Git y GitHub

Mg. Flor Cerdán León

Agenda

- 1. Repaso
- 2. Creando repositorios
- 3. Agregando, quitando y viendo el estatus de archivos
- 4. Confirmando cambios
- 5. Etiquetando confirmaciones

1. Repaso

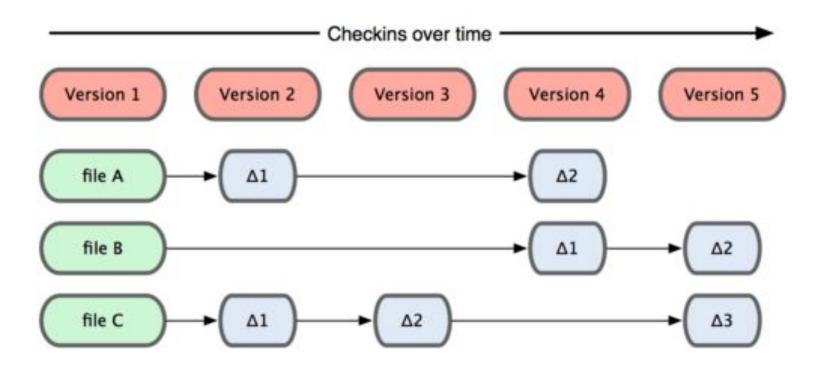
1. ¿Qué es un VCS?

- (Español) Sistema Control de Versiones
- Es un **Sistema** que registra los cambios realizados sobre un archivo o conjunto de archivos a lo largo del tiempo.

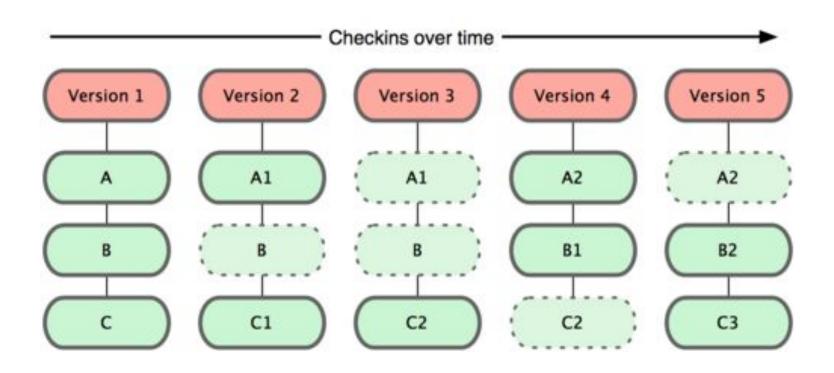


Sistema de Control de Versiones Distribuido

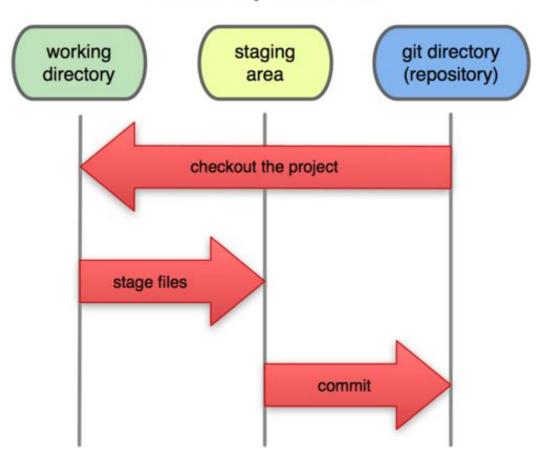
Otros VCS



Git



Local Operations



Comandos útiles de la terminal

- cd que nos permite movernos dentro de carpetas.
 cd .. podemos salir de las carpetas.
- mkdir para crear carpetas
- Is vamos a ver las carpetas que tenemos creadas.
- *clear* para limpiar la terminal.
- touch nos permite crear archivos.
- rm para borrar archivos.
- rm -rf y el nombre de la carpeta para borrar carpetas

VIM (Editor de Archivos)

- i para poder insertar
- o :wq para guardar y salir
- q! para salir sin guardar

2. Creando Repositorios

Comando: git init

Previo: Creamos una carpeta git

```
USUARIO@TOSHIBA MINGW64 ~
$ mkdir git
USUARIO@TOSHIBA MINGW64 ~
$ cd git
USUARIO@TOSHIBA MINGW64 ~/git
 1s
USUARIO@TOSHIBA MINGW64 ~/git
```

