

OBJETIVO

El objetivo es poner en práctica los contenidos sobre control de versiones.

- 1) Uso de comandos Git en local. 2,5 puntos.
- 2) Trabajando con repositorios remotos y ramas. 5 puntos.
- 3) Git en entorno gráfico. 2,5 puntos

ESCENARIO

- 1) Disponer de cuenta en GitHub y haber asociado una clave ssh para poder conectarse desde local sin necesidad de user y pass.
- 2) El punto 3 de la práctica se realizará obligatoriamente por parejas. El resto de la práctica se realizará obligatoriamente de manera individual.

PUNTOS A REALIZAR

1. Preparación del escenario. 0 puntos.

Este apartado solo es necesario para aquellas personas que no hayan podido crear y configurar su cuenta en Github.com durante el transcurso del curso. En caso de ya disponer de una cuenta configurada pasar al apartado 2.

1.a) Creación de una cuenta en Github

Entre en <https://github.com/> y registra una cuenta en github.
Acepta el email de confirmación.
Logueate con tu cuenta en Github.com

1.b) Asociar las claves ssh

Chequea si tienes claves ssh ya generadas. <https://help.github.com/articles/checking-for-existing-ssh-keys/>
Genera una nueva clave ssh y añadelas al ssh-agent. <https://help.github.com/articles/generating-a-new-ssh-key-and-adding-it-to-the-ssh-agent/>
Añade tu nueva clave ssh a tu perfil de github.com <https://help.github.com/articles/adding-a-new-ssh-key-to-your-github-account/>

1.c) Configuración de la terminal con zsh y oh my Zsh (no es obligatorio pero sí recomendable)

“Oh my Zsh!” es una herramienta que nos permite aplicar temas a nuestro *prompt* y agregar diversos plugins a nuestra shell Zsh. Nosotros lo vamos a usar para tener información integrada de GIT en nuestro terminal, como por ejemplo información del repositorio Git en el que nos encontramos. Oh my zsh requiere instalar el shell zsh o z shell.

Fuente: <https://github.com/robbyrussell/oh-my-zsh/blob/master/README.md>

Instala zsh + oh-my-zsh para ello:

Instala zsh y configuralo como tu shell por defecto

(Instalo)

```
sudo apt install zsh
```

(Cambio de shell changeshell →)

```
chsh -s $(which zsh)
```

Instalación oh my zsh

```
sh -c "$(curl -fsSL https://raw.githubusercontent.com/robbyrussell/oh-my-zsh/master/tools/install.sh)"
```

2. Trabajando con comandos de git en local. (0,5 cada apartado)

- 2.a) Crea un directorio **nuevo** de trabajo llamado “proyectoGit” que será la zona de trabajo del proyecto que vas a controlar con Git. Entra en el directorio “proyectoGit” e inicializa un repositorio. Investiga sobre el directorio .git creado al inicializar el proyecto. ¿Qué carpetas contiene y para qué sirven?
- 2.b) Crea 2 archivos vacíos index_a.html y index_b.html en el directorio de trabajo. Comprueba el estado de tu proyecto git. Añade los 2 ficheros al área de preparación y confirma los cambios con el mensaje. “Inicializados ficheros index_a e index_b” ¿Cuántos ficheros cambiados, insertions (+) y deletions (-) aparecen? Añade contenido al index_a.html y al fichero index_b.html.
- 2.c) Pasa **solo** el fichero el index_a.html a la zona de preparación (stage area). Vuelve a comprobar el estado de tu repo. Confirma **solo** los cambios del fichero index_a.html con el mensaje “Añadidos detalles a index_a.html”. Vuelve a comprobar el estado de tu repo.
- 2.d) Comprueba las diferencias entre el último commit y tu directorio de trabajo. Añade el index_b.html a la zona de preparación. Comprueba los cambios entre la zona de preparación y el último commit. Confirma los cambios con el mensaje “Añadidos detalles a index_b.html”.
- 2.e) Comprueba el historial de commits del repositorio mostrando una línea por cada commit. Comprueba el historial de commits del repositorio mostrando los cambios en cada commit
- 2.f) Taguea la versión v1.0.0 (busca información sobre cómo hacerlo si no está en apuntes)
- 2.g) Ignora todos los ficheros de log. (extensión .log), pruébalo.
- 2.h) Comprueba el estado de tu repo del apartado 2 y confirma que no tienes cambios pendientes. Añade cambios al fichero index_a.html y vuelve a comprobar el estado de tu repo. Descarta los cambios del fichero index_a.html del directorio de trabajo y vuelve a comprobar el estado de tu repo.
- 2.i) Añade cambios al fichero index_b.html. Añade el fichero index_b.html al área de preparación y comprueba el estado de tu repo. Saca el fichero index_b.html del área de preparación y comprueba el estado de tu repo. Añade más modificaciones al fichero index_b.html y **con un solo comando** confirma las modificaciones pendientes del fichero index_b.html con el mensaje “Añadidos cambios al fichero b”
- 2.j) Añade modificaciones al fichero index_a.html y añade al área de preparación el fichero index_a.html. Modifica el último commit (Añadidos cambios al fichero b) hasta el momento para que incluya también los cambios en index_a.html y que el mensaje sea “Añadidos cambios a los ficheros a y b”. Comprueba en el histórico que ha sobrescrito el commit.

