

Sistema de Control de Versiones, Tema 3

Uso de Git – Práctica 1

1. Entra al git bash o a la línea de comandos de tu equipo (de preferencia usar git bash) y establece tu nombre de usuario y correo electrónico en Git, esto es importante porque las confirmaciones de cambios (commits) en Git usan esta información. (captura en la que aparezcan todas las configuraciones realizadas), usa los siguientes comandos, obviamente poniendo tu nombre y correo electrónico.

```
$ git config --global user.name "Nombre Apellido"
Se usa para vincular los commits a tu nombre. Puedes establecer o
cambiar el nombre de usuario utilizando el comando git config.

$ git config --global user.email "mail@gmail.com"
Úsalo ara cambiar la dirección de correo electrónico asociada a
tus commits.
```

Anexa captura de pantalla de lo que aparece al ejecutar el comando: **git config –list**

2. Crea un directorio en tu equipo con el nombre “**ProyectoGIT**” y ubícate o cámbiate a dicho directorio (línea de comandos git bash) y Crea 2 archivos en la carpeta **ProyectoGIT** (puede ser un html y un CSS) que hagan algo básico como, **poner un título, un párrafo y una imagen** (puedes usar Visual Studio Code, para manejar el contenido de dichos archivos, anexa captura de los 2 archivos en tu directorio, el código fuente de cada archivo y resultado que aparece en tu navegador.
3. Una vez guardados los cambios en ambos archivos, ahora vamos a crear nuestro primer repositorio GIT, para ello usaremos el comando “**git init**” desde el directorio

de nuestro proyecto y luego ejecutamos el comando “**git status**” y aquí incluimos la captura de pantalla que aparece al ejecutar el comando anterior, que será el status o estado actual de nuestro proyecto, aparecerá algo como esto y deben tomar captura:

```
personal > EjemplosPHP > main > git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
```

4. Una vez que hemos creado el repositorio, en realidad lo que sucedió es que el comando “git init” creo un directorio oculto, con los archivos necesarios para git, para tener todo el control de las diferentes versiones de nuestro desarrollo y administrador eficientemente, para ver los archivos creados, realizar lo siguiente:

\$ ls -a (muestra la lista del contenido del directorio, incluyendo los ocultos).

\$ cd .git (permite cambiar nuestra ubicación, pasando al directorio creado por git).

\$ ls (muestra el contenido de la carpeta creada por git).

\$ cd .. (permite volver a la carpeta contenedora de nuestro proyecto, para continuar con nuestro trabajo).

```
personal > EjemplosPHP > main > ls -a
./          '12 manejoCadenas.php'  '6 operadoresAritmeticos.php'
../         '13 FuncionesPHP.php' '7 operadoresComparacion.php'
.git/       '2 html+php.php'       '8 operadoresLogicos.php'
'1 holamundo.php' '3 comentarios.php'    '9 condicionales.php'
'10 ciclos.php'   '4 variables.php'      README.md
'11 SalidaPrintf.php' '5 concatenar.php'

personal > EjemplosPHP > main > cd .git
personal > .git > main > ls
HEAD      description  index  logs/    packed-refs
config    hooks/         info/   objects/ refs/
personal > .git > main > cd ..
personal > EjemplosPHP > main >
```

**FIN DE LA PRÁCTICA 1. SUBIR SU EVIDENCIA INDIVIDUAL USANDO
EL ENLACE QUE LES INDIQUE EL DOCENTE.**