Crear repositorio

Método 1: git init NombreCarpeta

```
USUARIO@TOSHIBA MINGW64 ~/git
$ git init repo_flor
Initialized empty Git repository in C:/Users/USUARIO/git/repo_flor/.git/
USUARIO@TOSHIBA MINGW64 ~/git
$ 1s
repo_flor/
USUARIO@TOSHIBA MINGW64 ~/git
$ cd repo_flor/
USUARIO@TOSHIBA MINGW64 ~/git/repo_flor (master)
$ 1s
USUARIO@TOSHIBA MINGW64 ~/git/repo_flor (master)
$ 1s -a
    ../ .git/
```

Crear repositorio: Método 2: git init

```
USUARIO@TOSHIBA MINGW64 ~/git/repo_flor (master)
$ cd ..
USUARIO@TOSHIBA MINGW64 ~/git
$ mkdir repo_flor2
USUARIO@TOSHIBA MINGW64 ~/git
$ 1s
repo_flor/ repo_flor2/
USUARIO@TOSHIBA MINGW64 ~/git
$ cd repo_flor2
JSUARIO@TOSHIBA MINGW64 ~/git/repo_flor2
$ git init
Initialized empty Git repository in C:/Users/USUARIO/git/repo_flor2/.git/
USUARIO@TOSHIBA MINGW64 ~/git/repo_flor2 (master)
  ls -a
         .git/
```

Eliminar repositorio:

```
USUARIO@TOSHIBA MINGW64 ~/git/repo_flor2 (master)
 ls -a
 ../ .git/
USUARIO@TOSHIBA MINGW64 ~/git/repo_flor2 (master)
 rm -rf .git
USUARIO@TOSHIBA MINGW64 ~/git/repo_flor2
```

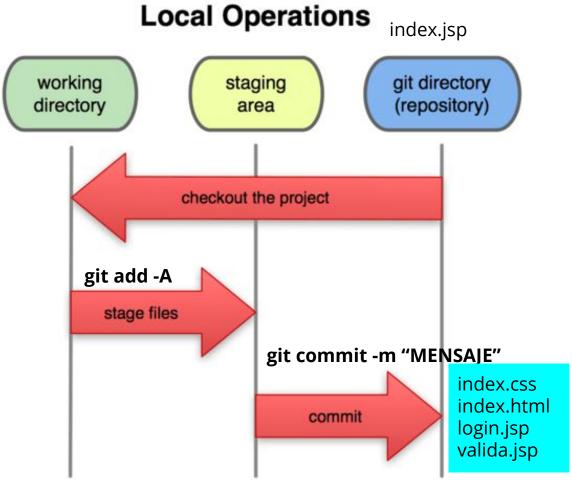
Ejercicio

Crear 4 repositorios (02 con el Método 1 y 02 con el Método 2) con los siguientes nombres:

CARPETA: SistemasTeam

- Ventas_TuApellido
- Sedapal_TuApellido
- Aerolinea_TuApellido
- Matricula_TuApellido

3. Agregando, quitando y viendo status de archivos Comando: git add | rm | status



De Working Directory → **Staging area**

Untracked files: Son los archivos que están en el working directory y requiere pasar al Staging area.

git add [Archivo]: Agrega al Archivo al staging area

git add -A Agrega todos los archivos al Staging area

git rm --cached: Regresa del Staging area al Working Directory

git rm -f [Archivo]: Elimina el archivo.

3.1 Creamos un repositorio: academico Creamos un archivo: index.jsp

```
USUARIO@TOSHIBA MINGW64 ~/git
$ git init academico
Initialized empty Git repository in C:/Users/USUARIO/git/academico/.git/
USUARIO@TOSHIBA MINGW64 ~/git
$ cd academico
USUARIO@TOSHIBA MINGW64 ~/git/academico (master)
$ 1s
USUARIO@TOSHIBA MINGW64 ~/git/academico (master)
$ touch index.jsp
```

3.2 Ver el estado del archivo: git status De color rojo archivos modificados o creados del Working Directory

```
USUARIO@TOSHIBA MINGW64 ~/git/academico (master)

$ git status
On branch master

No commits yet

Untracked files:
   (use "git add <file>..." to include in what will be committed)
        index.jsp

nothing added to commit but untracked files present (use "git add" to track)
```

3.3 WD → Staging Area: git add Archivo De color verde que están en Staging Area

```
USUARIO@TOSHIBA MINGW64 ~/git/academico (master)
$ git add index.jsp
USUARIO@TOSHIBA MINGW64 ~/git/academico (master)
$ git status
On branch master
No commits yet
Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file: index.jsp
```

