



# UD 03.CONTROL Y GESTIÓN DE VERSIONES

## PARTE 2 DE 4: GIT MONOUSUARIO

v1.0 10.11.20

ceedcv  
CENTRE ESPECÍFIC  
D'EDUCACIÓ A DISTÀNCIA DE  
LA COMUNITAT VALENCIANA

## Entornos de desarrollo (ED)

Sergio Badal

Carlos Espinosa

c.espinosamoreno@edu.gva.es



# UD 03.CONTROL Y GESTIÓN DE VERSIONES

- **Pasos a seguir**

- 1) Lee la documentación (PDF)
- 2) Instala el software necesario (sigue los pasos)
- 3) Realiza los TESTS todas las veces que quieras
- 4) Acude al FORO DE LA UNIDAD
  - Para cualquier duda sobre esta unidad
- 5) Acude al FORO DEL MÓDULO
  - Para cualquier duda sobre el módulo

# UD 03.CONTROL Y GESTIÓN DE VERSIONES

- **¿Qué veremos en esta UNIDAD?**

- **SEMANA 1**

- PARTE 1 DE 4: HERRAMIENTAS CASE Y CONTROL DE VERSIONES
    - PARTE 2 DE 4: CONTROL DE VERSIONES MODO MONOUSUARIO (con GIT)
      - ¿Qué vamos a hacer con Git?
      - Tutorial paso a paso
      - Práctica no evaluable

- **SEMANA 2**

- PARTE 3 DE 4: CONTROL DE VERSIONES MODO MULTIUSUARIO (con GIT)
    - (INICIO) PRÁCTICA EVALUABLE: CONTROL DE VERSIONES

- **SEMANA 3**

- PARTE 4 DE 4: CONTROL DE VERSIONES: INTEGRACIÓN DE GIT
    - (ENTREGA) PRÁCTICA EVALUABLE: CONTROL DE VERSIONES



# UD 03.CONTROL Y GESTIÓN DE VERSIONES

## 3. CONTROL Y GESTIÓN DE VERSIONES

### 3.2.1 ¿QUÉ VAMOS A HACER?

### 3.2.2 GIT MONOUSUARIO

### 3.2.3 PRÁCTICA NO EVALUABLE

# 3.2.1 ¿QUÉ VAMOS A HACER?

## ESTA SEMANA (SEMANA 1)

### GIT MONOUSUARIO

- Primero
  - Instalaremos git.
- Segundo
  - Crearemos un repositorio de prueba **01\_REP\_INICIAL** (de un solo archivo) en nuestro PC, realizando las gestiones básicas sobre él.
- Tercero
  - Crearemos una cuenta en GitHub (si no la tenemos creada ya), subiremos nuestro repositorio e interactuaremos con la cuenta.
- Cuarto
  - Repetiremos el proceso con otro repositorio algo más complejo **02\_REP\_SAGAS**.
- Quinto
  - Subiremos las capturas del resultado del último paso a una práctica no evaluable en PDF.

## LA SIGUIENTE SEMANA (SEMANA 2)

### GIT MULTIUSUARIO

- Primero
  - Crearemos un repositorio con un proyecto HTML de complejidad básica **03\_REP\_HIELO\_CASA** en nuestro PC, simulando que trabajamos desde CASA y uno en la nube llamado **03\_REP\_HIELO**, donde subiremos nuestro repositorio **03\_REP\_HIELO\_CASA**.
- Segundo
  - Clonaremos el repositorio en 2 carpetas más (3 repositorios virtuales), simulando que trabajamos desde casa, desde la oficina y desde el portátil, quedando **03\_REP\_HIELO\_CASA**, **03\_REP\_HIELO\_OFICINA**, **03\_REP\_HIELO\_PORTATIL**.
  - Haremos ciertos cambios en el proyecto HTML desde cada uno de los repositorios.
- Tercero
  - Simularemos un conflicto entre versiones.
- Cuarto
  - Subiremos las capturas del resultado del último paso a una práctica no evaluable en PDF.

## LA ÚLTIMA SEMANA (SEMANA 3)

### GIT EN NETBEANS/ECLIPSE

- Primero
  - Instalaremos en Eclipse/NetBeans los plugins necesarios para poder trabajar con Git.
- Segundo
  - Interactuaremos con los repositorios con los que ya hemos trabajado desde los propios IDE (NetBeans/Eclipse).
- Tercero
  - Subiremos las capturas del resultado del último paso a una práctica no evaluable en PDF.
- Cuarto
  - Realizaremos el test de conceptos y el acumulativo de las unidades anteriores.

### • Cuarto (práctica evaluable)

- (Individualmente) Repetiremos el proceso cambiando el proyecto por uno similar que os proporcionaremos, **04\_REP\_METROPOLITA** esta vez simulando que somos 3 miembros de un equipo de desarrolladores web.



# UD 03.CONTROL Y GESTIÓN DE VERSIONES

## 3. CONTROL Y GESTIÓN DE VERSIONES

3.2.1 ¿QUÉ VAMOS A HACER?

**3.2.2 GIT MONOÚSUARIO**

3.2.3 PRÁCTICA NO EVALUABLE

# 3.2.2 GIT MONOUSUARIO

- Para estas sesiones prácticas de Git **no existe un PDF de APUNTES** como hasta ahora, ya que todos los pasos están explicados con capturas a continuación.
- Si quieres o necesitas ampliar la información tienes, literalmente, centenares de tutoriales de GIT en la red . que pueden ayudarte.
  - Uno muy interesante y recomendado es este de Youtube:
    - <https://www.youtube.com/watch?v=HiXLkL42tMU>
  - Si lo prefieres en modo texto, te recomendamos este:
    - <https://aprendeconalf.es/docencia/git/>



# 3.2.2 GIT MONOUSUARIO

- Las siguientes indicaciones son las recomendadas para poder seguir sin problemas este pequeño tutorial **si eres nuevo en GIT**.
- **Si ya has trabajado alguna vez con GIT** y lo tienes instalado en cualquier otra versión, sistema operativo o con cualquier otra configuración **no es necesario que desinstales nada**.





# 3.2.2 GIT MONOUSUARIO



## ESTA SEMANA (SEMANA 1)

### GIT MONOUSUARIO

#### Primero

- Instalaremos git

- Segundo

- Crearemos un repositorio de prueba

**01\_REP\_INICIAL** (de un solo archivo) en nuestro PC, realizando las gestiones básicas sobre él.

- Tercero

- Crearemos una cuenta en GitHub (si no la tenemos creada ya), subiremos nuestro repositorio e interactuaremos con la cuenta.

- Cuarto

- Repetiremos el proceso con otro repositorio algo más complejo **02\_REP\_SAGAS**

- Quinto

- Subiremos las capturas del resultado del último paso a una práctica no evaluable en PDF.