Respuesta: Pantallazos que demuestren la realización del ejercicio con todos los comandos que se piden

3. Trabajando con repositorios remotos y ramas (1pto cada apartado)

Asegúrate que la profesora te ha añadido como colaborador a la organización dawenlaces en github. Sino envíale por correo tu usuario de github.

- 3.a) Clona por ssh el repo “practica2” de la organización dawenlaces2018 de GitHub (debes ser miembro porque te habrá llegado invitación de la profesora al correo). Lista los repositorios remotos que tienes en local, qué nombre le pone al remoto por defecto?

Crea en local una rama “nombre” donde nombre debe ser vuestros nombres. Ejemplo: teresa_juan.

- 3.b) Crea en esa rama un directorio con tu nombre y un fichero de texto con información: Ejemplo: teresa/info.txt. Añade al fichero de texto la siguiente información (en líneas separadas):

- tu nombre y apellidos
- tu dirección de correo electrónico
- tu equipo de futbol favorito (atención: el equipo elegido puede afectar a la nota final ☺)
- Añade la imagen de tu avatar con el nombre avatar.extension. Donde extensión es la extensión de la imagen (jpg, png, ...) Ejemplo: avatar.jpg

Sube la nueva rama con los cambios al repo remoto “practica2”.

Actualiza la rama master local por si ha habido cambios en remoto. Si hay algún cambio ¿A qué es debido? Comprueba el estado de tu repo. Comprueba los últimos commits. Mergea la rama “nombre” en master. Pushea la rama master al remoto.

- 3.c) Añade al fichero README del raíz de master una nueva línea con tu nombre y apellidos, acto seguido tu compañero deberá clonar también el proyecto en otra máquina y deberá también añadir al fichero README la línea con sus datos. Subid ambos los cambios al remoto para generar un conflicto y arregladlo. Accede al historial de commits en Github de ambas ramas. Accede al último commit de master para ver tus cambios.

- 3.d) Ahora vais a replicar el proyecto a

Desde la cuenta de github de 1 compañero cread un nuevo repo llamado "daw" sin el archivo README. Añade por https el repo remoto con el nombre "daw". Lista los repositorios remotos (debes tener ambos). Sube el proyecto a ese nuevo repo.

El otro compañero desde su cuenta hace un fork del proyecto de su compañero para replicarlo en su cuenta, después modificará algún fichero y lanzará la pull request a su compañero para mergear los cambios en el proyecto del compañero. El compañero comprobará las modificaciones y aceptará la pull request, chequead que se han realizado los cambios.

Respuesta: Pantallazos reales que demuestren la realización del ejercicio con todos los comandos ordenados según pide el enunciado.

4. Netbeans y Git (1,25ptos)

- 4.a) Crea un proyecto con Netbeans (o bien utiliza un proyecto que hayas desarrollado

para dwes), inicializa el proyecto Git.

- 4.b) Añade algún cambio a tu proyecto y confirma los cambios
- 4.c) Vuelve a modificar algún fichero y compáralo con la última versión commiteada.
- 4.d) Sube tu proyecto a un repositorio en Github por ssh.
- 4.e) Modifica el proyecto en Github y actualízalo en local.

Respuesta: Pantallazos que demuestren la realización del ejercicio. Debes usar el entorno gráfico del ide y no hacerlo por comandos.

5. Visual Studio Code (1,25pto)

- 5.a) Crea un proyecto con Visual Studio Code (o bien utiliza un proyecto que hayas desarrollado para dwec/diw), inicializa el proyecto Git
- 5.b) Añade algún cambio a tu proyecto y confirma los cambios
- 5.c) Vuelve a modificar algún fichero y compáralo con la última versión commiteada.
- 5.d) Sube tu proyecto a un repositorio en Github por ssh.
- 5.e) Modifica el proyecto en Github y actualízalo en local.

Respuesta: Pantallazos que demuestren la realización del ejercicio. Debes usar el entorno gráfico del ide y no hacerlo por comandos.