Unstage: git rm --cached Archivo Para regresar del Staging Area al Working Directory

```
USUARIO@TOSHIBA MINGW64 ~/git/academico (master)
$ git rm --cached index.jsp
rm 'index.jsp'
USUARIO@TOSHIBA MINGW64 ~/git/academico (master)
$ git status
On branch master
No commits yet
Untracked files:
  (use "git add <file>..." to include in what will be committed)
nothing added to commit but untracked files present (use "git add" to track)
```

Pasar varios archivos al Staging Area: git add -A

- Primero creamos varios archivos

```
USUARIO@TOSHIBA MINGW64 ~/git/academico (master)
$ touch index.jsp login.jsp menu.jsp valida.jsp index.css

USUARIO@TOSHIBA MINGW64 ~/git/academico (master)
$ ls
index.css index.jsp login.jsp menu.jsp valida.jsp
```

Pasar varios archivos al Staging Area: git add - A

- Pasamos en bloque al Staging Area

```
USUARIO@TOSHIBA MINGW64 ~/git/academico (master)
$ git add -A
USUARIO@TOSHIBA MINGW64 ~/git/academico (master)
$ git status
On branch master
No commits yet
Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
       new file: index.css
       new file: index.jsp
       new file: login.jsp
       new file: menu.jsp
                   valida.jsp
       new file:
```

Eliminación física: git rm -f

```
USUARIO@TOSHIBA MINGW64 ~/git/academico (master)
$ git status
On branch master
No commits yet
Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
       new file: index.css
       new file: index.jsp
       new file: login.jsp
       new file: menu.jsp
       new file: valida.jsp
USUARIO@TOSHIBA MINGW64 ~/git/academico (master)
$ git rm -f menu.jsp
  'menu.jsp'
```

4. Confirmando cambios

Comando: git commit

De Staging area → **Repositorio**

git commit -m ["mensaje"]: Envía archivos al repositorio

git commit --amend -m ["mensaje"]: Grabar un cambio y sobreescribir el mensaje del commit anterior

git commit --amend: Anexar cambios con el mensaje del commit anterior

git log: Nos muestra el historial de cambios que hemos realizado en el repositorio

git log -p: Nos muestra las diferencias en cada commit

Staging Area → Repositorio: git commit -m "mensaje"

```
USUARIO@TOSHIBA MINGW64 ~/git/academico (master)
$ git commit -m "Iniciando proyecto"
[master (root-commit) 2026475] Iniciando proyecto
4 files changed, 0 insertions(+), 0 deletions(-)
create mode 100644 index.css
create mode 100644 index.jsp
create mode 100644 login.jsp
create mode 100644 valida.jsp
```

Ver cambios: git log

```
Toshiba@DESKTOP-3Q8L8PT MINGW64 ~/git/academico (master)
$ git log
commit 3844069a364351f47df27500b1181eb40fb41bfd (HEAD -> master)
Author: Flor Cerdan <cerdanflor@gmail.com>
       Mon Apr 9 00:10:53 2018 -0500
Date:
    USuario y clave del logueo
commit f3405d58bcac7db317c01cbeb79587133f72436f
Author: Flor Cerdan <cerdanflor@gmail.com>
       Mon Apr 9 00:09:13 2018 -0500
Date:
    Envia mensaje HOla Mundo
commit e97caa08c616e4305d17831def4b0203ee247a87
Author: Flor Cerdan <cerdanflor@gmail.com>
```

Mostrar las diferencias en cada commit: git log -p

```
Mon Apr 9 00:10:53 2018 -0500
Date:
    Usuario, Clave y boton enviar
diff --git a/index.html b/index.html
index fbf869b..af45b39 100644
--- a/index.html
+++ b/index.html
@@ -2,6 +2,9 @@
        <head>
        </head>
        <body>
                hola mundo <br>
                Usuario: <br>
                clave:
                ACEPTAR - CANCELAR
        </body>
 </html>
 No newline at end of file
```

Sobreescribir el mensaje anterior: git commit --amend -m "mensaje"

```
MINGW64:/c/Users/Toshiba/git/academico
oshiba@DESKTOP-3Q8L8PT MINGW64 ~/git/academico (master)
 git commit --amend -m "Usuario, Clave y boton enviar"
[master 9325a89] Usuario, Clave y boton enviar
Date: Mon Apr 9 00:10:53 2018 -0500
1 file changed, 4 insertions(+), 1 deletion(-)
Toshiba@DESKTOP-3Q8L8PT MINGW64 ~/git/academico (master)
$ git log
commit 9325a89272477d27bcf878ec25ce0a9e0c91bd8c (HEAD -> master)
Author: Flor Cerdan <cerdanflor@gmail.com>
Date: Mon Apr 9 00:10:53 2018 -0500
   Usuario, Clave y boton enviar
commit f3405d58bcac7db317c01cbeb79587133f72436f
Author: Flor Cerdan <cerdanflor@gmail.com>
```

