I.E.S. Punta del Verde

2024/2025

Junta de Andalucía
Consejería de Educación y Deporte

Unión Europea
Fondo Social Europeo
"El FSE invierte en tu futuro"

GOBIERNO DE ESPAÑA
MINISTERIO DE EDUCACIÓN Y FORMACIÓN PROFESIONAL

Introducción Git local

Dos primeros capítulos página oficial de git:

1. Añade tus credenciales en git estableciendo tu nombre de usuario y dirección de correo electrónico. Esto es importante porque las confirmaciones de cambios (commits) en Git usan esta información, y es introducida de manera inmutable en los commits que envías:

```
$ git config --global user.name "John Doe"

$ git config --global user.email johndoe@example.com
```

Creación y trabajo con repositorio

2. Crea un directorio en tu usuario llamado workspace.

3. Dentro de workspace crea un directorio DAW_tunombre_tarea1. Añade captura de pantalla. Haz que este directorio pase a ser tu repositorio. ¿Qué significa que un directorio sea repositorio?

4. Configura mediante el archivo .gitignore que se ignore todo archivo ".txt" y añade un README.md que contenga "Tarea 1 de Git".

5. Sube el estado actual del repositorio con un commit cuyo mensaje sea "initial commit"

6. Añade un fichero "index.html" en DAW_tunombre_tarea1 que contenga un título <h1> "Hola mundo" </h1> y un fichero .txt con algún texto.

7. Ejecuta los comandos necesarios para añadir los cambios al repositorio y subirlos (no olvides al subir los cambios añadir un comentario de los cambios realizados).

8. Modifica el fichero html anterior añadiendo un párrafo con "Modificación 1"

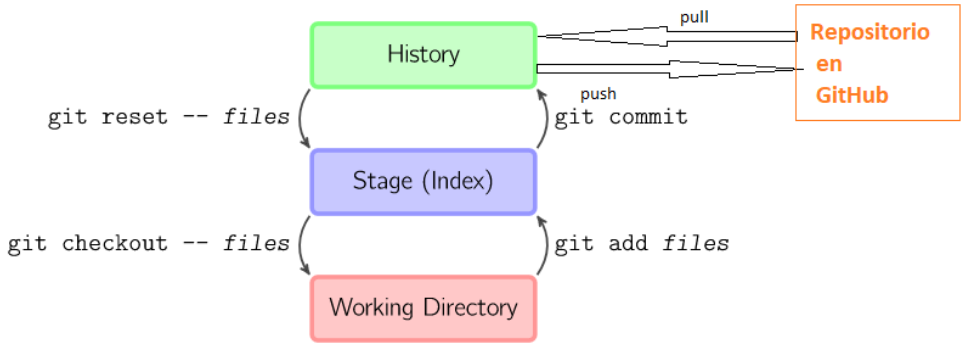
9. Visualiza los cambios que se han producido en tu directorio de trabajo con respecto al repositorio local y posteriormente, sube los cambios al repositorio añadiendo el comentario "Modificación primera realizada"

10. Muestra el histórico de confirmaciones de git.

Deshacer cambios

11. Eliminar la última línea del fichero index.html y guardarlo.
12. Comprobar el estado del repositorio.
13. Deshacer los cambios realizados en el fichero para volver a la versión anterior del fichero.
14. Volver a comprobar el estado del repositorio.
15. Eliminar la última línea del fichero index.html y guardarlo. Añade este cambio a stage.
16. Comprobar de nuevo el estado del repositorio.
17. Quitar los cambios de la zona de intercambio temporal, pero mantenerlos en el directorio de trabajo.
18. Comprobar de nuevo el estado del repositorio.
19. Realiza múltiples cambios (añade ficheros, modifica, directorios, lo que quieras). Añade estos cambios (stage) y utiliza vuelve al estado inicial del commit actual.
20. Vuelve al estado del primer commit y posteriormente al vuelve al último
21. Realiza un revert de los cambios de un commit.

Junta de Andalucía
Consejería de Educación y Deporte
Unión Europea
Fondo Social Europeo
"El FSE invierte en tu futuro"
GOBIERNO DE ESPAÑA
MINISTERIO DE EDUCACIÓN Y FORMACIÓN PROFESIONAL



Los cuatro comandos de arriba copian archivos entre el directorio de trabajo, el stage (también llamado index), y la historia (en la forma de commits).

- `git add archivos` copia *archivos* (en su estado actual) al stage.
- `git commit` guarda una snapshot del stage como un commit.
- `git reset -- archivos` quita *archivos* de stage; esto es, que copia *archivos* del último commit al stage. Usá este comando para "deshacer" un `git add archivos`. También podés usar `git reset` para quitar de stage todo lo que hayas agregado.
- `git checkout -- archivos` copia *archivos* desde el stage al directorio de trabajo. Usá esto para descartar los cambios locales.