

Introducción a Bash y Git

Carlos Malanche

1 de febrero de 2018

1. Pequeña introducción

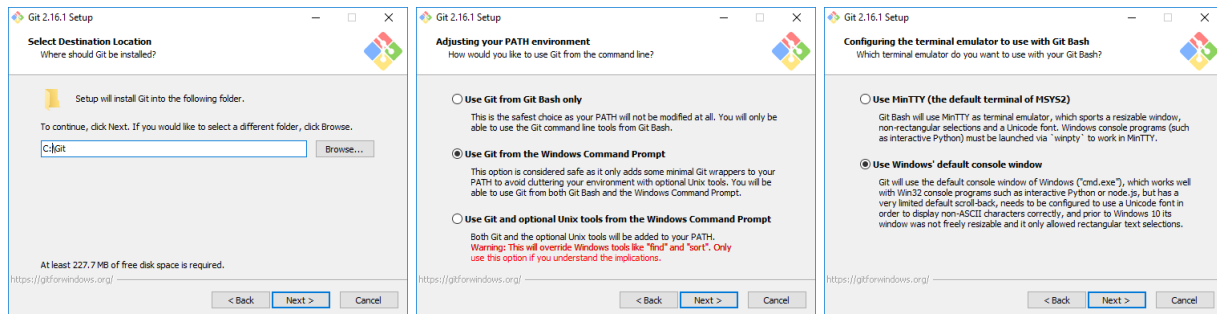
Para los que no sepan lo que es bash, es el intérprete de comandos por defecto en la gran mayoría de las distribuciones de Linux, a veces sólo le llaman *la terminal*. Su utilidad es la de navegar por el sistema e interactuar con los archivos (editarlos, copiarlos, pegarlos, hacer búsquedas, etc.) que en él viven.

Bash siempre ha sido un privilegio de los sistemas Unix (MacOS y Linux), pero desde hace unos meses hay soporte nativo para bash en Windows 10. Aquí, por cuestiones de simplicidad, no usaremos este soporte básico de bash (pues consiste en una máquina virtual, y eso lo hace un tanto menos... compatible con Windows, por así decirlo). En mi opinión, lo mejor es descargar el cliente de Git para windows, que incluye todos los comandos que utilizaremos.

2. Git

Sacado directamente de la [página de git](#): Es un sistema de control de versiones gratuito y libre (de código abierto vaya). Digamos que es una especie de Dropbox que funciona desde la terminal, y que está optimizado para trabajar con códigos fuente. Git, como ya lo había mencionado antes, viene con sus *implementaciones* de varios comandos de bash (además de una emulación de bash), comandos que veremos en esta clase.

Para los alumnos que tengan Windows, basta con descargar el cliente de Git de la página de internet e instalarlo, aceptando casi todos los valores por defecto en la instalación; Windows tiene esta *espantosa mañana* de usar espacios en los nombres de los archivos, por lo que git propondrá la ruta “C:\Program Files\Git”. Hay que cambiarla por “C:\Git”.



Con esto ya se puede utilizar git desde cmd (la terminal de windows) o Powershell. Sin embargo, yo recomiendo utilizar [cmder](#) como emulador de una terminal de Linux. (Si alguien tiene duda de cómo se usa, me pueden mandar un correo).

Para los usuarios de MacOS es suficiente abrir una terminal (`command+space` para abrir la búsqueda, y escriben *terminal*) y ejecutar el comando `git --version`. Si Git no está instalado, les va a preguntar si quieren instalarlo.

Para los que quieran utilizar Linux, basta con ejecutar `sudo apt install git` desde la terminal.

3. Comandos de Bash

Vamos a aprender a navegar por un sistema de archivos por medio de una terminal. Esto va a ser de gran utilidad pues a la mejor un día necesitaremos correr simulaciones en servidores que no cuentan con una interfaz gráfica.

3.1. Algunos ejercicios

Para agarrar confianza con la máquina, ejecute los siguientes comandos en la terminal (las palabras en verdes son argumentos forzosos):

- `man`: Sólo disponible en Unix. Despliega información de un comando y sus usos (por ejemplo, `man ls`). Junto con *Google*, su herramienta más útil **antes** de venir a preguntarme algo que no entienden.
- `pwd`: Corto para *path of working directory*. Dice el directorio en el que nos encontramos trabajando.
- `ls`: Lista los documentos y carpetas del directorio en el que estamos. Yo siempre lo ejecuto con la opción `-la`, pues salen todos los archivos (también los ocultos) en forma de lista con permisos de ejecución.
- `cd dirname`: Corto para *change directory*. Nos permite cambiar al directorio *dirname*. Como nota importante, todo directorio contiene dos directorios con nombres “.” y “..”. El primero es la ruta al directorio actual, y el segundo la ruta al directorio superior.
- `mkdir dirname`: Crea un directorio bajo el nombre *dirname*.
- `rmdir dirname`: Borra un directorio de nombre *dirname*.
- `rm filename`: Borra un archivo
- `mv filename otherfilename`: Mueve (o renombra) el archivo *filename* al archivo *otherfilename*.
- `cp filename otherfilename`: Copia el archivo *filename* al archivo *otherfilename*.

Eso debiera ser suficiente por el momento. Hay muchas otras instrucciones *divertidas* en bash, pero por simplicidad, usaremos sólo lo necesario.

4. Comandos de Git

Como ya debieron haber notado, el material del curso se encuentra en la página <https://github.com/Malanche/fc-adml>. Github es un sitio donde se nos permite tener una *copi*a de un repositorio en la nube (por así decirlo). Lo equivalente al cliente web de *Google Drive* o *Dropbox*.

Git es un comando más de la terminal (si ya lo han instalado...). Si ustedes quieren tener una copia local del repositorio `fc-adml`, basta con abrir una terminal en el directorio que gusten (por ejemplo, *Mis Documentos*) y ejecuten:

```
git clone https://github.com/Malanche/fc-adml
```

El repositorio es público, por lo que no será necesario que hagan una cuenta en Github (por el momento) para ver los contenidos del repositorio en su computadora. Se va a crear un directorio bajo el nombre `fc-adml` que contiene los archivos del repositorio. Una vez que se clonó, no vuelve a haber comunicación con el servidor, por lo que (a diferencia de *Google Drive* y *Dropbox*) los archivos no estarán sincronizados con lo que hay en el servidor. Para eso hay un nuevo y divertido comando:

```
git pull
```

Si hay cambios nuevos en el servidor, llegarán a ustedes con este comando.

Ustedes deberán utilizar Git para subir sus tareas al servidor, por lo que deberían abrir una cuenta en github (es gratuito).

No es necesario ser un experto con Git, pero sí que hay que entender lo que se puede hacer con él (cuando hagan tareas en equipo verán la importancia de este software). Les dejo, por último, [este tutorial interactivo de git](#) (donde les van a explicar como crear un repositorio, añadir archivos, borrarlos, hacer *commits*, etc.).