



Práctica.GIT

1.-Introducción

Git es un software de control de versiones diseñado por Linus Torvalds, pensando en la eficiencia y la confiabilidad del mantenimiento de versiones de aplicaciones cuando éstas tienen un gran número de archivos de código fuente. Su propósito es llevar registro de los cambios en archivos de computadora y coordinar el trabajo que varias personas realizan sobre archivos compartidos.

2.-Objetivos

- Familiarizarse con el entorno de trabajo GIT.
- Gestionar tu propio repositorio.

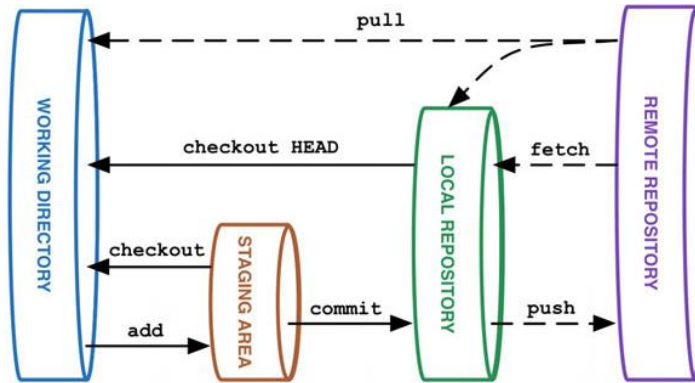
3.-Ejercicios

3.1. Inicialización

1. Dirígete a [GitHub](https://github.com) y créate una cuenta.
2. Crea un repositorio local con un nombre similar al siguiente: FP_ESPECIALIDAD_AÑO.
3. Añade un directorio por módulo que tengas.
4. Añade todo el temario que lleves hasta este momento, **no utilices la información original, cópiala.**
5. Almacena y/o indica la secuencia de pasos que has realizados.

3.2 GIT Bash

Utilizando el terminal de comandos con el programa GIT realiza el siguiente ejercicio. Escribe los comandos que has necesitado introducir en el terminal para llevar a cabo el ejercicio. Busca las siguientes vistas. Realiza una captura de pantalla de cada una y sitúala en los recuadros.



1. Crea 3 archivos con cualquier contenido: archivo1.txt, archivo2.txt y archivo3.txt. Estos archivos están contenidos en sus respectivos directorios directorio1, directorio2 y directorio3.
2. Inicializa un repositorio local
3. Añade al stage los archivos de los directorios 1 y 2.
4. Comprueba con `git status` que el stage tiene lo que quieres.
5. Crea un commit con esos dos archivos.
6. Realiza una modificación en el texto del archivo 1/1.txt y otra en el texto del archivo 2/2.txt.
7. Añade al stage únicamente las modificaciones del archivo 2/2.txt.
8. Haz `git status` e interpreta la información que te da.
9. Crea un commit con las modificaciones que has hecho en el archivo 2/2.txt y con el archivo 3/3.txt.
10. Haz `git status` y comprueba que el stage está vacío y que en el working directory está el cambio del archivo 1/1.txt.
11. Desversiona el archivo 1/1.txt (es decir, dile a Git que no lo versione), pero sin eliminarlo.
12. Elimina el archivo 2/2.txt.
13. Haz `git status` y comprueba que la eliminación de los dos archivos está en el stage.
14. Crea un commit que almacene la eliminación de esos dos últimos archivos.
15. Haz `git status` y comprueba que todo está como esperas (el archivo 1/1.txt está sin versionar y el stage está sin cambios).
16. Configura el repositorio para que ignore la carpeta 1.
17. Haz `git status` y comprueba la diferencia.