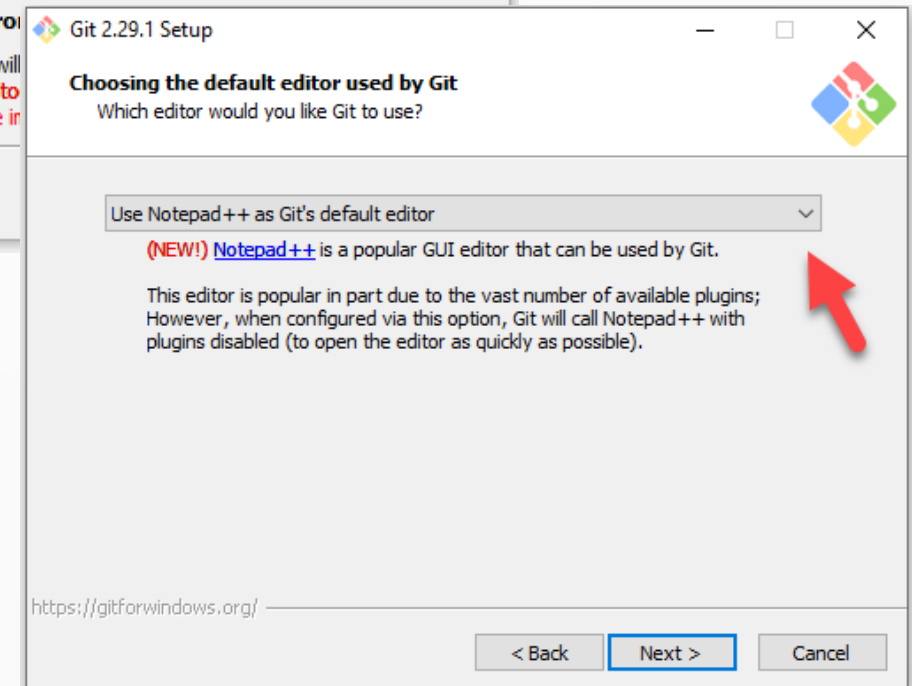
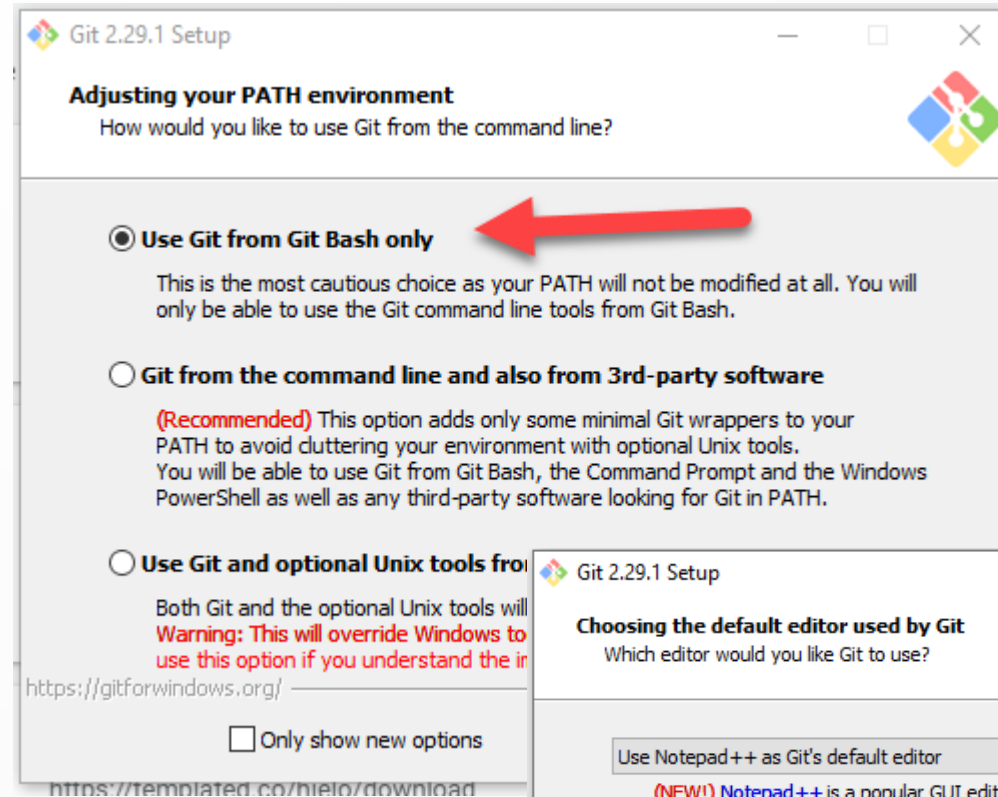
# 3.2.2 GIT MONOUSUARIO

Primero

- Instalaremos git

- **Paso 1: Instalar GIT**

- Instala previamente **Notepad++**, si no lo has hecho ya, para tu sistema operativo y procesador (32/64):
  - <https://notepad-plus-plus.org/downloads/>
- Descarga la última versión de Git para tu sistema operativo:
  - <https://git-scm.com/downloads>
- Si instalas en Windows, te recomendamos que **dejes todas las opciones por defecto salvo estas que verás a la derecha:**



# 3.2.2 GIT MONOUSUARIO

## ESTA SEMANA (SEMANA 1)

### GIT MONOUSUARIO

- Primero
  - Instalaremos git
- Segundo
  - Crearemos un repositorio de prueba **01\_REP\_INICIAL** (de un solo archivo) en nuestro PC, realizando las gestiones básicas sobre él.
- Tercero
  - Crearemos una cuenta en GitHub (si no la tenemos creada ya), subiremos nuestro repositorio e interactuaremos con la cuenta.
- Cuarto
  - Repetiremos el proceso con otro repositorio algo más complejo **02\_REP\_SAGAS**.
- Quinto
  - Subiremos las capturas del resultado del último paso a una práctica no evaluable en PDF.



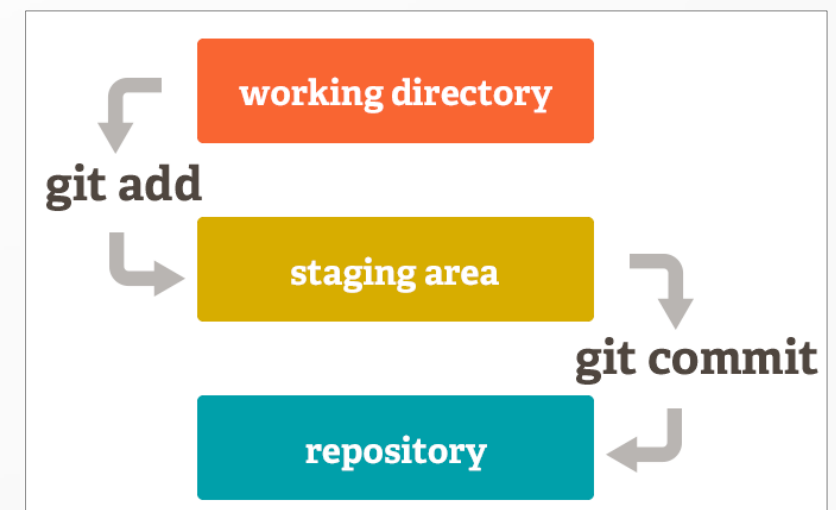
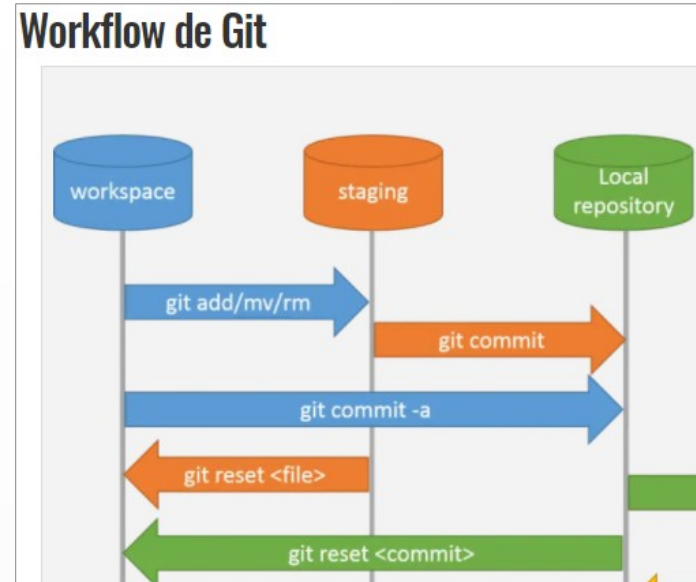
# 3.2.2 GIT MONOUSUARIO

## Segundo

- Crearemos un repositorio de prueba **01\_REP\_INICIAL** (de un solo archivo) en nuestro PC, realizando las gestiones básicas sobre él.