5. Etiquetando confirmaciones

Comando: git tag

Git tag

Permite etiquetar nuestros cambios

git tag -a VERSIÓN -m "MENSAJE": (Anotado) Para registrar una versión con un mensaje

git tag VERSION: (ligero) Para registrar una versión

git tag -l: Muestra la lista de etiquetas

git tag -f: Renombra una etiqueta

git tag -d: Elimina etiqueta

Etiquetar version:

git tag -a VERSION -m "MENSAJE"

```
MINGW64:/c/Users/Toshiba/git3/academico
Toshiba@DESKTOP-3Q8L8PT MINGW64 ~/git3/academico (master)
$ git tag -a 0.5 -m "versi[on estable del proyecto"
Toshiba@DESKTOP-3Q8L8PT MINGW64 ~/git3/academico (master)
 git tag -1
Toshiba@DESKTOP-3Q8L8PT MINGW64 ~/git3/academico (master)
```

Etiquetar versión de un commit anterior git tag VERSION SHA

```
Toshiba@DESKTOP-3Q8L8PT MINGW64 ~/git3/academico (master)
$ ait loa
commit ae8e8be512ab9c2c94f17dc549f4b86d8bea308d (HEAD -> master, tag: 0.5)
Author: Flor Cerdan <florc@upeu.edu.pe>
Date: Sun Apr 15 21:11:45 2018 -0500
   usuario y contraseña aprobado
commit 6de011a1206570171b1740e3f832acaf200c19b4
Author: Flor Cerdan <florc@upeu.edu.pe>
Date: Sun Apr 15 20:58:40 2018 -0500
   Mensaje de Saludo
commit 14145c1fe57a9f4e8935841e2f8599e9c522a646
Author: Flor Cerdan <florc@upeu.edu.pe>
Date: Sun Apr 15 20:47:39 2018 -0500
```

Etiquetar versión de un commit anterior git tag VERSION SHA

```
Toshiba@DESKTOP-3Q8L8PT MINGW64 ~/git3/academico (master)
$ git tag 0.3 6de011a1206570171b1740e3f832acaf200c19b4
Toshiba@DESKTOP-3Q8L8PT MINGW64 ~/git3/academico (master)
$ git tag -1
0.3
Toshiba@DESKTOP-3Q8L8PT MINGW64 ~/git3/academico (master)
$ git log
commit ae8e8be512ab9c2c94f17dc549f4b86d8bea308d (HEAD -> master, tag: 0.5)
Author: Flor Cerdan <florc@upeu.edu.pe>
Date: Sun Apr 15 21:11:45 2018 -0500
    usuario y contraseña aprobado
commit 6de011a1206570171b1740e3f832acaf200c19b4 (tag: 0.3)
Author: Flor Cerdan <florc@upeu.edu.pe>
```

Eliminar versión

git tag -d VERSION

```
MINGW64:/c/Users/Toshiba/git3/academico
Toshiba@DESKTOP-3Q8L8PT MINGW64 ~/git3/academico (master)
$ git tag -1
Toshiba@DESKTOP-3Q8L8PT MINGW64 ~/git3/academico (master)
$ git tag -d 0.3
Deleted tag '0.3' (was 6de011a)
Toshiba@DESKTOP-3Q8L8PT MINGW64 ~/git3/academico (master)
```

Renombrar versión

git tag -f -a VERSION -m "MENSAJE" SHA

```
MINGW64:/c/Users/Toshiba/git3/academico
Toshiba@DESKTOP-3Q8L8PT MINGW64 ~/git3/academico (master)
$ git tag -f -a 0.4 -m "Version estable" 6de011a1206570171b1740e3f832acaf200c19b
Toshiba@DESKTOP-3Q8L8PT MINGW64 ~/git3/academico (master)
 git tag -1
Toshiba@DESKTOP-3Q8L8PT MINGW64 ~/git3/academico (master)
$ git tag -d 0.3
Deleted tag '0.3' (was 6de011a)
Toshiba@DESKTOP-3Q8L8PT MINGW64 ~/git3/academico (master)
```