

# Práctica 1 - Uso de Git y GitHub

Pedro Prieto Alarcón

16/09/2022

## Índice

<b>1. Preparación</b>	<b>2</b>
<b>2. Fundamentos de Git I</b>	<b>2</b>
<b>3. Fundamentos de Git II</b>	<b>4</b>
<b>4. GitHub</b>	<b>6</b>
<b>5. Entrega de la tarea</b>	<b>7</b>

## 1. Preparación

Realiza las tareas que se indican a continuación. Incluye las **capturas de pantalla** que se pidan en un documento tipo **LibreOffice** o **Word**.

Cuando se pida realizar un *commit* recuerda que previamente hay que añadir los archivos al área de preparación si no se ha indicado antes en las instrucciones. En esos casos, un *commit* significa ejecutar los comandos `git add` y `git commit`.

**¡IMPORTANTE!** No utilices el bloc de notas de Windows para editar los archivos de texto de las tareas. Utiliza en su lugar un editor específico. Algunas sugerencias son:

- Visual Studio Code
- Atom
- Brackets
- Sublime Text
- Notepad ++

El bloc de notas de Windows utiliza una codificación de caracteres específica de Windows y además incluye la extensión `.txt` por defecto al final de los nombres de archivo. Por tanto, no es una buena elección para trabajar.

## 2. Fundamentos de Git I

1. Instala Git en tu sistema operativo. Adjunta una captura de pantalla en la que aparezca el resultado de la ejecución del comando `git --version`.

```
PS E:\CESA\Puesta en producción\Codigo\S1R1> git --version
git version 2.39.2.windows.1
```

2. Realiza la **configuración de Git** según lo indicado en el tema (nombre, correo electrónico y editor de preferencia). Adjunta una captura de pantalla con el resultado de la ejecución de los comandos de configuración.

```
PS E:\CESA\Puesta en producción\Codigo\S1R1> git config --global user.name "Roque Serrano" --replace-all
PS E:\CESA\Puesta en producción\Codigo\S1R1> git config --global user.name
Roque Serrano
```

```
PS E:\CESA\Puesta en producción\Codigo\S1R1> git config --global user.email
roqsercas@alu.edu.gva.es
```

```
PS E:\CESA\Puesta en producción\Código\S1R1> git config --global core.editor "C:\Program Files\Microsoft VS Code\Code.exe"
PS E:\CESA\Puesta en producción\Código\S1R1> git config --global core.editor
C:\Program Files\Microsoft VS Code\Code.exe
```

3. Crea una carpeta denominada S1R1. Realiza las siguientes acciones en ella:
- a) Crea un repositorio Git.
  - b) Crea un fichero denominado `libros.txt`. Añade tres títulos de libros cada uno en una línea distinta.
  - c) Haz un primer *commit*.
  - d) Añade dos libros al archivo `libros.txt`.

- e) Haz un segundo *commit*.
- f) Crea un fichero denominado `peliculas.txt`. Añade tres títulos de películas a dicho archivo.
- g) Haz una captura de pantalla del comando `git status`.

```
PS E:\CESA\Puesta en producción\Codigo\S1R1> git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
      peliculas.txt

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

- h) Crea un fichero denominado `comidas.txt`. Añade tres nombres de comidas a dicho archivo.
- i) Haz un tercer *commit* que incluya los archivos `peliculas.txt` y `comidas.txt`.
- j) Elimina el archivo `comidas.txt` desde el navegador de archivos.
- k) Añade dos películas más al archivo `peliculas.txt`.
- l) Haz una captura de pantalla que muestre los cambios en el directorio de trabajo.

```
PS E:\CESA\Puesta en producción\Codigo\S1R1> git log --graph
* commit 7b7901609e8f2f94b11cd4a4c3c5f2aa46b26b4a (HEAD -> master)
| Author: Roque Serrano <roqsercas@alu.edu.gva.es>
| Date: Tue Oct 17 17:12:29 2023 +0200
|
| Pelis y comidas
|
* commit 8182b07de9efc2a7f2d7b91e653c8a8c97038d7d
| Author: Roque Serrano <roqsercas@alu.edu.gva.es>
| Date: Tue Oct 17 17:04:25 2023 +0200
|
| Lista de 5 libros
|
* commit 39cf373219e797091f3313e870c74c5d3809368b
| Author: Roque Serrano <roqsercas@alu.edu.gva.es>
| Date: Tue Oct 17 17:02:37 2023 +0200
|
| Lista de 3 libros
```

- m) Añade los cambios al área de preparación.
- n) Haz una captura de pantalla del comando `git status`. Debe indicar que se ha borrado el archivo `comidas.txt` y que se ha modificado

el archivo `peliculas.txt`.