- ¿Y ahora? **(NO HAGAS NADA, SOLO LEE)**

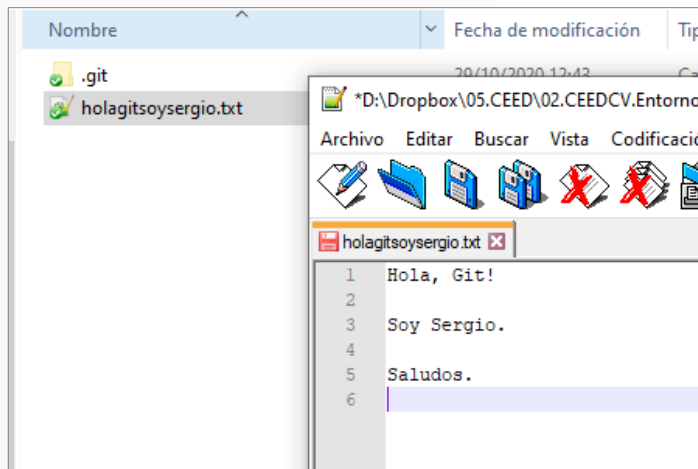
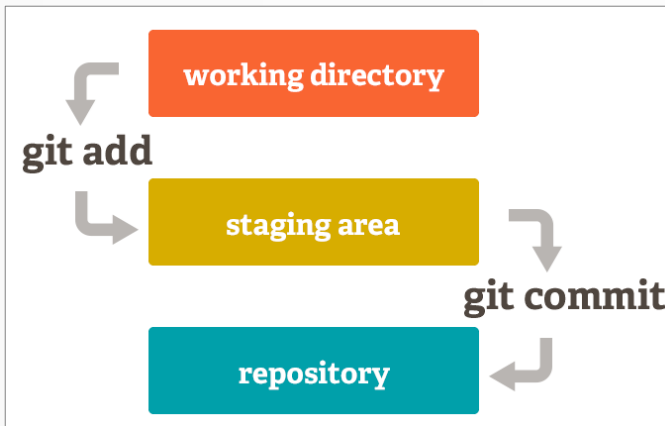
- 1) Crearemos una nueva carpeta.
- 2) Le diremos que esa carpeta es el WORKSPACE de nuestro repositorio con **[git init]**
- 3) Nos identificaremos **[git config --global]**
- 4) Copiaremos un único archivo en el WORKSPACE.
- 5) Lo añadiremos al almacenamiento temporal (STAGING) con la orden **[git add]**.
- 6) Lo grabaremos en el repositorio (LOCAL REPOSITORY) con la orden **[git commit]**



# 3.2.2 GIT MONOUSUARIO

## Segundo

- Crearemos un repositorio de prueba  
**01\_REP\_INICIAL** (de un solo archivo) en nuestro PC, realizando las gestiones básicas sobre él.



## • Paso 2.1: Crear un repositorio de prueba con un único archivo

- 1) Crea una carpeta llamada ED.Practicas.GIT y, dentro de ella, crea otra una carpeta llamada **01\_REP\_INICIAL**
- 2) Crea un archivo llamado **holasoyXXX.txt** con tres frases como las de la captura, indicando tu nombre en lugar de "XXX", y grábalo en esa carpeta.
- 3) Haz botón derecho sobre la carpeta y pulsa **Git Bash Here** para abrir la consola de BASH. Estaremos emulando una consola Linux.
- 4) Luego, desde la consola:
  - 1) Crea el Repositorio [**git init**]
  - 2) Identifícate [**git config --global user.name**] y ... **user.email**
  - 3) Mira el estado del repositorio [**git status**]
  - 4) Añade al staging el archivo [**git add .**]
  - 5) Grábalo en el repositorio con:  
**[git commit -m "mi primer commit"]**

```
MINGW64:/d/Dropbox/05.CEED/02.CEEDCV.Entornos/ED.UD03. Control y gestión de versione...
Sergio@HERAKLION MINGW64 /d/Dropbox/05.CEED/02.CEEDCV.Entor
y gestión de versiones/_ED.Practicas.GIT/01.REP_INICIAL
$ git init
Initialized empty Git repository in D:/Dropbox/05.CEED/02.C
nes/_ED.Practicas.GIT/01.REP_INICIAL/.git/

Sergio@HERAKLION MINGW64 /d/Dropbox/05.CEED/02.CEEDCV.Entor
cticas.GIT/01.REP_INICIAL (master)
$ git config --global user.name "Sergio Badal"

Sergio@HERAKLION MINGW64 /d/Dropbox/05.CEED/02.CEEDCV.Entor
cticas.GIT/01.REP_INICIAL (master)
$ git config --global user.email "sergio.badal@ceedcv.es"

Sergio@HERAKLION MINGW64 /d/Dropbox/05.CEED/02.CEEDCV.Entor
cticas.GIT/01.REP_INICIAL (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be commi
holagitsoysergio.txt

nothing added to commit but untracked files present (use "g
d" to track)

Sergio@HERAKLION MINGW64 /d/Dropbox/05.CEED/02.CEEDCV.Entor
D.UD03. Control y gestión de versiones/_ED.Practicas.GIT/01
INICIAL (master)
$ git add .

Sergio@HERAKLION MINGW64 /d/Dropbox/05.CEED/02.CEEDCV.Entor
.UD03. Control y gestión de versiones/_ED.Practicas.GIT/01.
ICIAL (master)
$ git commit -m "mi primer commit"
[master (root-commit) e5f5222] mi primer commit
1 file changed, 5 insertions(+)
create mode 100644 holagitsoysergio.txt
```

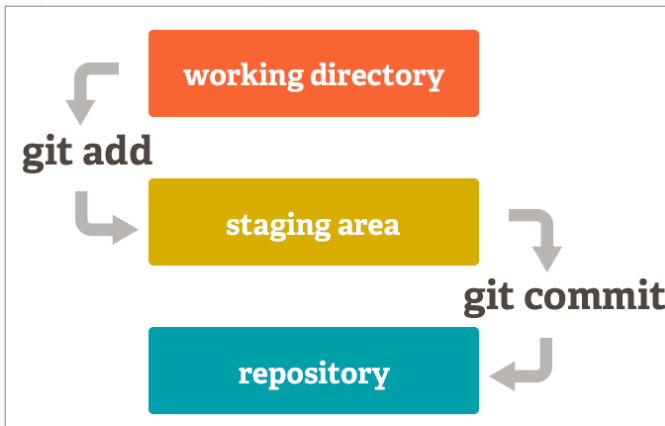
Paso 2.2:



# 3.2.2 GIT MONOUSUARIO

## Segundo

- Crearemos un repositorio de prueba  
**01\_REP\_INICIAL** (de un solo archivo) en nuestro PC, realizando las gestiones básicas sobre él.



```
holagitsoysergio.txt x
1  Hola, Git!
2
3  Soy Sergio. ¿Qué tal?
4
5  Saludos.
6
```

