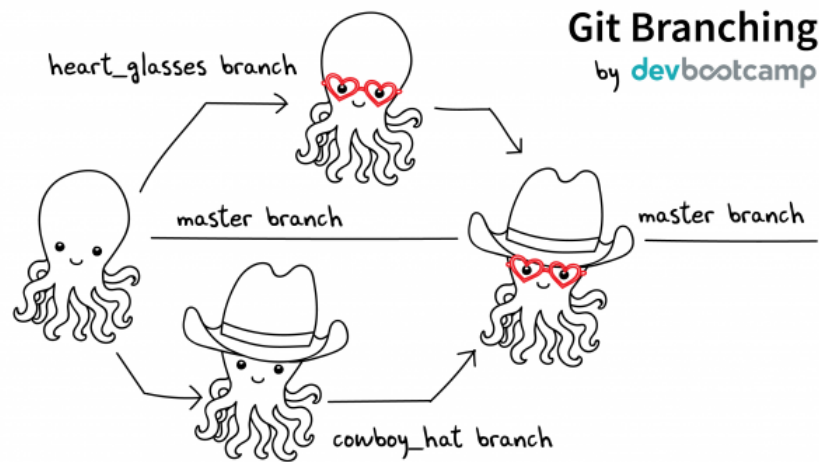


1. Haga un repositorio nuevo en su cuenta.

```
git remote add origin
```

```
git push -u origin master
```

### Ramificaciones



```
git branch
```

```
git remote -v
```

### Trabajo en grupo

1. Haga una carpeta nueva que se llame **gitColaboracion**
2. Vaya a esa carpeta en la línea de comandos de Git

```
git clone
```

3. Añadan a su colaborador

[Agregar colaboradores](#)

### ¿Sin conflictos o con conflictos?

```
git push
git pull
```

### Catdog

Cada vez que un colaborador haga **cambios** y los suba el otro debe de usar **pull** si **no** lo hace van a haber **conflictos**

Lo bueno es que se pueden resolver fácil 🐔

1. Hagan cambios los dos y sin hacer **pull** hagan **push**

En caso de ser necesario:

```
git push --force origin master
```

2. Ahora el "dueño" arregle los cambios y los sube
3. El colaborador haga **pull**
4. Edite el documento
5. Suba los cambios
6. Revisen

# Integración con RStudio

---

Ruta para configuración:

Tools --> Global options... --> Git/SVN

**Se usa para proyectos**

## Ejercicio

1. Cree un proyecto en RStudio de la carpeta *tallerGit* 1.2. Explórelo
2. Cree un proyecto en RStudio 2.1. Hágle un repositorio Git desde RStudio



## Recursos

---

[Cheatsheet](#)

Presentaciones base para el taller

[https://jiffyclub.github.io/2015-07-06-scipy/slides-local\\_version\\_control.html#/5](https://jiffyclub.github.io/2015-07-06-scipy/slides-local_version_control.html#/5)  
[http://slides.com/abostroem/collaborating\\_using\\_git#/](http://slides.com/abostroem/collaborating_using_git#/)  
<http://slides.com/abostroem/deck-4#/>

[Más sobre ramificaciones](#)