

## Ejercicio Git

Instalamos los programas para convertir el .md a PDF

```
sudo apt install pandoc textlive-xetex
```

Después configuramos la extensión de VSCode Pandoc con el comando `--pdf-engine=xelatex` en la opción PDF Opt String

## Repositorio Git local

Inicializamos el repositorio con

```
git init ud02Git.git
```

Creamos los archivos miguel.txt y aficiones.txt y los añadimos

```
git add .
```

Para hacer commit escribimos el comando

```
git commit -m "Creación de los archivos .txt"
```

Creamos una carpeta con dos archivos para ser ignorada y hacemos un commit de los cambios que realizamos en el fichero aficiones. Para ello solo hacemos seguimiento de los archivos trackeados. Incluimos la carpeta al archivo creado por nosotros .gitignore

```
git commit -a
```

Escribimos

```
git log
```

y copiamos los 10 primeros dígitos del commit. A continuación:

```
git checkout d6a256cc17
```

## Ejercicio 2

Creamos el repositorio nuevo en la carpeta pagina\_web `bash git init . ls -la` Comprobamos el estado del repositorio con `bash git status` No hay nada para confirmar ni ningún commit

Creamos el fichero index.html, hacemos `git add` y `git commit`.

Con `git status` comprobamos que se ha realizado un commit, además de que el árbol de trabajo está limpio.

Modificamos el fichero con las 3 ciudades a visitar y `bash git commit -a git log`

Y enseñamos el historial de commits.

Creamos una carpeta por cada ciudad y creamos el index html dentro de cada una con la info

```
mkdir Oslo Venecia Rio
touch ./Oslo/index.html

git add .
git commit -m "Añadida información de las ciudades a visitar"
git log
```

## Ejercicio 3

Creamos un nuevo repositorio en local con mkdir y git init

```
mkdir ejercicio3
cd ejercicio3
git init
```

Creamos el fichero indice.txt con la info del libro.

Y lo añadimos a git

```
git add .
git commit -a
git status
git log
```

Vemos que el arbol de trabajo está limpio y se ha añadido un commit con “Añadido indice del libro”

Cambiamos la info del indice.txt y hacemos commit

```
git commit -a
git status
git log
```

Creamos la carpeta capítulos y dentro el fichero capítulo2.txt

```
mkdir capítulos
touch capitulo2.txt
git add .
git commit -m "Añadido capitulo 2"
```

Y lo mismo con capítulo3

```
touch capitulo3.txt
```

Con git status –short vemos el estado del repositorio de foma abreviada.

Archivo modificado (M) Archivo no rastreado (??) Archivo preparado (A).

nos dice ?? capitulo3.txt

Modificamos el indice.txt y subimos los cambios ignorando el capitulo3. creamos el archivo .gitignore y lo añadimos.

Modificamos el archivo .gitignore para que ignore todos aquellos ficheros que comiencen por \_ a excepción del fichero \_\_ayuda.txt ([\_])

Creamos \_\_logs.txt con el contenido “Fichero para almacenar logs”

```
echo "Fichero para almacenar logs" -> __logs.txt
```

Creamos el fichero \_\_ayuda.txt con el contenido “Fichero de ayuda” de la misma forma que realizamos antes

Preparar todo con git add \*. Explicad qué pasa. Se añaden los archivos para trackear

Hacer un commit de los cambios con el mensaje “Añadido capitulo 2”. Comprobar/explicar qué se sube al repositorio.

```
git commit -m "añadido capítulo 2"
```

Modificar el fichero capitulo2.txt (elimina lo que había antes). Añadimos al capitulo 2: Caperucita iba por el bosque con su capa roja Lo modificamos con el nano

```
git -m "capitulo 2 modificado"
```

Lo volvemos a cambiar por

Caperucita iba por el bosque con su capa roja cuando llegó a casa de su abuela le dijo “Abuela qué ojos más grandes tienes”

Eliminamos \_\_ayuda.txt Cambiamos el nombre del fichero indice.txt por indice\_libros.txt

```
git rm ayuda.txt
git mv indice.txt indice_libros.txt
git add *
git commit -m "Ejercicio 2 finalizado"
```