## • Paso 2.2: Nuevos cambios

- 1) Prueba ahora a hacer cambios en el archivo de prueba, añadiendo otra frase diferente. Al cambiarlo, debes saber que se modifica en el WORKSPACE (en la carpeta de tu PC).
- 2) Luego, vuelve a ver el estado del repositorio con `[git status]`.
- 3) Te saldrá en rojo como modificado. Pásalo al STAGING con `[git add .]`
- 4) Finalmente, grábalo en el REPOSITORIO con `[git commit -m "mi segundo commit"]`

```
MINGW64:/d/Dropbox/05.CEED/02.CEEDCV.Entornos/ED.UD03. Control y gestión de versi
Sergio@HERAKLION MINGW64 /d/Dropbox/05.CEED/02.CEEDCV.Entornos/ED.UD03. Control y gestión de versiones/_ED.Practicas.GIT/01.REP_INICIAL
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   holagitsoysergio.txt

no changes added to commit (use "git add" and/or "git commit -a")

Sergio@HERAKLION MINGW64 /d/Dropbox/05.CEED/02.CEEDCV.Entornos/ED.UD03. Control y gestión de versiones/_ED.Practicas.GIT/01.REP_INICIAL
$ git add .
Sergio@HERAKLION MINGW64 /d/Dropbox/05.CEED/02.CEEDCV.Entornos/ED.UD03. Control y gestión de versiones/_ED.Practicas.GIT/01.REP_INICIAL
$ git commit -m "mi segundo commit"
[master 12f8c24] mi segundo commit
1 file changed, 1 insertion(+), 3 deletions(-)
```

# 3.2.2 GIT MONOUSUARIO

## Segundo

- Crearemos un repositorio de prueba  
**01\_REP\_INICIAL** (de un solo archivo) en nuestro PC, realizando las gestiones básicas sobre él.

- **Paso 2.3: Consultar los cambios en el repositorio (ver log)**

1) Prueba las órdenes de la derecha:

[clear] limpia la consola

[ls -la] para listar el WORKSPACE

[git log] para ver el log

2) Ten en cuenta que el HEAD apunta al último **commit**

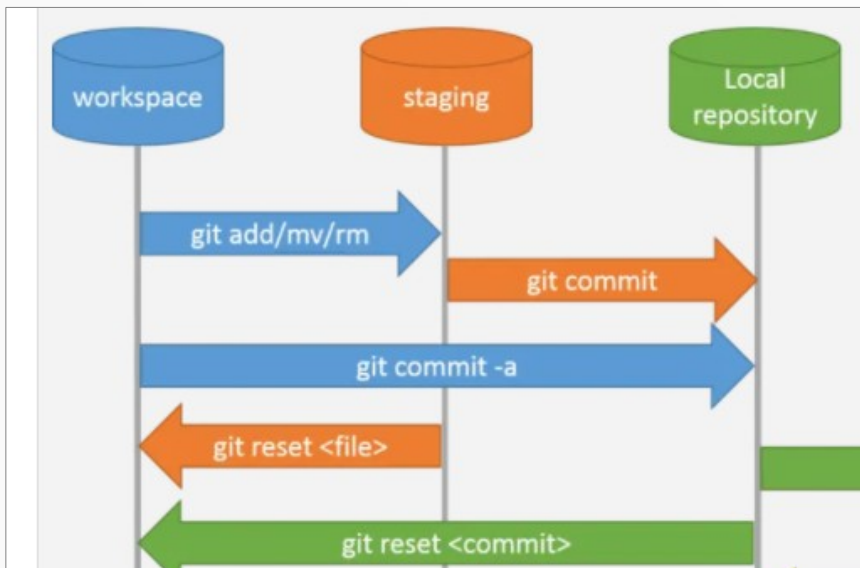
```
MINGW64/d:/Dropbox/05.CEED/02.CEEDCV.Entornos/ED.UD03. Control y gestión de versiones/_ED.Practicas.GIT/01.REP_
Sergio@HERAKLION MINGW64 /d:/Dropbox/05.CEED/02.CEEDCV.Entornos/ED.UD03. Control y gestión de versiones/_ED.Practicas.GIT/01.REP_INICIAL (master)
$ ls -la
total 13
drwxr-xr-x 1 Sergio 197121  0 oct. 29 14:18 ./
drwxr-xr-x 1 Sergio 197121  0 oct. 29 17:19 ../
drwxr-xr-x 1 Sergio 197121  0 oct. 29 18:51 .git/
-rw-r--r-- 1 Sergio 197121 63 oct. 29 18:51 holagitsoysergio.txt

Sergio@HERAKLION MINGW64 /d:/Dropbox/05.CEED/02.CEEDCV.Entornos/ED.UD03. Control y gestión de versiones/_ED.Practicas.GIT/01.REP_INICIAL (master)
$ git log
commit 403dff3d9450185d96da46716f789fd7c46e3b42 (HEAD -> master)
Author: Sergio Badal <sergio.badal@ceedcv.es>
Date: Thu Oct 29 18:51:50 2020 +0100

    mi segundo commit

commit 2abd05d375dea55a4e1643d8db7a9318bb946de4
Author: Sergio Badal <sergio.badal@ceedcv.es>
Date: Thu Oct 29 18:51:34 2020 +0100

    mi primer commit
```



# 3.2.2 GIT MONOUSUARIO

## ESTA SEMANA (SEMANA 1)

### GIT MONOUSUARIO

- Primero
  - Instalaremos git
- Segundo
  - Crearemos un repositorio de prueba **01\_REP\_INICIAL** (de un solo archivo) en nuestro PC, realizando las gestiones básicas sobre él.
- Tercero
  - Crearemos una cuenta en GitHub (si no la tenemos creada ya), subiremos nuestro repositorio e interactuaremos con la cuenta.
- Cuarto
  - Repetiremos el proceso con otro repositorio algo más complejo **02\_REP\_SAGAS**.
- Quinto
  - Subiremos las capturas del resultado del último paso a una práctica no evaluable en PDF.



# 3.2.2 GIT MONOUSUARIO

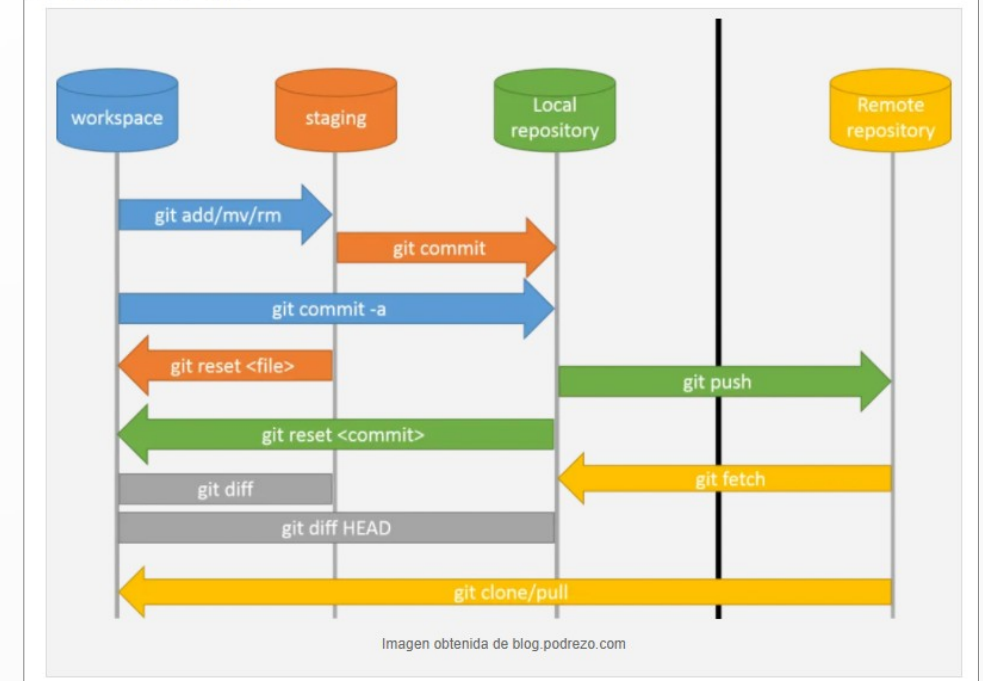
## Tercero

- Crearemos una cuenta en GitHub (si no la tenemos creada ya), subiremos nuestro repositorio e interactuaremos con la cuenta.

