# Git. actividad 1

**Instrucciones:** Ejecutar los comandos correspondientes de git para resolver los puntos del enunciado

# Parte 1

1. Crear una carpeta *MiWeb* con un archivo *index.html* con un código estándar, un archivo *styles.css* que dé un color de fondo a la web y un archivo *info.txt* con los nombres de los compañeros del grupo.
2. Iniciar un repositorio en la carpeta.
3. Crear un archivo *.gitignore* que excluya del repositorio el archivo *info.txt.*
4. Agregar al repositorio todos los archivos.
5. Excluir del repositorio el archivo *styles.css*.
6. Ver el estado del repositorio (el archivo de estilos ya no debe figurar como parte del repositorio)
7. Agregar al repositorio el archivo *styles.css*.
8. Ver el estado del repositorio (el archivo de estilos debe figurar como parte del repositorio)
9. Hacer un primer commit del repositorio.
10. Eliminar el archivo *index.html* y cambiar el color de fondo de la regla de estilos de *styles.css.*
11. Deshacer estos cambios.
12. Crear un archivo *index.js* con un código para que al cargarse la web aparezca un mensaje. Agregar el archivo *index.html* la etiqueta *script* con el objetivo de vincular el archivo JavaScript.
13. Agregar el archivo JavaScript al repositorio y hacer un segundo commit con estos cambios.
14. Ver todos los commits hechos en el repositorio.
15. Modificar el mensaje del archivo JavaScript.
16. Utilizar el comando git necesario para ver las diferencias entre el último commit y el estado actual del repositorio.

# Parte 2

1. Crear 3 archivos html y por cada uno hacer un commit
2. Hacer un reset soft al penúltimo commit.
3. Volver a hacer el commit.
4. Hacer un reset mixed al penúltimo commit.
5. Ver el estatus del repositorio y comprobar el resultado.
6. Hacer un reset hard al primer commit.
7. Volver al commit que recupera todos los archivos.

# Parte 3

1. Crear una rama llamada *imagenes*.
2. Moverse a esa rama.
3. Crear una carpeta con el nombre *images*, incluir en la carpeta algunas imágenes. Modificar el archivo *index.html* incluyendo alguna imagen.
4. Hacer un commit con los dos cambios.
5. Moverse a la rama master.
6. Cambiar el archivo *index.html*
7. Fusionar los cambios incorporando ambos cambios en master.
8. Eliminar la rama imágenes.
9. Etiquetar el último commit como v.1.0
10. Crear una rama con el nombre *estilos.*
11. Moverse a la rama y crear una carpeta con el nombre *css* y un archivo de estilos.
12. Hacer un commit con los cambios.
13. Volver a la rama *master.*
14. Crear un archivo *about.html.* Hacer un commit para incluir el archivo.
15. Incluir este archivo en la rama *estilos* mediante *rebase.*
16. Volver a la rama *master*. Crear el archivo *login.html* y hacer un commit.
17. Crear el archivo *register.html* y hacer un commit.
18. Incluir este último commit en la rama *estilos* mediante c*herry pick.*
19. Volver a la rama *master,* cambiar el archivo *index.html* y hacer un commit.
20. Moverse a la rama *estilos* y cambiar el *index.html* para incluir el link al archivo css.
21. Fusionar los cambios y gestionar los conflictos.
22. Eliminar la rama *estilos.*

# Parte 4

1. Crear un repositorio en GitHub con el nombre *PrimerRepoGithub*
2. Clonarlo en una carpeta de nuestro repositorio local.
3. Crear un archivo *descripción.txt* y hacer un commit. Hacer el push al repositorio de GitHub.
4. Comprobar los cambios en la web del repositorio.
5. Crear una rama con el nombre *RamaDesarrolloFront* y movernos a ella.
6. Crear un archivo *index.html* y hacer un commit. Hacer un push de la rama al repositorio de GiHub.
7. Comprobar la existencia de la rama y su contenido en la web del repositorio.
8. Hacer el *merge* de la rama *RamaDesarrolloFront* a la rama *master*. Hacer un push de la rama al repositorio de GiHub.
9. Eliminar la rama *RamaDesarrolloFront.* Hacer un push de la rama al repositorio de GiHuby comprobar el resultado en la web del repositorio.
10. Etiquetar la versión actual de la rama main y actualizar la etiqueta en Github.

# Parte 5

1. Crear un repositorio en GitHub con el nombre *ContenidosCurso*
2. Invitar a los compañeros del grupo a formar parte del repositorio. Cada compañero deberá aceptar la invitación.
3. Clonar el repositorio (todos los miembros del grupo)
4. Cada compañero se encargará de incluir un contenido en el repositorio y de actualizarlo.
5. El resto de compañeros actualizarán el repositorio para mantener los cambios sincronizados.

**Objetivos:** Utilizar comandos de git para mantener un repositorio de código