**Resumen contenidos Periodismo de datos**

En este archivo se encuentra un resumen de los conocimientos adquiridos en la asignatura de periodismo de datos expuesto según el orden seguido en las clases.

1. **Instalar un programa de emulación de la terminal**

Yo personalmente he trabajado desde el sistema operativo Windows, por lo que he instalado dos programas de emulación de la terminal: WSL (<https://docs.microsoft.com/es-es/windows/wsl/>) y [Cygwin](https://docs.microsoft.com/es-es/windows/wsl/).

**WSL: Instalación de Ubuntu**

En la primera clase práctica instalamos WSL a través de una terminal desarrollada por Microsoft: PowerShell. Para instalarlo abrimos una sesión como administradores y ejecutamos el comando wsl –install -d Ubuntu, reiniciamos el ordenador y ya podíamos acceder al programa, solo faltaba crear nuestro usuario y elegir una contraseña para el usuario administrador (usuario UNIX)

**Cygwin: Unix en Windows**

Cierto es que Cygwin fue instalado semanas más tarde en el trascurso de la asignatura, sin embargo, lo incluyo ahora porque fue un proceso similar al recién expuesto.

Cygwin en su caso no es una herramienta nativa de Microsoft, por lo que tuvimos que descargar su instalador de su propia web. Los pasos a seguir fueron los siguientes:

* Cuando nos preguntó desde dónde queríamos instalarlo seleccionamos la opción “Instalar desde Internet”.
* ¿Dónde instalarlo? En la ruta que se establece por defecto
* Elegimos el *mirror*, dominio.es al estar en España.
* Finalmente, instalamos los paquetes libcurl4, wget, ca-certificates-letsencrypt, lynx, nano y openssl.

**2. Configuración del programa**

Para facilitar el uso de los programas debimos configurar algunas cuestiones.

**En primer lugar, creamos un alias en la terminal,**

de este modo evitamos escribir toda la ruta de nuestra carpeta.

Es decir, para que en vez de tener que teclear cd ‘/mnt/c/Users/usuaria/Documentos/Apuntes/Periodismo de Datos/’, pudiésemos acceder a dicha ruta tecleando simplemente  micasa,.

¿Cómo lo hacemos?

* Editando el archivo de configuración de Bash: en la línea de comandos escribimos el comando echo  y enviamos la salida con >>al final del archivo. .bashrc: echo “alias micasa=“cd ‘/mnt/c/Users/usuario/Documentos/Apuntes/Periodismo de Datos/’” >> $HOME/.bashrc.
* De este modo, al acceder a Ubuntu o a Cygwin, con tan solo teclear nuestro alias en la terminal podremos acceder a su ruta asignada.
* Además, utilizaremos cat $HOME/.bashrc para ver el archivo de configuración del alias y nano $HOME/.bashrc para editarlo.

**En segundo lugar, cambiamos la home (nuestra “casa”, nuestro usuario) de Cygwin a la de Windows.**

Para ello deberemos editar el archivo “hogar” o “casa” de nuestro propio usuario.

* Editamos el archivo nsswith.conf con nano: nano /etc/nsswith.conf poniendo db\_home: windows al final del archivo.
* Guardamos (CTRL + O) y cerramos (CTRL + X) para conservar los cambios.
* Para comprobar que funciona, cuando volvamos a abrir el programa, escribimos pwd , y nos debería contestar.
* Para que se actualice, cerramos Cygwin y volvemos a entrar.
* Comprobamos con pwd: /cygdrive/c/Users/usuario.

**Finalmente, configuramos nuestro usuario de GitHub en la terminal.**

De este modo, podemos vincular los cambios que se hacen en nuestros archivos locales con nuestro repositorio en línea.

* Escribimos git config --global user.name nuestrousuario (en mi caso saracse) en nuestra terminal.
* Escribimos git config --global user.email correogithub (siendo el correogithub el correo que hayamos utilizado para crear nuestra cuenta GitHub

**3. Configuración de un programa de edición de texto**

Para poder editar los textos de nuestros archivos desde la terminal, utilizamos el editor de texto de Linux, nano.

* Antes que nada, configuramos nano para que el texto se adapté a la resolución de la pantalla y para que aparezca el número de líneas. Ejecutamos nano $HOME/.nanorc y en el editor, ponemos: # Ajustar el texto a pantalla

set softwrap

# Numerar las líneas

set linenumbers

En el caso de Cygwin,

* Copiamos el archivo de configuración de nano con cp /etc/nanorc .nanorc, y lo editamos: nano .nanorc.
* Buscamos con CTRL + W “linenumbers” y en la línea (# set linenumbers) le quitamos la #.
* Volvemos a buscar con CTRL + W “softwrap” y en la línea (# set softwrap) le quitamos la #.