- ¿Y ahora? **(NO HAGAS NADA, SOLO LEE)**

- 1) Ahora crearemos una cuenta en GitHub (en la nube) [<https://github.com/join>]
- 2) Crearemos un repositorio llamado **01\_REP\_INICIAL** (desde la web)
- 3) Enlazaremos el repositorio local con esa cuenta/repositorio [**git remote add**]
- 4) Subiremos nuestro archivo a la nube [**git push**]
- 5) Haremos un cambio en la nube (desde la web)
- 6) Lo descargaremos a nuestro repositorio [**git pull**]
- 7) Haremos un cambio en local (notepad++)
- 8) Lo subiremos a la nube [**git push**]

## Workflow de Git

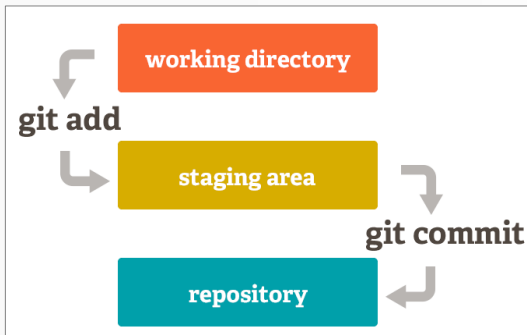




# 3.2.2 GIT MONOUSUARIO

## Tercero

- Crearemos una cuenta en GitHub (si no la tenemos creada ya), subiremos nuestro repositorio e interactuaremos con la cuenta.



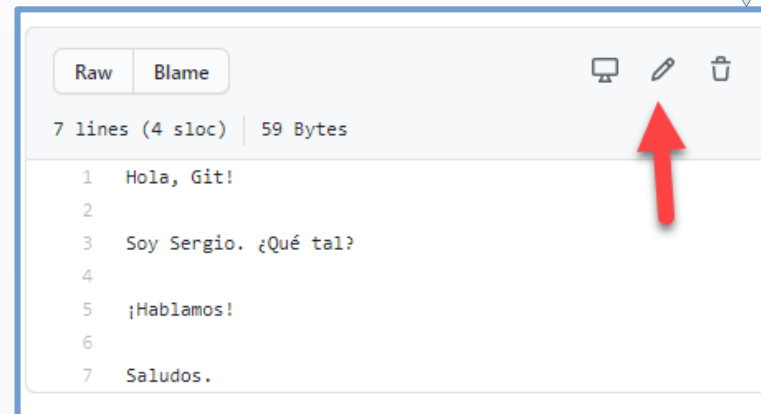
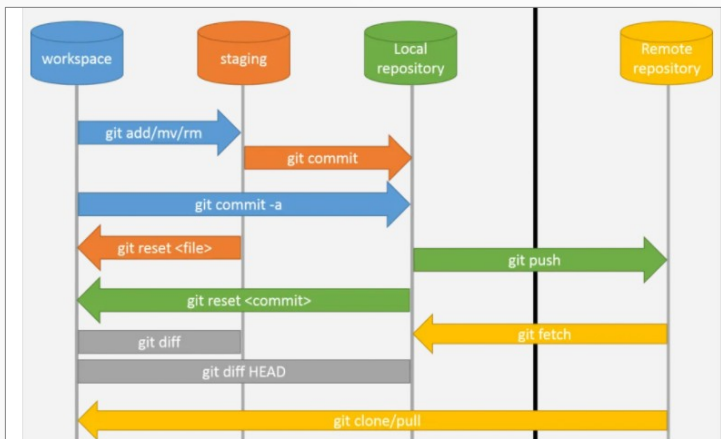
### • Paso 3.1: Crear un repositorio en la nube e interactuar con los dos

- 1) Entra en [<https://github.com/join>] y crea una cuenta
- 2) Desde la web, crea un repositorio llamado **01\_REP\_INICIAL**
- 3) Ya en la consola, en local, enlaza el local con esa cuenta/repositorio  
`[git remote add origin https://github.com/usuario/01_REP_INICIAL]`
- 4) Sube el repositorio a la nube [`git push origin master`]
- 5) Haz un cambio en la nube (desde la web de gitHub)
- 6) Descárgalo a tu repositorio en local [`git pull origin master`]

```
MINGW64:/d/Dropbox/05.CEED/02.CEEDCV.Entornos/ED.UD03. Control y gestión de versione...
Sergio@HERAKLION MINGW64 /d/Dropbox/05.CEED/02.CEEDCV.Entornos/ED
_UD03. Control y gestión de versiones/_ED.Practicas.GIT/01.REP_IN
ICIAL (master)
$ git remote add origin https://github.com/enyzing/01_REP_INICIAL
```

```
Sergio@HERAKLION MINGW64 /d/Dropbox/05.CEED/02.CEEDCV.Entornos/ED
_UD03. Control y gestión de versiones/_ED.Practicas.GIT/01.REP_IN
ICIAL (master)
$ git push origin master
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (6/6), 531 bytes | 177.00 KiB/s, done.
Total 6 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/enyzing/01_REP_INICIAL
 * [new branch]      master -> master
```

```
Sergio@HERAKLION MINGW64 /d/Dropbox/05.CEED/02.CEEDCV.Entornos/ED
_UD03. Control y gestión de versiones/_ED.Practicas.GIT/01.REP_IN
ICIAL (master)
$ git pull origin master
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 687 bytes | 2.00 KiB/s, done.
From https://github.com/enyzing/01_REP_INICIAL
 * branch                master      -> FETCH_HEAD
  fa65c99..f592926 master      -> origin/master
Updating fa65c99..f592926
Fast-forward
 holagitsoysergio.txt | 2 ++
 1 file changed, 2 insertions(+)
```



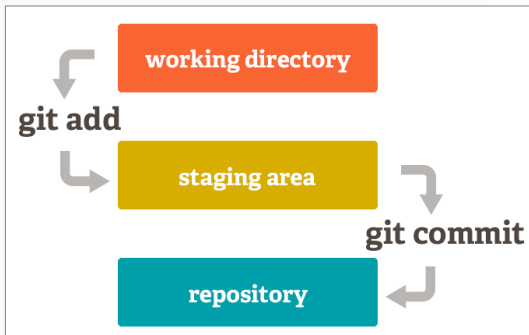
Paso 3.2:



# 3.2.2 GIT MONOUSUARIO

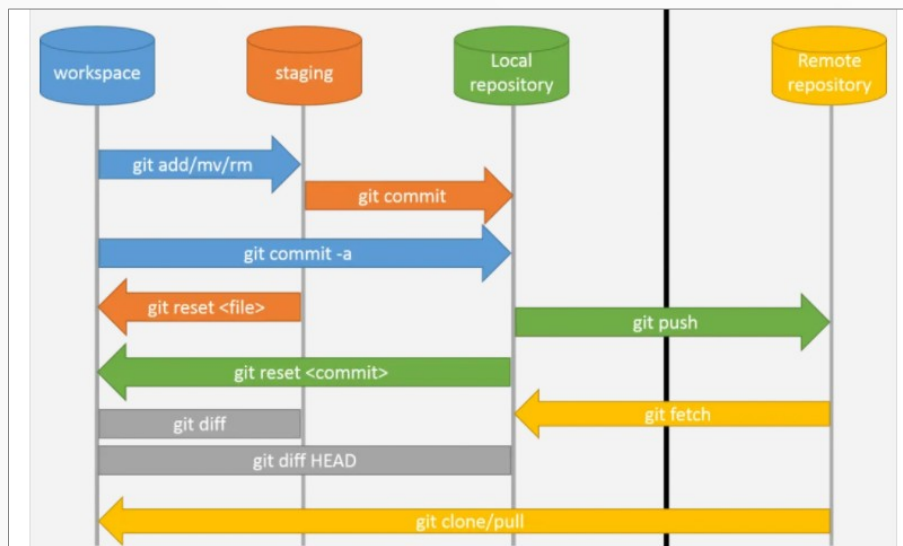
## Tercero

- Crearemos una cuenta en GitHub (si no la tenemos creada ya), subiremos nuestro repositorio e interactuaremos con la cuenta.