```
PS E:\CESA\Puesta en producción\Codigo\S1R1> git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    deleted:    comidas.txt
    modified:   peliculas.txt
```

- ñ) Haz un cuarto *commit*.
- o) Crea un archivo denominado `datos.bak`. Añade tres títulos de libros a dicho archivo. **¡IMPORTANTE! No añadas el archivo al área de preparación ni hagas ningún commit.**
- p) Crea una subcarpeta denominada `output`. Crea un archivo denominado `salida.txt` en su interior. Escribe tu nombre y apellidos en dicho archivo. **¡IMPORTANTE! No añadas los archivos al área de preparación ni hagas ningún commit.**
- q) Haz una captura de pantalla del comando `git status`. Deben aparecer los archivos `datos.bak` y `output/salida.txt` como archivos nuevos (color rojo).

```
PS E:\CESA\Puesta en producción\Codigo\S1R1> git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    datos.bak
    output/

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

- r) Crea un archivo `.gitignore` para que los ficheros con extensión `.bak` y el contenido de la carpeta `output/` no se incluyan en el repositorio.
- s) Haz una nueva captura de pantalla del comando `git status`. Ahora no deben aparecer los archivos `datos.bak` y `output/salida.txt` como archivos nuevos, sino que en su lugar debe aparecer únicamente el archivo `.gitignore`.

```
PS E:\CESA\Puesta en producción\Codigo\S1R1> git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    .gitignore

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

- t)* Haz un último *commit* para incluir el archivo `.gitignore` en el repositorio.

- u) Haz una captura de pantalla que muestre el histórico de cambios del repositorio.

```
PS E:\CESA\Puesta en producción\Codigo\S1R1> history

Id CommandLine
--
1 git --version
2 git user.name
3 git config --global user.name
4 git config --global user.name Roque Serrano
5 git config --global user.name Roque Serran2o
6 git config --global user.name Roque Serrano
7 git config --global user.name
8 git config --global user.name "Roque Serrano"
9 git config --global user.name
10 git config --global user.name
11 git config --global user.name "Roque Serrano" --replace-all
12 git config --global user.name
13 git config --global user.email
14 git config --global core.editor
15 code --wait
16 git config --global core.editor
17 git config --global core.editor "C:\Program Files\Microsoft VS Code\Code.exe"
18 git config --global core.editor
19 git init
20 git add .
21 git commit "Lista de 3 libros"
22 git commit -m "Lista de 3 libros"
23 git add .
24 git commit -m "Lista de 5 libros"
25 git status
26 git add .
27 git commit -m "Pelis y comidas"
28 git diff
29 git log --graph
30 git log
31 git log --graph
32 git add .
33 git status
34 git commit -m "Comidas eliminadas"
35 git status
36 git status
37 git status
38 git status
39 git status
40 git commit -m "Ignorar"
41 git add .
42 git commit -m "Ignorar"
```

### 3. Fundamentos de Git II

1. Crea una carpeta denominada S2R1. Realiza las siguientes acciones en ella:

- a) Crea un repositorio Git.
- b) Crea un fichero denominado `actores.txt`. Añade tres nombres de actores cada uno en una línea distinta.
- c) Haz un primer *commit*.
- d) Crea una rama denominada `test`.
- e) Cambia a la rama `test`
- f) En la rama `test` crea un fichero denominado `actrices.txt`. Añade tres nombres de actrices y realiza un *commit* en dicha rama.
- g) Haz una captura de pantalla del resultado del comando `git log --graph --all`.

```
PS E:\CESA\Puesta en producción\Codigo\S2R1> git log --graph --all
* commit e5d10d058a6a0231c0f6f36b8a035fa5e5c37889 (HEAD -> test)
 | Author: Roque Serrano <roqsercas@alu.edu.gva.es>
 | Date: Tue Oct 17 17:45:01 2023 +0200
 |
 |   Actrices
 |
 * commit 8845a2fd1cc9cca9b7c5711a95cdaf1798cc10b3 (master)
 | Author: Roque Serrano <roqsercas@alu.edu.gva.es>
 | Date: Tue Oct 17 17:43:07 2023 +0200
 |
 |   Actores
```

- h) Cambia a la rama `master`.
- i) Incorpora los cambios de la rama `test` a la rama `master`. Haz una captura de pantalla de los comandos que has utilizado y de su resultado.

```
PS E:\CESA\Puesta en producción\Codigo\S2R1> git checkout master
Switched to branch 'master'
PS E:\CESA\Puesta en producción\Codigo\S2R1> git merge test
Updating 8845a2f..e5d10d0
Fast-forward
 actrices.txt | 3 +++
 1 file changed, 3 insertions(+)
 create mode 100644 actrices.txt
```



- j) Crea una segunda rama denominada `test2`. La rama `test2` apunta al mismo *commit* que la rama `master` en este momento.
- k) En la rama `master`, añade una actriz al fichero `actrices.txt` y haz un *commit*.
- l) Cambia a la rama `test2`
- m) En la rama `test2`, añade una actriz al fichero `actrices.txt` y haz otro *commit*.
- n) Haz una captura de pantalla del resultado del comando `git log --graph --all`. Debe haber dos caminos distintos: uno para la rama `master` y otro para la rama `test2`.

