Alumna: BLANCA ARIAS SAEZ GRUPO Y

Práctica 1 - Uso de Git y GitHub

## PROFESOR: Pedro Prieto Alarcón 16/09/2022

**Índice**

1. [Preparación](#_bookmark0) 2
2. [Fundamentos de Git I](#_bookmark1) 2
3. [Fundamentos de Git II](#_bookmark2) 4
4. [GitHub](#_bookmark3) 6
5. [Entrega de la tarea](#_bookmark4) 7

1

# Preparación

Realiza las tareas que se indican a continuación. Incluye las capturas de pan- talla 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 instruccio- nes. 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](https://code.visualstudio.com/)

[Atom](https://atom.io/)

[Brackets](http://brackets.io/)

[Sublime Text](https://www.sublimetext.com/)

 [Notepad ++](https://notepad-plus-plus.org/)

El bloc de notas de Windows utiliza una codificación de caracteres específi- ca 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.

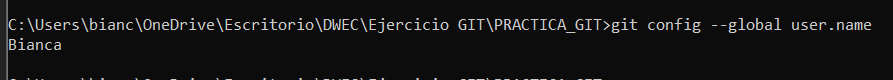
# Fundamentos de Git I

* 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.

Texto

Descripción generada automáticamente

* 1. 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.





* 1. Crea una carpeta denominada S1R1. Realiza las siguientes acciones en ella:
     1. Crea un repositorio Git.
     2. Crea un fichero denominado libros.txt. Añade tres títulos de li- bros cada uno en una línea distinta.
     3. Haz un primer *commit*.
     4. Añade dos libros al archivo libros.txt.
     5. Haz un segundo *commit*.

*f* ) Crea un fichero denominado peliculas.txt. Añade tres títulos de películas a dicho archivo.

1. Haz una captura de pantalla del comando git status.

Texto

Descripción generada automáticamente

1. Crea un fichero denominado comidas.txt. Añade tres nombres de comidas a dicho archivo.

Captura de pantalla de un celular

Descripción generada automáticamente

1. Haz un tercer *commit* que incluya los archivos peliculas.txt y

comidas.txt.

Texto

Descripción generada automáticamente

1. Elimina el archivo comidas.txt desde el navegador de archivos.
2. Añade dos películas más al archivo peliculas.txt.
3. Haz una captura de pantalla que muestre los cambios en el directorio de trabajo.

Texto

Descripción generada automáticamente

1. Añade los cambios al área de preparación.

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

1. 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.

Texto

Descripción generada automáticamente

*ñ*) Haz un cuarto *commit*.

Texto

Descripción generada automáticamente

1. Crea un archivo denominado datos.bak. Añade tres títulos de li- bros a dicho archivo. ¡IMPORTANTE! No añadas el archivo al área de preparación ni hagas ningún commit.
2. Crea una subcarpeta denominada output. Crea un archivo denomi- nado 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.

Texto

Descripción generada automáticamente

1. Haz una captura de pantalla del comando git status. Deben apa- recer los archivos datos.bak y output/salida.txt como ar- chivos nuevos (color rojo).

Texto

Descripción generada automáticamente

1. Crea un archivo .gitignore para que los ficheros con extensión

.bak y el contenido de la carpeta output/ no se incluyan en el repositorio.

1. 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.

Texto

Descripción generada automáticamente

1. Haz un último *commit* para incluir el archivo .gitignore en el re- positorio.
2. Haz una captura de pantalla que muestre el histórico de cambios del repositorio.

Texto

Descripción generada automáticamente

# Fundamentos de Git II

* 1. Crea una carpeta denominada S2R1. Realiza las siguientes acciones en ella:
     1. Crea un repositorio Git.
     2. Crea un fichero denominado actores.txt. Añade tres nombres de actores cada uno en una línea distinta.
     3. Haz un primer *commit*.
     4. Crea una rama denominada test.
     5. Cambia a la rama test

Texto

Descripción generada automáticamente

*f* ) En la rama test crea un fichero denominado actrices.txt. Aña- de tres nombres de actrices y realiza un *commit* en dicha rama.

1. Haz una captura de pantalla del resultado del comando git log

--graph --all.

Texto

Descripción generada automáticamente

1. Cambia a la rama master.

Texto

Descripción generada automáticamente

Captura de pantalla con letras

Descripción generada automáticamente

1. 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 resulta- do.

Texto

Descripción generada automáticamente

1. Crea una segunda rama denominada test2. La rama test2 apunta al mismo *commit* que la rama master en este momento.

Captura de pantalla de un celular

Descripción generada automáticamente

1. En la rama master, añade una actriz al fichero actrices.txt y haz un *commit*.

Texto

Descripción generada automáticamente

1. Cambia a la rama test2
2. En la rama test2, añade una actriz al fichero actrices.txt y haz otro *commit*.
3. 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.

Texto

Descripción generada automáticamente

*ñ*) Cambia a la rama master

1. 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.

