

Práctica - Uso de Git y GitHub

Pedro Prieto Alarcón

30/09/2024

Índice

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

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`.
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.
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`.
- 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.
- 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`.
- 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).
- 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`.
- 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.

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`.
- 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.
- 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`.
- 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`.

- 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.
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`:
- 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.
 - 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.
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.
 - 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.

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

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.
- 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

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.