```
* commit 892f15591f459b15fc135f0bafd96dad838025c (HEAD -> test2)
| Author: Roque Serrano <roqsercas@alu.edu.gva.es>
| Date: Thu Oct 19 16:36:56 2023 +0200
|
| Penélope Cruz
|
* commit 31826090990839bb1e95b740c81cfbc6231f9e35 (master)
| / Author: Roque Serrano <roqsercas@alu.edu.gva.es>
| Date: Thu Oct 19 16:36:13 2023 +0200
|
| Frances McDormand
|
* commit e5d10d058a6a0231c0f6f36b8a035fa5e5c37889 (test)
| Author: Roque Serrano <roqsercas@alu.edu.gva.es>
|
:|
```

- ñ) Cambia a la rama `master`
- o) Incorpora los cambios de la rama `test2` a la rama `master`. ¿Se produce un conflicto? De ser así realiza una captura del comando `git status`.

```
PS E:\CESA\Puesta en producción\Codigo\S2R1> git status
On branch master
You have unmerged paths.
  (fix conflicts and run "git commit")
  (use "git merge --abort" to abort the merge)

Unmerged paths:
  (use "git add <file>..." to mark resolution)
    both modified:   actrices.txt

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

- p) Resuelve el conflicto incorporando los dos nombres de actrices.
- q) Haz una captura de pantalla del resultado del comando `git log --graph --all`. Observa que se ha creado un nuevo *commit* que integra los dos caminos anteriores.

```
*   commit 1f148c2f59a48500b3b564ddfc2b41a662398db6 (HEAD -> master)
|\  Merge: 3182609 892f155
| | Author: Roque Serrano <roqsercas@alu.edu.gva.es>
| | Date:   Thu Oct 19 16:38:40 2023 +0200
| |
| |     Conflicto resuelto
| |
| | *   commit 892f15591f459b15fc135f0bafd96dad838025c (test2)
| | | Author: Roque Serrano <roqsercas@alu.edu.gva.es>
| | | Date:   Thu Oct 19 16:36:56 2023 +0200
| | |
| | |     Penélope Cruz
```

2. Crea una carpeta denominada `S2R2-remoto`. Inicializa un repositorio Git en su interior mediante el comando `git init --bare`. Esta carpeta se utilizará como repositorio remoto.
3. Clona el repositorio `S2R2-remoto` en una carpeta denominada `S2R2`. Adjunta captura de pantalla del resultado del comando de clonado. A continuación realiza las siguientes acciones en el repositorio `S2R2`:

```
PS E:\CESA\Puesta en producción\Codigo\S2R2> git clone ../S2R2-remoto
Cloning into 'S2R2-remoto'...
warning: You appear to have cloned an empty repository.
done.
```

- a) Crea un archivo denominado `directores.txt`. Añade el nombre de tres directores de cine.
- b) Haz un *commit*.
- c) Realiza un *push* al repositorio remoto. Adjunta captura de pantalla del resultado.

```
PS E:\CESA\Puesta en producción\Codigo\S2R2\S2R2-remoto> git push
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 262 bytes | 262.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To E:\CESA\Puesta en producción\Codigo\S2R2/../S2R2-remoto
* [new branch]      master -> master
```

- d) Crea una rama denominada `version1`.
- e) Cambia a la rama `version1`.
- f) En la rama `version1` añade el nombre de dos directores de cine más al archivo `directores.txt` y haz un *commit* de los cambios.
- f) Realiza un *push* de la rama al repositorio remoto de manera que **quede asociada a la rama remota del mismo nombre**. Adjunta captura de pantalla del resultado.

```
PS E:\CESA\Puesta en producción\Codigo\S2R2\S2R2-remoto> git push -u origin version1
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), 330 bytes | 330.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To E:/CESA/Puesta en producción/Codigo/S2R2/./S2R2-remoto
 * [new branch]      version1 -> version1
branch 'version1' set up to track 'origin/version1'.
```

4. Clona el repositorio `S2R2-remoto` en una segunda carpeta denominada `S2R3`. Realiza las siguientes acciones sobre ella:

