Ejercicio 1. Repositorio Local.

1. Crea una carpeta con el nombre repositorioLocal.
2. Inicializa un repositorio git. Comprueba que está inicializado.
3. 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.).
4. Realizar otro commit añadiendo una nueva línea a tu fichero comentando por qué estás en este ciclo.
5. Crear una carpeta (aux) con dos ficheros (a.txt y b.txt), todo desde línea de comandos.
6. Realizar otro commit con las siguientes modificaciones:
   - Eliminar del segundo fichero una de tus aficciones.
   - Ignora la carpeta con los dos ficheros.
7. Realiza un *checkout* para volver a las primeras versiones de los ficheros .txt (el primer commit).

```
# 1) Crear carpeta e inicializar repo
mkdir repositorioLocal && cd repositorioLocal
git init
git status

# 2) Primer commit con dos .txt
echo "Soy Luis Gómez Gil. Breve descripción sobre mí." > luis.txt
printf "leer\nhacer deporte\n" > actividades.txt
git add luis.txt actividades.txt
git commit -m "Primer commit: luis.txt y actividades.txt"

# 3) Segundo commit: nueva línea en tu fichero con por qué estás en el ciclo
echo "Estoy en este ciclo porque me apasiona el desarrollo web." >> luis.txt
git add luis.txt
git commit -m "Añadido motivo en luis.txt"

# 4) Crear carpeta aux con a.txt y b.txt
mkdir -p aux && : > aux/a.txt && : > aux/b.txt
git add aux
git commit -m "Añadida carpeta aux con a.txt y b.txt"

# 5) Otro commit: modificar actividades y empezar a ignorar la carpeta aux
# (elimina una afición del segundo fichero)
```

```
sed -i '1d' actividades.txt  # borra la primera línea (ajústalo si quieres borrar otra)
printf "/aux/\n" > .gitignore
git add actividades.txt .gitignore
git commit -m "Editar actividades (elimino una afición) e ignorar carpeta aux"


# (opcional) Confirma el estado y guarda la reversión en un commit
git add luis.txt actividades.txt
git commit -m "Revertir luis.txt y actividades.txt a su versión del primer commit"
```

Ejercicio 2. Trabajando con repositorios locales

1. Crear un repositorio nuevo (todo desde línea de comandos) con el nombre *pagina_web* y muestra su contenido desde línea de comandos.
2. Comprueba y explica el estado del repositorio.
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”.
5. Muestra y explica el estado del repositorio.
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”.
8. Muestra el historial de commits del repositorio.
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 [ùʃlu] (?·i)), llamada Christiania de 1624 a 1897 y Kristiania de 1897 a 1925 (Cristianía en español), es la capital y la ciudad más poblada de Noruega, además de ser su centro político, económico y cultural. Políticamente constituye un municipio y a la vez una de las diecinueve provincias del país. Según el censo del 21 de noviembre de 2018, su población era de 673 469 habitantes.2 Es la tercera ciudad y área urbana escandinava más poblada, solo superada por Copenhague y Estocolmo.

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

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

```
# 1) Crear repo pagina_web y mostrar su contenido
cd ..                           # vuelve a la carpeta superior
mkdir pagina_web && cd pagina_web
git init
ls -la                          # contenido desde línea de comandos
git status                        # estado del repo (untracked, etc.)

# 2) Crear index.html e inicial
cat > index.html <<'HTML'
<html>
```

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

git add index.html
git commit -m "Primera página html"
git status                          # explica: limpio, árbol en HEAD

# 3) Cambiar la página para listar 3 ciudades y hacer commit
cat > index.html <<'HTML'
<html>
  <head>
    <title>Página de Luis Gómez Gil</title>
  </head>
  <body>
    Página en la que vamos a mostrar un listado de ciudades que visitar.
    <ul>
      <li>Oslo</li>
      <li>Venecia</li>
      <li>Kioto</li>
    </ul>
  </body>
</html>
HTML

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

# 4) Historial de commits
git log --oneline --graph --decorate

# 5) Crear carpeta por ciudad con su index.html y hacer commit
mkdir -p Oslo Venecia Kioto

cat > Oslo/index.html <<'HTML'
<html>
  <head><title>Oslo</title></head>
  <body>
```

```
    Oslo, capital de Noruega, centro político, económico y cultural del país.
  </body>
</html>
HTML

cat > Venecia/index.html <<'HTML'
<html>
  <head><title>Venecia</title></head>
  <body>
    Venecia, ciudad italiana famosa por sus canales, arquitectura y arte.
  </body>
</html>
HTML

cat > Kioto/index.html <<'HTML'
<html>
  <head><title>Kioto</title></head>
  <body>
    Kioto, antigua capital de Japón, célebre por sus templos, jardines y tradición.
  </body>
</html>
HTML

git add Oslo Venecia Kioto
git commit -m "Añadida información sobre las ciudades a visitar"

# 6) Volver a mostrar historial
git log --oneline --graph –decorate
```

Ejercicio 3. Fundamentos de GIT

1. Crear un repositorio nuevo con el nombre libro y mostrar su contenido. Comprueba el estado del respositorio.
2. Crear un fichero indice.txt con el siguiente contenido:
3.