Texto

Descripción generada automáticamente

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza media

1. Resuelve el conflicto incorporando los dos nombres de actrices.

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

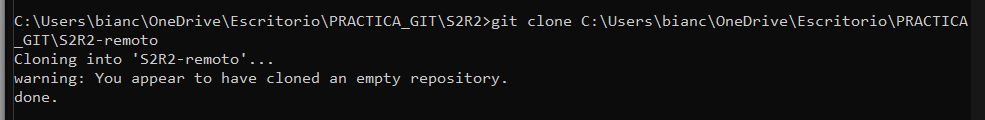
1. Haz una captura de pantalla del resultado del comando git log

--graph --all. Observa que se ha creado un nuevo *commit* que integra los dos caminos anteriores.

Texto

Descripción generada automáticamente

* 1. 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.
  2. Clona el repositorio S2R2-remoto en una carpeta denominada S2R2. Adjunta captura de pantalla del resultado del comando de clonado. A conti- nuación realiza las siguientes acciones en el repositorio S2R2:



* + 1. Crea un archivo denominado directores.txt. Añade el nombre de tres directores de cine.
    2. Haz un *commit*.
    3. Realiza un *push* al repositorio remoto. Adjunta captura de pantalla del resultado.

Texto

Descripción generada automáticamente

* + 1. Crea una rama denominada version1.
    2. 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.

* + 1. 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: git push --set-upstream origin version1

Captura de pantalla de un celular

Descripción generada automáticamente

Texto

Descripción generada automáticamente

* 1. Clona el repositorio S2R2-remoto en una segunda carpeta denominada

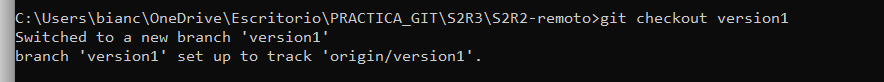
S2R3. Realiza las siguientes acciones sobre ella:

* + 1. Muestra en la consola el contenido del fichero directores.txt y el resultado del comando git status. Debe mostrar tres directores.

Texto

Descripción generada automáticamente

* + 1. Cambia a la rama version1. Muestra el resultado del comando. Comprueba que se crea una rama local version1 con el conteni- do de la rama remota origin/version1 y enlazada con ella. Al clonar el repositorio la rama no existía (solo se clona la rama princi- pal, master), pero al cambiar a una rama que existe en el remoto se produce su creación local y enlazado con su correspondiente remota.



* + 1. Muestra el contenido del fichero directores.txt por la pantalla. Comprueba que se muestran los 5 nombres de directores esperados. Adjunta captura de pantalla.

Texto

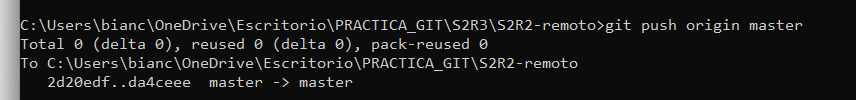
Descripción generada automáticamente

* + 1. Cambia a la rama master.
    2. Incorpora los cambios de la rama version1 a la rama master.

Texto

Descripción generada automáticamente

*f* ) Sube la rama master actualizada al servidor. Adjunta captura de pan- talla del resultado del comando.



* 1. Vuelve de nuevo a la carpeta S2R2 y realiza las siguientes acciones:
     1. Obtén los cambios que hay en el repositorio remoto sin fusionarlos en la rama local. Adjunta captura de pantalla del resultado del comando utilizado.

Texto

Descripción generada automáticamente

* + 1. Actualiza la rama master local con el contenido de la rama master del repositorio remoto. Adjunta captura de pantalla del resultado del comando utilizado.

Remoto: Texto

Descripción generada automáticamente

Local:

Captura de pantalla de un celular

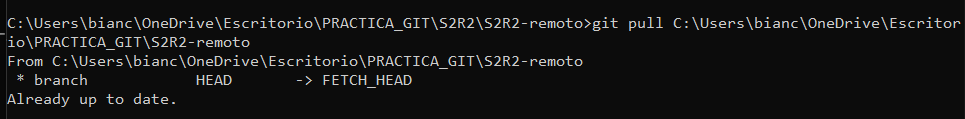
Descripción generada automáticamente

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente



* + 1. Comprueba que aparecen los 5 nombres de directores esperados.

Texto

Descripción generada automáticamente

# 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.](https://education.github.com/pack) [com/pack](https://education.github.com/pack)
  4. Haz un *fork* del repositorio localizado en la siguiente url: [https://github.](https://github.com/curso-github-cefire/sesion3-practica) [com/curso-github-cefire/sesion3-practica](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 reposi- torio (el que has clonado mediante el *fork*).

* + 1. Realiza un primer commit para poner tu nombre y apellidos en el fi- chero README.md
    2. 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

* + 1. Crea una milestone denominada Tareas sesión 3-2 que con- tenga los 3 issues creados.
    2. Modifica los ficheros correspondientes y realiza 3 commits para rea- lizar 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.
    3. Haz una captura de pantalla de los comandos que has utilizado para hacer los commits y subir los cambios a GitHub.

*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.

# 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).