

Ejercicio Control de versiones con GIT 1.

Ejercicio 1.

- En un directorio de trabajo, por ejemplo, en la unidad C: y con el nombre GIT, crea la carpeta *ejercicioCV1*.
 - Si se hace con CMD - > md ejercicioCV1
 - Si se hace con git bash - > mkdir ejercicioCV1
- Abre la consola *git bash* y posíciónate en dicha carpeta desde la línea de comandos.
 - Si se hace con CMD - > cd c:/git
 - Si se hace con git bash - > cd c:\git
- Convierte la carpeta en un repositorio local.
 - git init.
- Confirma que se han creado la carpeta oculta .git y su contenido.
 - Si se hace con CMD - > dir /ah
 - Si se hace con git bash - > dir -ah
- Anula que la carpeta sea un repositorio local, para ello basta con borrar la carpeta oculta .git y todo su contenido.
 - Si se hace con CMD - > rd .git /s /q
 - Si se hace con git bash - > rm -rf .git

Ejercicio 2.

- Vuelve a crear la carpeta *ejercicioCV1* como en el primer punto del ejercicio 1.
- Abre Visual Studio Code.
- Dentro del programa, selecciona la carpeta *ejercicioCV1*.
- Abre una terminal en VSC y usa comandos GIT a partir de aquí.
 - Terminal / Nuevo Terminal
- Convierte la carpeta en un repositorio local.
 - git init
- Identifícate con un nombre y un correo electrónico en GIT.
 - git config --global user.name yomismo
 - git config --global user.email yomismo@gmail.com
- Dentro de la carpeta crea la página web *index.html* y añádela el título *Ejercicio Control de Versiones 1*.
- Consulta el estado de cambios del archivo recién creado.
 - git status -s
- Añade el archivo al área intermedia para su seguimiento.
 - git add index.html
 - git add .
- Consulta el estado de cambios del archivo recién creado.
 - git status -s
- Confirma los cambios y crea una instantánea con el nombre *Inicio*.
 - git commit -m "Inicio"
- Consulta el historial de commits.
 - git log

- `git log --oneline`
- Añade a la página web *index.html* un párrafo con el texto *Primer párrafo añadido*.
- Consulta el estado del archivo.
 - `git status -s`
- Añade el archivo a la zona intermedia.
 - `git add index.html`
 - `git add .`
- Crea una nueva instantánea con el nombre *Primer cambio. Párrafo 1 añadido*.
 - `git commit -m "Primer cambio. Párrafo 1 añadido"`
- Consulta el historial de commits.
 - `git log`
 - `git log --oneline`
- Añade a la página web *index.html* un párrafo con el texto *Segundo párrafo añadido*.
- Consulta el estado del archivo.
 - `git status -s`
- Añade el archivo a la zona intermedia.
 - `git add index.html`
 - `git add .`
- Crea una nueva instantánea con el nombre *Segundo cambio. Párrafo 2 añadido*.
 - `git commit -m "Segundo cambio. Párrafo 2 añadido"`
- Consulta el historial de commits.
 - `git log`
 - `git log --oneline`
- Añade a la página web *index.html* una lista no ordenada con los elementos *HTML, CSS y JavaScript*.
- Consulta el estado del archivo.
 - `git status -s`
- Añade el archivo a la zona intermedia.
 - `git add index.html`
 - `git add .`
- Crea una nueva instantánea con el nombre *Tercer cambio. Lista 1 añadida*.
 - `git commit -m "Tercer cambio. Lista 1 añadida"`
- Consulta el historial de commits.
 - `git log`
 - `git log --oneline`
- Volver al momento anterior en que no existía la lista.
 - `git reset --haré código.`
- Consulta el historial de commits.
 - `git log`
 - `git log --oneline`
- Volver al momento inicial.
 - `git reset --hard código.`
- Consulta el historial de commits.

- `git log`
 - `git log --oneline`
- Volver de nuevo al último cambio.
 - `git reset --hard codigo`.
- Consulta el historial de commits.
 - `git log`
 - `git log --oneline`