

Ejercicio 1

1. Realizar un primer commit en el que añades dos ficheros .txt Uno con tu nombre y otro con el nombre actividades; por ejemplo (sabela.txt y actividades.txt). En el primero introduce una breve descripción de ti y en el segundo añade alguna de tus aficciones (leer, hacer deporte, etc.).

Lo primero es crear los archivos con:

```
nano dylan.txt
nano actividades.txt
```

Y le introducimos los datos que nos indican.

Antes de nada hay que instalar el GIT con (LINUX):

```
apt-get install git
```

Para empezar a realizar el seguimiento de los archivos usamos:

```
git init
git add dylan.txt
git add actividades.txt
```

Y ahora haremos el primer comentario:

```
git commit -m "Primer Commit De Prueba"
```

Resultado:

```
[master (commit-raíz) 1afd8aa] Primer Commit De Prueba
2 files changed, 2 insertions(+)
create mode 100644 actividades.txt
create mode 100644 dylan.txt
```

2. Realizar otro commit modificando el primer .txt añadiendo una nueva línea, comentando por qué estás en este ciclo.

Voy a modificar el archivo con nano y pondré la informacion que indica el ejercicio:

```
nano dylan.txt
```

Y hacer el commit:

```
git add Dylan.txt
git commit -m "Porque me meti al ciclo cambios Dylan.txt"
```

Resultado:

```
[master 22ce07b] Porque me meti al ciclo cambios Dylan.txt
1 file changed, 1 insertion(+), 1 deletion(-)
```

3. Crear una carpeta con dos ficheros, todo desde línea de comandos.

```
mkdir NuevaCarpeta
nano fichero1.txt
nano fichero.txt
```

4. Realizar otro commit con las siguientes modificaciones:

- Eliminar del segundo fichero una de tus aficciones
- Ignora la carpeta con los dos ficheros

Lo primero va a ser modificar el fichero actividades.txt: `bash nano actividades.txt`

Ahora vamos a crear el fichero que hace que ignore los archivos que le indiques: `bash nano .gitignore NuevaCarpeta/`

Y por ultimos el commit: `bash git add actividades.txt git commit -m "Commit gitIgnore"`

5. Realiza un checkout para volver a las primeras versiones de los ficheros .txt (el primer commit).

Lo primero es hacer un git log para ver los commits:

```
git log
```

Para volver al estado actual: `bash git checkout 22ce07b830`

Resultado: ““bash Nota: cambiando a ‘22ce07b8300’.

Te encuentras en estado ‘detached HEAD’. Puedes revisar por aquí, hacer cambios experimentales y hacer commits, y puedes descartar cualquier commit que hayas hecho en este estado sin impactar a tu rama realizando otro checkout. ““

Ejercicio2

1. Crear un repositorio nuevo (todo desde línea de comandos) con el nombre pagina_web y muestra su contenido desde línea de comandos.

Crear el nuevo repositorio:

```
mkdir pagina_web
```

2. Comprueba y explica el estado del repositorio.

Con ls comprobamos que está vacío el repositorio ya que lo acabamos de crear

3. Crear un fichero index.html con el siguiente contenido :

```
<html>
  <head>
    <title>Página de tu_nombre</title>
  </head>
  <body>
    Página en la que vamos a mostrar un listado de ciudades/países que visitar.
```

```
</body>
</html>
```

4. Realizar un commit con el mensaje “Primera página html”. Para hacer el primer comit hay que:

```
git init
git add index.html
git commit -m "Primera página html"
```

5. Muestra y explica el estado del repositorio.

```
git status
```

Me pone que no hay nada para hacer commit, el arbol de trabajo está limpio

6. Cambiar la página web para que muestre en un listado 3 ciudades que te gustaría visitar: Por ejemplo:

```
<html>
  <head>
    <title>Página de tu_nombre</title>
  </head>
  <body>
    Página en la que vamos a mostrar un listado de ciudades que visitar.
    <ul>
      <li>Oslo</li>
      <li>Venecia</li>
    </ul>
  </body>
</html>
```

7. Hacer un commit de los cambios, con el mensaje “Añadidas 3 ciudades que visitar”.

```
git add index.html
git commit -m "Añadidas 3 ciudades que visitar"
```

8. Muestra el historial de commits del repositorio.

```
git log
```

9. Crea una carpeta por cada ciudad que hayas indicado en el listado anterior. Introduce dentro de cada carpeta un fichero index.html con información por cada ciudad. Por ejemplo:

```
<html>
  <head>
    <title>Oslo</title>
  </head>
  <body>
    Oslo (Acerca de este sonido [ù lu] (?·i)), llamada Christiania de 1624 a 1897 y Kristian
```

```
</body>
</html>
```

