**Ejercicios básicos de GIT**

1. Crea una carpeta en tu equipo.
2. Inicia en la carpeta anterior un repositorio de git.

Git init entornosprueba

Git commit -m entornosprueba

1. Añade 5 ficheros de texto: ciudades europeas, ciudades americanas, provincias españolas, paises de la UE, paises fuera de la UE.

Añado en la misma carpeta el contenido

1. Añade contenido en todos los ficheros anteriores y haz commit.

Git commit -m “añado contenido”

1. Crea otra rama (llámala develop por ejemplo) y cámbiate a ella.

Git Branch develop

1. Añade contenido en los ficheros del repositorio y haz commit.

Git commit -m “he añadido contenido

1. Vuelve a la rama anterior (master) y añade más contenido a los ficheros.

Git checkout master

1. Fusiona (merge) las ramas que tienes. Si aparecen conflictos, resuélvelos. ¿Qué has visto en los ficheros del repositorio?

Git merge

Reviso los archivos ya que me aparecen los archivos con modificación y error

1. Crea un tag o etiqueta de la versión fusionada.

git tag -a v1.4 -m 'my version 1.4'

1. Sube a tu cuenta de github el repositorio. Deberás primero crear un repositorio **vacío** (sin ficheros) en tu cuenta de github y seguir las instrucciones de la página.
2. Añade un fichero .gitignore a tu repositorio que tenga en cuenta que tu repositorio se va a manejar desde windows y en él se deberían ignorar los ficheros propios de windows, así como .pdf y .doc
3. Copia en la carpeta de tu repositorio un documento de word y un pdf.
4. Haz commit y push del contenido del repositorio. ¿Qué ha ocurrido?
5. Investiga los comandos de git: *log*, *diff* y *status*. ¿Qué hacen?

**Comando git log**

El comando que podemos usar para ver el histórico de commits, estando situados en la carpeta de nuestro proyecto, es:

git log

De un commit podemos ver diversas informaciones básicas como:

* Identificador del commit
* Autor
* Fecha de realización
* Mensaje enviado

Podría ser algo como esto:

commit cd4bcc8bad230f895fcadd5baf258715466a8eaf

Author: Miguel Angel Alvarez <malvarez@desarrolloweb.com>

Date: Fri Feb 10 18:38:41 2017 -0200

**Log en una línea por commit**

Es muy útil lanzar el log en una sola línea, lo que permite que veamos más cantidad de commits en la pantalla y facilita mucho seguir la secuencia, en vez de tener que ver un montón de páginas de commits.

Para ello usamos el mismo comando, pero con la opción "--oneline":

git log --oneline

**Ver información extendida del commit**

Si queremos que el log también nos muestre los cambios en el código de cada commit podemos usar la opción -p. Esta opción genera una salida mucho más larga, por lo que seguramente nos tocará movernos en la salida con los cursores y usaremos CTRL + Z para salir.

git log -2 -p

Eso nos mostrará los cambios en el código de los últimos dos commits. Sin embargo, si quieres ver los cambios de cualquier otro commit que no sea tan reciente es mucho más cómodo usar otro comando de git que te explicamos a continuación "git show".

**Comando Git Diff**

Ver los cambios realizados en un archivo

Cuando trabajamos sobre un archivo, para realizar alguna modificación o corrección de algo, a menudo, necesitamos ver que hemos cambiado sobre el archivo original.

Git nos permite ver los cambios realizados sobre un archivo. Y se hace con lo siguiente:

git diff nombre archivo

Con esto, se listaran en la consola aquellas secciones que se hayan modificado lo viejo y lo nuevo, de modo que podamos ver exactamente lo que hemos modificado.

Si no queremos ver todo el contenido de los cambios del comando diff, podemos emplear el comando git diff –stat que muestra un sumario de los cambios

git diff –stat

**Git status, ver los archivos que se han modificado**

git status

Al escribir este comando aparece una lista con los archivos creados, modificados o eliminados que aún no se encuentran en el área de ensayo (en color verde).

Para ver los cambios de forma más esquematizada:

Git status -s

Con el comando esquematizado, dependiendo de las modificaciones que se hayan hecho y si se han añadido o no al área de ensayo, aparecerá una letra en color verde o rojo:

A Archivo creado y añadido M Archivo modificado y añadido M Archivo modificado y no añadido D Archivo eliminado UU Conflicto entre ramas

1. Pon en un fichero de texto los comandos de git que has introducido en cada ejercicio junto con las respuestas a las cuestiones que se plantean y añádelo a tu repositorio. Haz commit, push y envíame el enlace por email.