```
Capitulo 1: Introducción
Capitulo 2: Los tres cerditos
Capitulo 3: Caperucita roja
```

3. Realizar un commit con el mensaje “Añadido índice del libro”.
4. Comprueba y explica el estado del repositorio.
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.
7. Muestra el historial del repositorio.
8. Crea la carpeta capítulos y dentro de ella el fichero capitulo2.txt con el siguiente texto:

Y el lobo sopló y sopló y la casa derribó.

9. Hacer un commit con el mensaje “Añadido capitulo 2”
10. Volver a mostrar el historial de cambios.
11. Crear el fichero capitulo3.txt en la carpeta capitulos con el siguiente texto:

Abuelita qué ojos más grandes tienes.

12. Ver el estado del repositorio de forma abreviada e indicar qué significa cada letra.
13. Modificar el índice.txt añadiendo “Capitulo 5: Forzen””
14. Subir los cambios al repositorio ingorando el capitulo3.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 capitulo3.txt)
16. Crear un fichero _logs.txt con el siguiente contenido:

Fichero para almacenar logs

Fichero para almacenar logs

17. Crear un fichero _ayuda.txt con el siguiente contenido:

Fichero de ayuda.

18. Preparar todo con git add *. Explicad qué pasa.
19. Hacer un commit de los cambios con el mensaje “Añadido capitulo 2”. Comprobar/explicar qué se sube al repositorio.
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 “Capitulo 2 modificado”
23. Vuelve a modificar el capitulo 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"

24. Prepara tus cambios y comprueba qué ha cambiado con la última instantánea confirmada.
25. Elimina del repositorio el fichero _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.

```
# 1) Crear repo "libro", mostrar contenido y estado
cd ..
mkdir libro && cd libro
git init
ls -la
git status

# 2) Crear indice.txt y primer commit
cat > indice.txt <<'TXT'
Capitulo 1: Introducción
Capitulo 2: Los tres cerditos
Capitulo 3: Caperucita roja
TXT
git add indice.txt
git commit -m "Añadido índice del libro"
git status   # limpio: nada que confirmar

# 3) Añadir capítulo 4 y commit
cat >> indice.txt <<'TXT'
Capitulo 4: La bella y la bestia
TXT
git add indice.txt
git commit -m "Añadido 4: La bella y la bestia"
git status
```

```
# 4) Historial
git log --oneline --graph --decorate

# 5) Carpeta capitulos y capitulo2.txt; commit
mkdir -p capitulos
echo "Y el lobo sopló y sopló y la casa derribó." > capitulos/capitulo2.txt
git add capitulos/capitulo2.txt
git commit -m "Añadido capitulo 2"
git log --oneline --graph --decorate

# 6) Crear capitulo3.txt (sin preparar aún)
echo "Abuelita qué ojos más grandes tienes." > capitulos/capitulo3.txt

# Ver estado abreviado e interpretar
git status -s
# Significados rápidos:
#  A = Added (en área de preparación), M = Modified, ?? = Untracked, D =
Deleted, R = Renamed

# 7) Modificar índice añadiendo Capítulo 5 y subir cambios ignorando
capitulo3.txt
echo "Capitulo 5: Forzen" >> indice.txt
echo "capitulos/capitulo3.txt" > .gitignore
git add indice.txt .gitignore
git commit -m "Añadido capítulo 5 y temporalmente ignorado capitulo3.txt"

# 8) Cambiar ignorados: ignorar todo lo que empiece por _ excepto _ayuda.txt
#    (y dejar de ignorar capitulo3.txt)
cat > .gitignore <<'GITIGNORE'
_*          # ignora ficheros que empiecen por _
!_ayuda.txt # excepción: NO ignorar _ayuda.txt
GITIGNORE

# Crear _logs y _ayuda
echo "Fichero para almacenar logs" > _logs.txt
echo "Fichero de ayuda." > _ayuda.txt

git commit -m "Añadidos cambios varios y reglas de ignorados (incluye
capitulo3 y _ayuda)"

# 9) Modificar capitulo2.txt (sustituir contenido)
cat > capitulos/capitulo2.txt <<'TXT'
Caperucita iba por el bosque
```

```
con su capa roja
TXT

# Ver qué ha cambiado y aún no has preparado:
git status -s     # verás ' M capitulos/capitulo2.txt'
git diff          # muestra diferencias SIN preparar

# Confirmar cambios:
git add capitulos/capitulo2.txt
git commit -m "Capitulo 2 modificado"

# 10) Volver a modificar capitulo2 y preparar; comprobar contra la última
instantánea
cat > capitulos/capitulo2.txt <<'TXT'
Caperucita iba por el bosque con su capa roja
cuando llegó a casa de su abuela le dijo
"Abuela qué ojos más grandes tienes"
TXT

git add capitulos/capitulo2.txt
git diff --cached         # diferencias preparadas respecto a HEAD
git diff --name-status HEAD  # qué cambió respecto a la última instantánea

# 11) Eliminar del repositorio _ayuda.txt
git rm _ayuda.txt
git commit -m "Eliminar _ayuda.txt del repositorio"

# 12) Renombrar indice.txt a indice_libros.txt y subir cambios
git mv indice.txt indice_libros.txt
git commit -m "Renombrado indice.txt a indice_libros.txt"

# 13) Cambiar el mensaje del último commit (en local)
git commit --amend -m "Renombrado índice a indice_libros.txt"
```