- a) Muestra en la consola el contenido del fichero `directores.txt` y el resultado del comando `git status`. Debe mostrar tres directores.
- b) Cambia a la rama `version1`. Muestra el resultado del comando. Comprueba que se crea una rama local `version1` con el contenido de la rama remota `origin/version1` y enlazada con ella. Al clonar el repositorio la rama no existía (solo se clona la rama principal, `master`), pero al cambiar a una rama que existe en el remoto se produce su creación local y enlazado con su correspondiente remota.

```
Stanley Kubrick
PS E:\CESA\Puesta en producción\Codigo\S2R3\S2R2-remoto> git switch version1
Switched to a new branch 'version1'
branch 'version1' set up to track 'origin/version1'.
PS E:\CESA\Puesta en producción\Codigo\S2R3\S2R2-remoto> █
```

- c) Muestra el contenido del fichero `directores.txt` por la pantalla. Comprueba que se muestran los 5 nombres de directores esperados. Adjunta captura de pantalla.

```
PS E:\CESA\Puesta en producción\Codigo\S2R3\S2R2-remoto> cat .\directores.txt
Martin Scorsese
Ridley Scott
Stanley Kubrick
Steven Spielberg
Ingmar Bergman
```

- d) Cambia a la rama `master`.
- e) Incorpora los cambios de la rama `version1` a la rama `master`.
- f) Sube la rama `master` actualizada al servidor. Adjunta captura de pantalla del resultado del comando.

```
PS E:\CESA\Puesta en producción\Codigo\S2R3\S2R2-remoto> git push
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
To E:/CESA/Puesta en producción/Codigo/S2R3/./S2R2-remoto
 81c1d5d..7096461 master -> master
PS E:\CESA\Puesta en producción\Codigo\S2R3\S2R2-remoto> cat .\directores.txt
Martin Scorsese
Ridley Scott
Stanley Kubrick
Steven Spielberg
Ingmar Bergman
```

5. Vuelve de nuevo a la carpeta `S2R2` y realiza las siguientes acciones:

- a) Obtén los cambios que hay en el repositorio remoto **sin fusionarlos en la rama local**. Adjunta captura de pantalla del resultado del comando utilizado.

```
PS E:\CESA\Puesta en producción\Codigo\S2R2\S2R2-remoto> git fetch
From E:/CESA/Puesta en producción/Codigo/S2R2/./S2R2-remoto
 81c1d5d..7096461 master -> origin/master
```

- b) Actualiza la rama `master` local con el contenido de la rama `master` del repositorio remoto. Adjunta captura de pantalla del resultado del comando utilizado.

```
PS E:\CESA\Puesta en producción\Codigo\S2R2\S2R2-remoto> git pull
Already up to date.
```

- c) Comprueba que aparecen los 5 nombres de directores esperados.

```
PS E:\CESA\Puesta en producción\Codigo\S2R2\S2R2-remoto> cat .\directores.txt
Martin Scorsese
Ridley Scott
Stanley Kubrick
Steven Spielberg
Ingmar Bergman
```

## 4. GitHub

1. Crea una cuenta en GitHub
2. Añade tu dirección de correo de educación
3. Solicita un descuento para uso educativo: <https://education.github.com/pack>
4. Haz un *fork* del repositorio localizado en la siguiente url: <https://github.com/curso-github-cefire/sesion3-practica>. A partir de este momento todas las tareas que se indican se deben realizar **en tu repositorio** (el que has clonado mediante el *fork*).
  - a) Realiza un primer commit para poner tu nombre y apellidos en el fichero `README.md`
  - b) Crea 3 issues con los siguientes títulos. Si no ves la pestaña de *issues*, actívala desde los ajustes (*settings*) del repositorio.
    - Añadir 3 libros
    - Añadir 3 películas
    - Añadir 3 discos
  - c) Crea una *milestone* denominada `Tareas sesión 3-2` que contenga los 3 issues creados.

- d) Modifica los ficheros correspondientes y realiza 3 commits para realizar cada una de las tareas que se indican en los issues. El mensaje del commit debe hacer que se cierren los issues correspondientes de manera automática.
- e) Haz una captura de pantalla de los comandos que has utilizado para hacer los commits y subir los cambios a GitHub.

Commit changes

Commit message

Update discos.txt

Extended description

close #1

☒ Commit directly to the master branch

☐ Create a new branch for this commit and start a pull request

[Learn more about pull requests](#)

Cancel

Commit changes

- f) Incluye las capturas de pantalla en el repositorio dentro de la carpeta capturas. Añádelas también al repositorio de manera que queden guardadas en tu repositorio en GitHub.
- g) Realiza una `pull request` indicando en el mensaje que has completado la tarea.

## 5. Entrega de la tarea

La entrega de la tarea se realiza a través de la `pull request` en GitHub. No olvides incluir en la carpeta `capturas` todas las capturas de pantalla realizadas en la práctica (Fundamentos de Git I, Fundamentos de Git II y GitHub).