

Ejercicio Git

David Llinares Escrivá

Fundamentos git 1

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`.

```
david@DESKTOP-DOI2K4D MINGW64 ~  
$ git --version  
git version 2.46.2.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.

```
david@DESKTOP-DOI2K4D MINGW64 ~  
$ git config --global user.name "David Llinares"
```

```
david@DESKTOP-DOI2K4D MINGW64 ~  
$ git config --global user.email davlliesc@alu.edu.gva.es  
  
david@DESKTOP-DOI2K4D MINGW64 ~  
$ git config --global core.editor "code --wait"  
  
david@DESKTOP-DOI2K4D MINGW64 ~  
$ git config --global -e
```

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`.

```

david@DESKTOP-DOI2K4D MINGW64 /f/DavidDocs/Ciberseguridad/Puesta en producción/
it/S1R1 (master)
$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
      películas.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 películas.txt y comidas.txt.
- j) Elimina el archivo comidas.txt desde el navegador de archivos.
- k) Añade dos películas más al archivo películas.txt.
- l) Haz una captura de pantalla que muestre los cambios en el directorio de trabajo.

```

david@DESKTOP-DOI2K4D MINGW64 /f/DavidDocs/Ciberseguridad/Puesta en producción/
it/S1R1 (master)
$ 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:   películas.txt

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

```

- 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 películas.txt.

```
david@DESKTOP-DOI2K4D MINGW64 /f/DavidDocs/Ciberseguridad/Puesta en producción
git/S1R1 (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   peliculas.txt

Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        deleted:    comidas.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).

```
david@DESKTOP-DOI2K4D MINGW64 /f/DavidDocs/Ciberseguridad/Puesta en producción
git/S1R1 (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        deleted:    comidas.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        datos.bak
        output/

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

r) Crea un archivo .gitignore para que los ficheros con extensión .bak y el contenido de la carpeta output/ no se incluyan en el

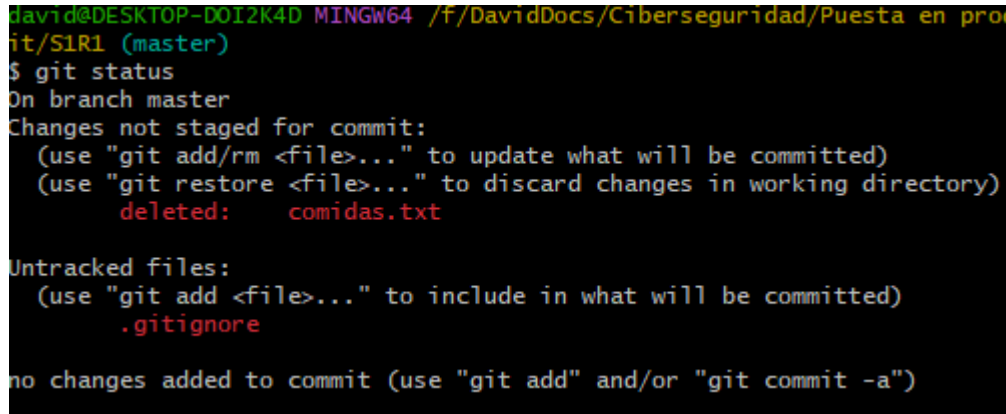
repositorio.

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.



```
david@DESKTOP-DO12K4D MINGW64 /f/DavidDocs/Ciberseguridad/Puesta en pro
it/S1R1 (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        deleted:    comidas.txt

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

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

t) Haz un último commit para incluir el archivo .gitignore en el re-

positorio.

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

repositorio.

```
commit f14b700f1dfc6d99b32e01507a257dfda67f71e2 (HEAD -> master)
Author: David Llinares <davlliesc@alu.edu.gva.es>
Date: Fri Oct 4 13:39:18 2024 +0200

    Añadido .gitignore y eliminado comidas

commit 9a0415e7bdef716af08601747ddf1da221f6ec9f
Author: David Llinares <davlliesc@alu.edu.gva.es>
Date: Fri Oct 4 13:31:50 2024 +0200

    Eliminado comidas y modificado peliculas

commit b008dce10a56d5cf943d47123f80145adf41f7a1
Author: David Llinares <davlliesc@alu.edu.gva.es>
Date: Fri Oct 4 13:19:55 2024 +0200

    Añadidas peliculas y comidas

commit 027182bc7218cb9c338db76e2239e81744936e01
Author: David Llinares <davlliesc@alu.edu.gva.es>
Date: Fri Oct 4 13:13:03 2024 +0200

    Añadir 2 títulos a libros

commit 5131c8d8229775f549ac53ba897e73ec4c27ea1b
Author: David Llinares <davlliesc@alu.edu.gva.es>
Date: Fri Oct 4 13:11:24 2024 +0200

    Añadir libros.txt
```

Fundamentos git 2

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.