### • Paso 3.2: Haz un cambio más en local

- 1) Haz un cambio en local con (notepad++)
- 2) Añádelo al STAGING [git add .]
- 3) Añádelo al REPOSITORIO  
con [git commit -m "mi tercer commit"]
- 4) Súbelo a la nube [git push origin master]



```
holagitsoysergio.txt x
1  Hola, Git!
2
3  Soy Sergio. ¿Qué tal?
4
5  Hablamos!
6
7  Más saludos.
8
```

```
Sergio@HERAKLION MINGW64 /d/Dropbox/05.CEED/02.CEEDCV.Entornos/E
.UD03. Control y gestión de versiones/_ED.Practicas.GIT/01.REP_
ICIAL (master)
$ git add .

Sergio@HERAKLION MINGW64 /d/Dropbox/05.CEED/02.CEEDCV.Entornos/E
.UD03. Control y gestión de versiones/_ED.Practicas.GIT/01.REP_
ICIAL (master)
$ git commit -m "mi tercer commit"
[master 38d3d72] mi tercer commit
1 file changed, 2 insertions(+)

Sergio@HERAKLION MINGW64 /d/Dropbox/05.CEED/02.CEEDCV.Entornos/E
.UD03. Control y gestión de versiones/_ED.Practicas.GIT/01.REP_
ICIAL (master)
$ git push origin master
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 318 bytes | 159.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/enyzing/01_REP_INICIAL
f592926..38d3d72 master -> master
```

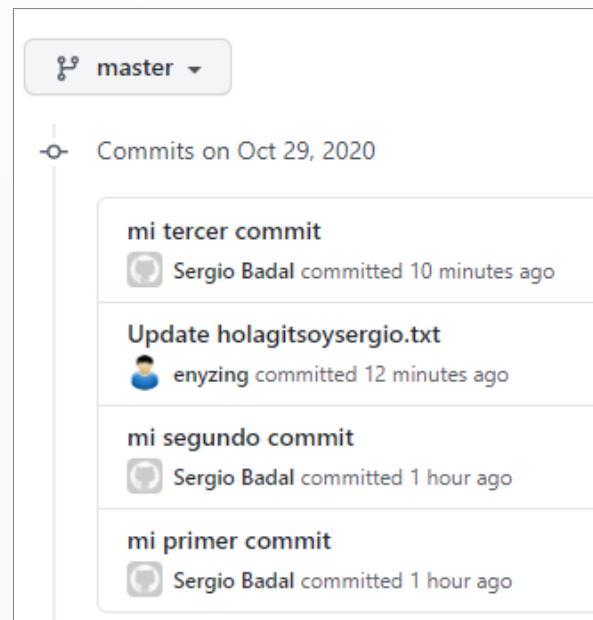
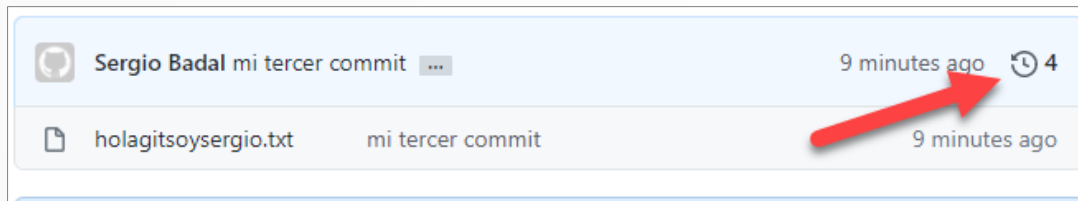
# 3.2.2 GIT MONOUSUARIO

## Tercero

- Crearemos una cuenta en GitHub (si no la tenemos creada ya), subiremos nuestro repositorio e interactuaremos con la cuenta.

### • Paso 3.3: Consultar los cambios en el repositorio (log)

- 1) Prueba las órdenes de la derecha [**git log**] para ver el log
- 2) Recuerda que [**clear**] limpia la consola.
- 3) Ten en cuenta que el HEAD apunta al último commit
- 4) Ve también a github y comprueba el log en la nube



```
MINGW64:/d/Dropbox/05.CEED/02.CEEDCV.Entornos/ED.UD03. Control y gestión de versione...
Sergio@HERAKLION MINGW64 /d/Dropbox/05.CEED/02.CEEDCV.Entornos/ED.UD03. Control y gestión de versiones/_ED.Practicas.GIT/01.REP_INICIAL (master)
$ ls -la
total 13
drwxr-xr-x 1 Sergio 197121  0 oct. 29 14:18 ./
drwxr-xr-x 1 Sergio 197121  0 oct. 29 12:42 ../
drwxr-xr-x 1 Sergio 197121  0 oct. 29 14:19 .git/
-rw-r--r-- 1 Sergio 197121 73 oct. 29 14:18 holagitsoysergio.txt

Sergio@HERAKLION MINGW64 /d/Dropbox/05.CEED/02.CEEDCV.Entornos/ED.UD03. Control y gestión de versiones/_ED.Practicas.GIT/01.REP_INICIAL (master)
$ git log
commit 38d3d726261c83d13ff5287c5e83cb533b3da150 (HEAD -> master, origin/master)
Author: Sergio Badal <sergio.badal@ceedcv.es>
Date: Thu Oct 29 14:19:39 2020 +0100

    mi tercer commit

commit f5929264d2ae8466ddcaafffa5a9316daed1ea66
Author: enyzing <sergio.badal@gmail.com>
Date: Thu Oct 29 14:17:13 2020 +0100

    Update holagitsoysergio.txt

commit fa65c9922f2304db311354d350b617c6ad5bc6ab
Author: Sergio Badal <sergio.badal@ceedcv.es>
Date: Thu Oct 29 13:31:46 2020 +0100

    mi segundo commit

commit e5f52224373cf57003bdd4addc5c2224be2f039e
Author: Sergio Badal <sergio.badal@ceedcv.es>
Date: Thu Oct 29 13:06:27 2020 +0100

    mi primer commit
```