10. Hacer un commit de los cambios, con el mensaje “Añadida información sobre las ciudades a visitar”.

```
git add .
git commit -m "Añadida informacion sobre ciudades"
```

11. Volver a mostrar el historial de cambios.

```
git log
```

Ejercicio 3. Fundamentos de GIT

1. Crear un repositorio nuevo con el nombre libro y mostrar su contenido. Comprueba el estado del repositorio.

```
mkdir libro
ls libro
git status
```

El status no va debido que aun no está inicializado el git

2. Crear un fichero indice.txt con el siguiente contenido: Capitulo 1: Introducción Capitulo 2: Los tres cerditos Capitulo 3: Caperucita roja
3. Realizar un commit con el mensaje “Añadido índice del libro”.

```
git init
git add index.html
git commit -m "Añadido índice del libro"
```

4. Comprueba y explica el estado del repositorio.

```
git add index.html
git status
```

Me pone que no hay nada para hacer commit, el arbol de trabajo está limpio

5. Cambiar el índice.txt para que contenga lo siguiente: Capitulo 1: Introducción Capitulo 2: Los tres cerditos Capitulo 3: Caperucita roja Capitulo 4: La bella y la bestia
6. Hacer un commit de los cambios con el mensaje “Añadido 4: La bella y la bestia”. Comprueba el estado del repositorio.

```
git add index.html
git commit -m "Añadido 4: La bella y la bestia"
```

7. Muestra el historial del repositorio.

```
git log
```

8. Crea la carpeta capítulos y dentro de ella el fichero capítulo2.txt con el siguiente texto: Y el lobo sopló y sopló y la casa derribó.

```
mkdir capitulos
cd capitulos
nano capítulo2.txt
```

9. Hacer un commit con el mensaje “Añadido capítulo 2”

```
git add .
git commit -m "Añadido capítulo 2"
```

10. Volver a mostrar el historial de cambios.

```
git add .
git log
```

11. Crear el fichero capítulo3.txt en la carpeta capitulos con el siguiente texto: Abuelita qué ojos más grandes tienes.

```
nano capítulo3.txt
```

12. Ver el estado del repositorio de forma abreviada e indicar qué significa cada letra.

```
git status
```

?? = Archivos sin seguimiento M = Archivos modificados

13. Modificar el índice.txt añadiendo “Capítulo 5: Forzen”

```
nano index.html
```

14. Subir los cambios al repositorio ingorando el capítulo3.txt

```
nano .gitignore
```

Dentro del gitignore pongo capítulo3.txt

```
git add .
git commit -m "Commit ignorando capítulo3.txt"
```

15. Modificar el fichero para que se ignoren todos aquellos ficheros que comiencen por _ a excepción del fichero __ayuda.txt (ya no se debe ignorar capítulo3.txt)

Añadimos esto al fichero .gitignore

```
_*
!_ayuda.txt
```

16. Crear un fichero __logs.txt con el siguiente contenido: Fichero para almacenar logs

```
nano __logs.txt
```

17. Crear un fichero __ayuda.txt con el siguiente contenido: Fichero de ayuda.

`nano _ayuda.txt`

18. Preparar todo con `git add *`. Explicad qué pasa.

`git add *`

Seleccionas todos los repositorios menos los que empiezan por `"_"`

19. Hacer un commit de los cambios con el mensaje "Añadido capítulo 2".
Comprobar/explicar qué se sube al repositorio.

`git commit -m "Añadido capítulo 2"`

Se añade el capítulo 2 y se ignoran los archivos que empiezan por `"_"`

20. Modificar el fichero `capitulo2.txt` (elimina lo que había antes). Caperucita iba por el bosque con su capa roja
21. Ver y explicar qué ha cambiado y aún no has preparado.
22. Hacer un commit con el mensaje "Capítulo 2 modificado"
23. Vuelve a modificar el capítulo con el siguiente contenido: Caperucita iba por el bosque con su capa roja cuando llegó a casa de su abuela le dijo "Abuela qué ojos más grandes tienes"

`git commit -m "Capítulo 2 modificado"`

24. Prepara tus cambios y comprueba qué ha cambiado con la última instantánea confirmada.

`git status`

25. Elimina del repositorio el fichero `_ayuda.txt`

`rm _ayuda.txt`

26. Cambia el nombre del fichero `indice.txt` por `indice_libros.txt` y sube los cambios.
27. Cambia el mensaje del último commit.