```
david@DESKTOP-DOI2K4D MINGW64 /f/DavidDocs/Ciberseguridad/Puesta e
it/S2R1 (test)
$ git log --graph --all
* commit b22efe28df1f9e46fc6bdb5be99632517192db91 (HEAD -> test)
| Author: David Llinares <davlliesc@alu.edu.gva.es>
| Date: Fri Oct 4 15:47:53 2024 +0200
|
| Añadido actrices.txt
|
* commit 69d571b51a839f30a2b9a2484911919284234592 (master)
| Author: David Llinares <davlliesc@alu.edu.gva.es>
| Date: Fri Oct 4 15:42:24 2024 +0200
|
| Actores.txt añadidos
```

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.

```
david@DESKTOP-DOI2K4D MINGW64 /f/Davi
it/S2R1 (master)
$ git merge test
Updating 69d571b..b22efe2
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 0cb9352e76d90572e6ad6c2dbcf1413e2ee05a1a (HEAD -> test2)
Author: David Llinares <davlliesc@alu.edu.gva.es>
Date: Fri Oct 4 16:59:15 2024 +0200

    Añadida una actriz a actrices en test 2

* commit 26bb08b5d21940e35b26de1b2f4af14f7532418d (master)
/ Author: David Llinares <davlliesc@alu.edu.gva.es>
Date: Fri Oct 4 16:57:33 2024 +0200

    Añadida nueva actriz a actrices

* commit b22efe28df1f9e46fc6bdb5be99632517192db91 (test)
Author: David Llinares <davlliesc@alu.edu.gva.es>
Date: Fri Oct 4 15:47:53 2024 +0200

    Añadido actrices.txt

* commit 69d571b51a839f30a2b9a2484911919284234592
Author: David Llinares <davlliesc@alu.edu.gva.es>
Date: Fri Oct 4 15:42:24 2024 +0200

    Actores.txt añadidos

```

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.

```

david@DESKTOP-DOI2K4D MINGW64 /f/DavidDocs/Ciberseguridad/Puesta e
it/S2R1 (master|MERGING)
$ 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.

```

david@DESKTOP-DOI2K4D MINGW64 /f/DavidDocs/Ciberseguridad/Puesta en producción/C
it/S2R1 (master)
$ git log --graph --all
*   commit eaa16a94b28b8a76b73568d61201cabcd17b0c7f (HEAD -> master)
|   Merge: 26bb08b 0cb9352
|   Author: David Llinares <davlliesc@alu.edu.gva.es>
|   Date:   Fri Oct 4 17:03:12 2024 +0200
|
|   Conflicto resuelto
|
*   commit 0cb9352e76d90572e6ad6c2dbcf1413e2ee05a1a (test2)
|   Author: David Llinares <davlliesc@alu.edu.gva.es>
|   Date:   Fri Oct 4 16:59:15 2024 +0200
|
|   Añadida una actriz a actrices en test 2
|
*   commit 26bb08b5d21940e35b26de1b2f4af14f7532418d
|   Author: David Llinares <davlliesc@alu.edu.gva.es>
|   Date:   Fri Oct 4 16:57:33 2024 +0200
|
|   Añadida nueva actriz a actrices
|
*   commit b22efe28df1f9e46fc6bdb5be99632517192db91 (test)
|   Author: David Llinares <davlliesc@alu.edu.gva.es>
|   Date:   Fri Oct 4 15:47:53 2024 +0200
|
|   ...skipping...
*   commit eaa16a94b28b8a76b73568d61201cabcd17b0c7f (HEAD -> master)
|   Merge: 26bb08b 0cb9352
|   Author: David Llinares <davlliesc@alu.edu.gva.es>
|   Date:   Fri Oct 4 17:03:12 2024 +0200
|
|   Conflicto resuelto
|
*   commit 0cb9352e76d90572e6ad6c2dbcf1413e2ee05a1a (test2)
|   Author: David Llinares <davlliesc@alu.edu.gva.es>
|   Date:   Fri Oct 4 16:59:15 2024 +0200
|
|   Añadida una actriz a actrices en test 2
|
*   commit 26bb08b5d21940e35b26de1b2f4af14f7532418d
|   Author: David Llinares <davlliesc@alu.edu.gva.es>
|   Date:   Fri Oct 4 16:57:33 2024 +0200
|
|   Añadida nueva actriz a actrices
|
*   commit b22efe28df1f9e46fc6bdb5be99632517192db91 (test)
|   Author: David Llinares <davlliesc@alu.edu.gva.es>
|   Date:   Fri Oct 4 15:47:53 2024 +0200
|
|   Añadido actrices.txt
|
*   commit 69d571b51a839f30a2b9a2484911919284234592
|   Author: David Llinares <davlliesc@alu.edu.gva.es>
|   Date:   Fri Oct 4 15:42:24 2024 +0200
|
|   Actores.txt añadidos