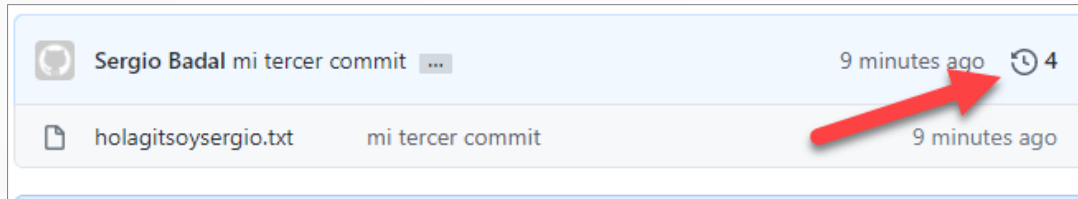
# 3.2.2 GIT MONOUSUARIO

## Tercero

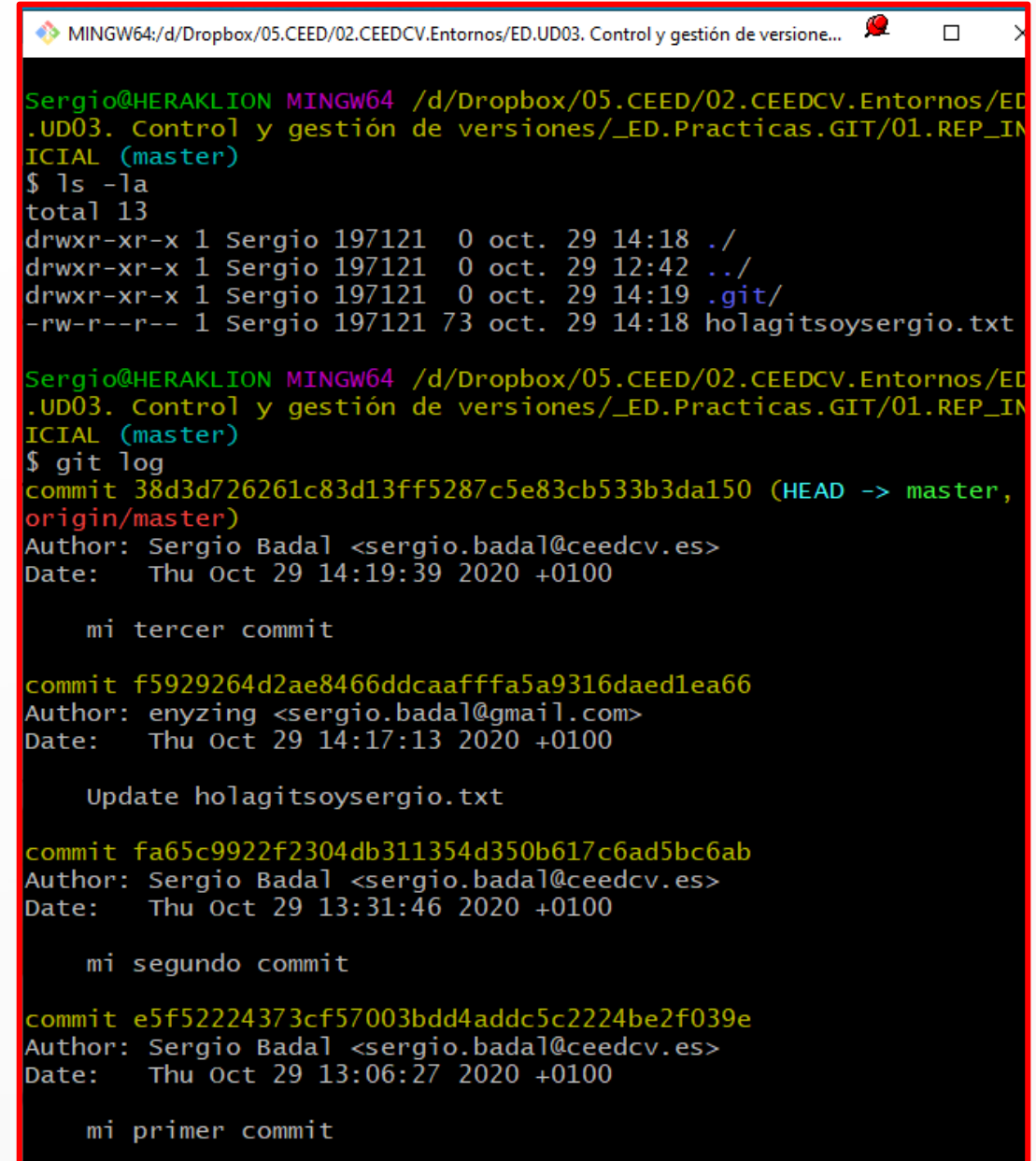
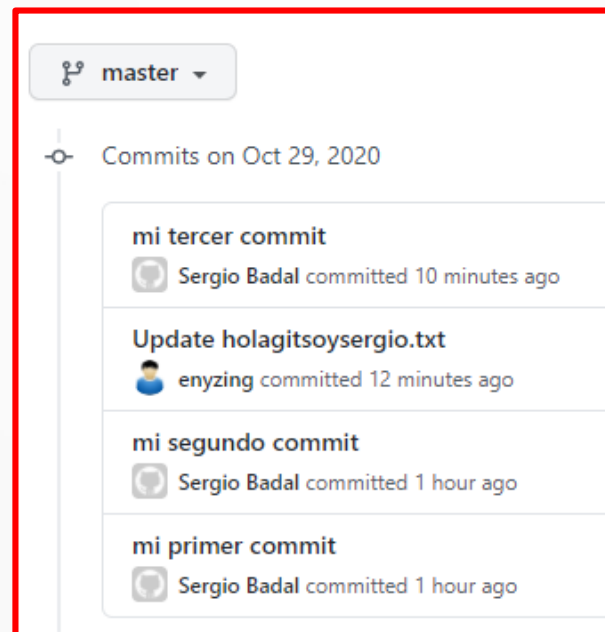
- Crearemos una cuenta en GitHub (si no la tenemos creada ya), subiremos nuestro repositorio e interactuaremos con la cuenta.

### • Paso 3.3: Consultar los cambios en el repositorio (log)

- 1) Prueba las órdenes de la derecha [**git log**] para ver el log
- 2) Recuerda que [**clear**] limpia la consola.
- 3) Ten en cuenta que el HEAD apunta al último commit
- 4) Ve también a github y comprueba el log en la nube



Esas dos capturas enmarcadas en rojo serán las que tendrás que entregar en las siguientes prácticas (evaluables y no evaluables)



# UD 03.CONTROL Y GESTIÓN DE VERSIONES

## **3. CONTROL Y GESTIÓN DE VERSIONES**

3.2.1 ¿QUÉ VAMOS A HACER?

3.2.2 GIT MONOUSUARIO

**3.2.3 PRÁCTICA NO EVALUABLE**

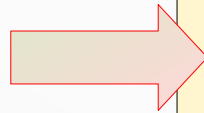


# 3.2.3 PRÁCTICA NO EVALUABLE

## ESTA SEMANA (SEMANA 1)

### GIT MONOUSUARIO

- Primero
  - Instalaremos git
- Segundo
  - Crearemos un repositorio de prueba **01\_REP\_INICIAL** (de un solo archivo) en nuestro PC, realizando las gestiones básicas sobre él.
- Tercero
  - Crearemos una cuenta en GitHub (si no la tenemos creada ya), subiremos nuestro repositorio e interactuaremos con la cuenta.
- Cuarto
  - Repetiremos el proceso con otro repositorio algo más complejo **02\_REP\_SAGAS**.
- Quinto
  - Subiremos las capturas del resultado del último paso a una práctica no evaluable en PDF.





# 3.2.3 PRÁCTICA N.E.

Cuarto

- Repetiremos el proceso con otro repositorio algo más complejo 02\_REP\_SAGAS.

- Vamos a simular que estamos catalogando **SAGAS DE PELÍCULAS** para una futura web de cine.
- De cada saga, de momento, solo queremos el prólogo/introducción y los capítulos (películas) que la conforman.
- Comenzaremos con unos archivos incompletos e iremos completandolos con interacciones entre los repositorios de Git locales y en la nube.