**4. Configuración y funcionamiento de un gestor de paquetes/programas del emulador de la terminal**

El gestor de paquetes de Ubuntu es apty, este viene ya incorporado al programa.

Sin embargo, en el caso de Cygwin es apt-cyg y hay que instalarlo.

* Ejecutamos lynx -source rawgit.com/transcode-open/apt-cyg/master/apt-cyg > apt-cyg y luego install apt-cyg /bin.

El gestor de paquetes nos permitirá instalar herramientas en un futuro, como lolcat.

**5. Versión del lenguaje de SHELL utilizado**

En el caso de la terminal de Ubuntu, Bash es el intérprete de lenguaje de comandos, es decir uno de los lenguajes posibles de Shell.

Para conocer que Shell utilizamos podemos usar el comando echo $0 que nos devolverá a partir de esta variable de entorno la shell que estamos utilizando en el caso de WSL nos devuelve -BASH.

**6. Valor de la variable de entorno PATH**

Las variables de entorno en mi caso la variable de entorno es $PATH propia de los sistemas de Microsoft.

Para ver el valor de la variable PATH podemos imprimirla directamente a través del comando echo $PATH.

Las variables de entorno tienen al principio el signo $

* si ponemos simplemente $PATH la terminal nos muestra las rutas con los programas que son posibles ejecutar.
* si quisiéramos saber el valor de las rutas de la variable escribiríamos echo. echo $PATH.

**7. Comandos utilizados y ejemplos**

*Los comandos tienen la siguiente estructura: comando / opciones / argumentos*

* ls: listar un directorio (ver qué archivos y carpetas hay en la ruta en la que estamos). Si queremos ver el directorio actual, no hace falta especificar la ruta.
* cd: acceder a un directorio en específico. Si escribimos solo cd vamos al directorio raíz. cd .., nos sube un directorio. Con cd -nos lleva al último sitio en el que estábamos.
* pwd: indica en qué ruta estamos actualmente.
* mkdir: crea una carpeta nueva especificando su nombre.
* cat: previsualizar el archivo en la propia terminal. Por ejemplo, cat README.md.
* mv: mueve o corta las carpetas o archivos de sitio, pero también sirve para renombrarlos. Si queremos mover la carpeta de periodismo de datos (/mnt/c/Users/usuario/Documentos/periodismo-de-datos) a otro lado, escribimos: mv /mnt/c/Users/usuario/Documentos/periodismo-de-datos /mnt/c/Users/usuario/Documentos/Apuntes/ . Como vemos, después de mv escribimos la ruta de origen y lo siguiente es el directorio de destino. Si queremos renombrar, escribimos mv periodismo-de-datos nuevo-nombre.
* cp: copia carpetas o archivos. La sintaxis es la misma que para mv.
* nano → editar textos. Para editar con nano, tenemos que escribir nano seguido del nombre del archivo. Si estamos en la carpeta de la asignatura, por ejemplo, escribimos nano README.md. Para guardar CTRL + O, y para salir CTRL + X.
* echo: sirve para que la terminal nos devuelva un texto. Si queremos que la terminal nos diga Hola, escribimos echo Hola.
* &&: ejecutar dos comandos en una misma línea o de una vez. Por ejemplo, si queremos crear una carpeta y luego situarnos dentro, escribimos: mkdir carpeta && cd carpeta.
* |: pasar de una salida a una entrada. Por ejemplo, si queremos hacer un cat para leer un archivo pero visualizarlo con less, escribimos cat README.md | less.
* man: nos dice qué hace un comando y nos muestra todas las opciones posibles. Por ejemplo, si queremos saber qué hace ls y qué opciones podemos usar hacemos man ls. Por defecto, utiliza el visor | more, que para avanzar por el documento tenemos que ir clickando en la barra espaciadora, y para salir pulsar la tecla q. También está disponible | less (man ls | less)
* env: visualizar en pantalla todas las variables de entorno. Si lo usamos con | less, es un visor más ameno porque son demasiadas variables y no caben en la pantalla: env | less.
  + touch: crea un archivo nuevo. Por ejemplo, si queremos crear un README.md en nuestra carpeta, escribimos touch README.mdy ya está creado.
  + tree: como ls pero con más opciones. Con tree -L 1 le decimos cuántos niveles queremos ver. También podemos poner tree -L 2 o tree -L 3, según el nivel de detalle (mayor el número mayor el grado de detalle).
  + rm: eliminar archivos o directorios. Hay que tener cuidado porque los elimina de forma definitiva, los archivos no van a la papelera.
  + wget: descargar un archivo/contenido de una web. Tenemos que copiar el enlace de lo que nos queramos descargar.
  + curl: igual que wget pero con más opciones.