```

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.


```
david@DESKTOP-DOI2K4D MINGW64 /F/DavidDocs/C
it
$ mkdir S2R2-remoto

david@DESKTOP-DOI2K4D MINGW64 /F/DavidDocs/C
it
$ cd S2R2-remoto/

david@DESKTOP-DOI2K4D MINGW64 /F/DavidDocs/C
it/S2R2-remoto
$ git init
Initialized empty Git repository in F:/David
ción/Git/S2R2-remoto/.git/
```

3. Clona el repositorio S2R2-remoto en una carpeta denominada S2R2.

Adjunta captura de pantalla del resultado del comando de clonado.

```
david@DESKTOP-DOI2K4D MINGW64 /F/DavidDocs/Ciberseguridad
it
$ git clone ./S2R2-remoto/ S2R2
Cloning into 'S2R2'...
warning: You appear to have cloned an empty repository.
done.
```

A continuación realiza las siguientes acciones en el repositorio S2R2:

- 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.

```
david@DESKTOP-DOI2K4D MINGW64 /f/DavidDocs/Ciberseguridad/Puesta en producción/Git/S2R2 (master)
$ git push origin master
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 295 bytes | 295.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote: error: refusing to update checked out branch: refs/heads/master
remote: error: By default, updating the current branch in a non-bare repository
remote: is denied, because it will make the index and work tree inconsistent
remote: with what you pushed, and will require 'git reset --hard' to match
remote: the work tree to HEAD.
remote:
remote: You can set the 'receive.denyCurrentBranch' configuration variable
remote: to 'ignore' or 'warn' in the remote repository to allow pushing into
remote: its current branch; however, this is not recommended unless you
remote: arranged to update its work tree to match what you pushed in some
remote: other way.
remote:
remote: To squelch this message and still keep the default behaviour, set
remote: 'receive.denyCurrentBranch' configuration variable to 'refuse'.
To F:/DavidDocs/Ciberseguridad/Puesta en producción/Git/./S2R2-remoto/
! [remote rejected] master -> master (branch is currently checked out)
error: failed to push some refs to 'F:/DavidDocs/Ciberseguridad/Puesta en producción/Git/./S2R2-remoto/'
```

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.

g) 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.

```
david@DESKTOP-DOI2K4D MINGW64 /f/DavidDocs/Ciberseguridad/Puesta en producción/Git/S2R2 (version1)
$ git push -u origin version1
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 4 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (6/6), 550 bytes | 550.00 KiB/s, done.
Total 6 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
To F:/DavidDocs/Ciberseguridad/Puesta en producción/Git/./S2R2-remoto/
* [new branch]      version1 -> version1
branch 'version1' set up to track 'origin/version1'.
```

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 es-

te 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.

```
david@DESKTOP-DOI2K4D MINGW64 /f/DavidDocs/Cibersegur
it/sesion3-practica (master)
$ git commit -m "Añadidos 3 libros. Close #1"
[master 4dde6a6] Añadidos 3 libros. Close #1
1 file changed, 3 insertions(+), 3 deletions(-)
```

```
david@DESKTOP-DOI2K4D MINGW64 /f/DavidDocs/Ciberseg
it/sesion3-practica (master)
$ git commit -m "Añadidas 3 películas. Close #2"
[master 5ec7683] Añadidas 3 películas. Close #2
1 file changed, 3 insertions(+), 3 deletions(-)
```

```
david@DESKTOP-DOI2K4D MINGW64 /f/DavidDocs/Ciberseg
it/sesion3-practica (master)
$ git commit -m "Añadidos los discos. Close #3"
[master 0c1399d] Añadidos los discos. Close #3
1 file changed, 3 insertions(+), 3 deletions(-)
```

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 com-

pletado la tarea.

5. Entrega de la tarea

Crea un documento con las capturas pedidas (Fundamentos de Git I, Fundamentos de Git II y GitHub) y guárdalo en formato PDF.

Guarda dicho documento en la carpeta capturas del repositorio y sube los

cambios a GitHub. La entrega de la tarea se realiza a través de la pull request

en GitHub.