# 3.2.3 PRÁCTICA N.E.

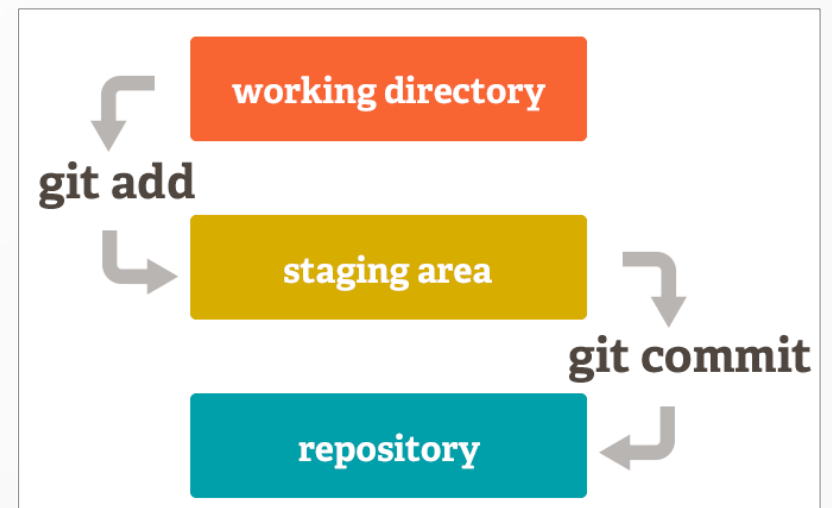
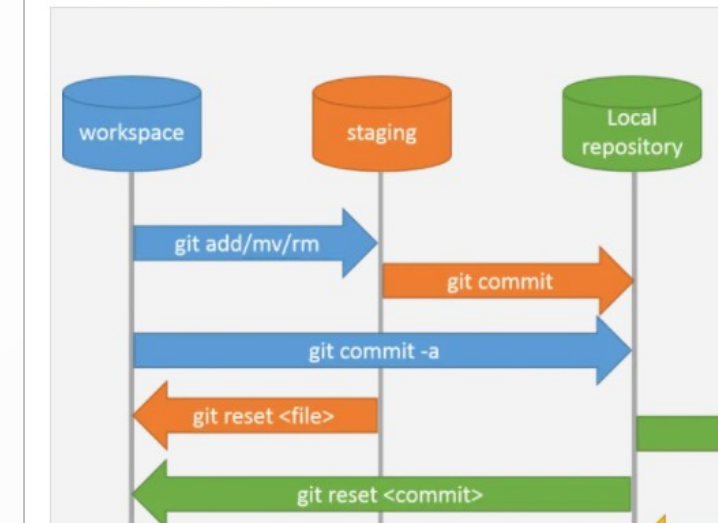
Cuarto

- Repetiremos el proceso con otro repositorio algo más complejo **02\_REP\_SAGAS**.

- ¿Y ahora? **(NO HAGAS NADA, SOLO LEE)**

- 1) Crearemos una nueva carpeta.
- 2) Le diremos que esa carpeta es el WORKSPACE de nuestro repositorio con `[git init]`
- 3) Nos identificaremos `[git config --global]`
- 4) Copiaremos **los archivos base del proyecto** en el WORKSPACE.
- 5) Los añadiremos al almacenamiento temporal (STAGING) con la orden `[git add]`.
- 6) Los grabaremos en el repositorio (LOCAL REPOSITORY) con la orden `[git commit]`
- 7) Crearemos un repositorio en GitHub.
- 8) Haremos cambios “arriba” y “abajo” hasta completar el proyecto.

## Workflow de Git

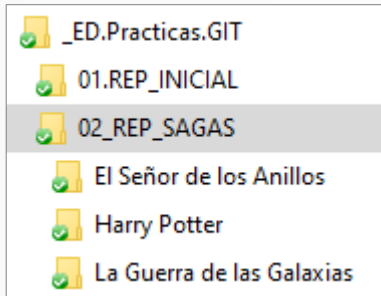
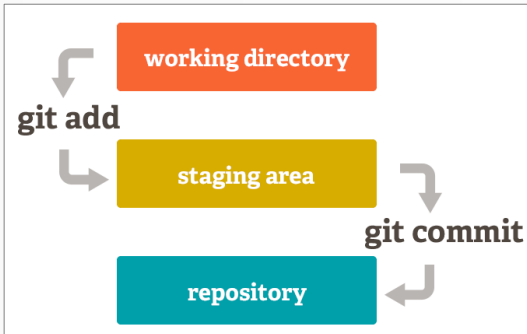




# 3.2.3 PRÁCTICA N.E.

Cuarto

- Repetiremos el proceso con otro repositorio algo más complejo **02\_REP\_SAGAS**.



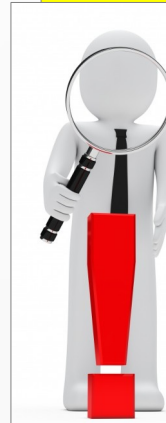
## • Paso 4.1: Crear un repositorio con varios archivos

- 1) Crea una carpeta dentro de ED.Practicas.GIT llamada **02\_REP\_SAGAS**
- 2) Dentro de esa nueva carpeta, descarga y descomprime el archivo **sagas.zip** que encontrarás en el Aula Virtual
- 3) Verás que cada saga tiene dos archivos:  
  
xxx\_capitulos.txt  
xxx\_prologo.txt
- 4) Haz botón derecho sobre la carpeta y pulsa **Git Bash Here** para abrir la consola de BASH. Estaremos emulando una consola Linux.
- 5) Luego, desde la consola:
  - 1) Crea el Repositorio [git init]
  - 2) Identifícate [git config --global user.name] y ... user.email
  - 3) Mira el estado del repositorio [git status]
  - 4) Añade al staging TODOS LOS ARCHIVOS [git add .]
  - 5) Grábalos en repositorio [git commit -m "commit inicial sagas"]

## • Paso 4.2: Sigue los pasos que hemos visto en el ejemplo anterior para conseguir estos hitos:

- 1) Crea en la nube un nuevo repositorio llamado **02\_REP\_SAGAS**
- 2) Sube (a la nube) los archivos.
- 3) Completa en la nube la saga de La Guerra de las Galaxias.
- 4) Baja (a local) los cambios.
- 5) Completa en local el resto de sagas.
- 6) Sube (a la nube) los cambios.

## Acostúmbrate a la terminología:



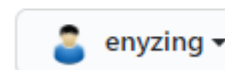
**En la nube:** en GitHub.com

**En local:** en tu ordenador

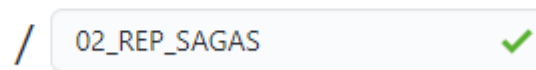
**Bajar:** pull desde gitHub

**Subir:** push hacia gitHub

Owner \*



Repository name \*



**Paso 4.3:**

# 3.2.3 PRÁCTICA N.E.

## ESTA SEMANA (SEMANA 1)

### GIT MONOUSUARIO

- Primero
  - Instalaremos git
- Segundo
  - Crearemos un repositorio de prueba **01\_REP\_INICIAL** (de un solo archivo) en nuestro PC, realizando las gestiones básicas sobre él.
- Tercero
  - Crearemos una cuenta en GitHub (si no la tenemos creada ya), subiremos nuestro repositorio e interactuaremos con la cuenta.
- Cuarto
  - Repetiremos el proceso con otro repositorio algo más complejo **02\_REP\_SAGA**.
- Quinto
  - Subiremos las capturas del resultado del último paso a una práctica no evaluable en PDF.



## • Paso 4.3: Entrega la práctica no evaluable

- Entrega un PDF con estas capturas:
  - Log local
    - Del repositorio en local
  - Log de la nube
    - Del repositorio en la nube