

# TEMA 2: Herramientas básicas

## 1. COMANDOS SISTEMA OPERATIVO

Es necesario aprender de comandos de operaciones realizadas por el sistema operativo ya que en ocasiones vamos a estar actuando sobre ordenadores remotos, además también lo necesitamos ya que en muchas ocasiones estaremos escribiendo programas que necesiten de estas operaciones básicas de los sistemas operativos.

**Códigos que aprender en Windows** (Se hacen siempre desde Windows Powershell):

- **Directorio:** es el concepto de carpeta. Para saber el directorio actual usamos el comando `pwd`.
- **Crear un directorio:** `mkdir nombre de la carpeta a crear`
- **Crear un subdirectorio:** `mkdir nombre_del_directorio \Nombre_del_nuevo_subdirectorio`

En macOS será igual pero cambiando la barra de dirección /

- **Moverte a otra carpeta:** `cd directorio\subdirectorio`
- **Visualizar archivos de directorio:** `dir` o `ls` (MAC)
- **Directorio anterior:** `..` Ejemplo: `cd..o cd "C:\Users\jolup\Desktop\Work\Cursos The Bridge\Curso de Data Science\Carpeta de trabajo VS code"`

**NOTA:** Cuando la ruta contenga espacios hay que poner comillas "" para que no nos de error.

- **Ruta:** la dirección que marca la ubicación de cierta carpeta. Podemos usar `cd` con la dirección absoluta de una ruta. Ejemplo: `C:\Users\jolup\Desktop\Work\Cursos The Bridge\Curso de Data Science`

Si no ponemos la ruta completa se utilizará siempre la ruta relativa a ese sitio

- **Renombrar un directorio:** `ren nombre_antiguo nombre_nuevo`
- **Renombrar un directorio:** `ren nombre_antiguo nombre_nuevo`
- **Borrar un directorio:** `rmdir nombre`
- **Crear un archivo:** `echo "Texto a introducir" > nombre.txt`
- **Revisar contenido del archivo:** `cat nombre.txt`
- **Renombrar archivo:** `ren nombre_antiguo nombre_nuevo`
- **Eliminar archivos:** `rm nombre.txt`
- **Mover archivos:** `mv nombre.txt ../` o `mv nombre.txt C:\Users\jolup\Desktop\Work\Cursos The Bridge\Curso de Data Science`
- **Limpiar la pantalla de la terminal:** `clear`

## 2. GIT Y GITHUB: Control de versiones y repositorios

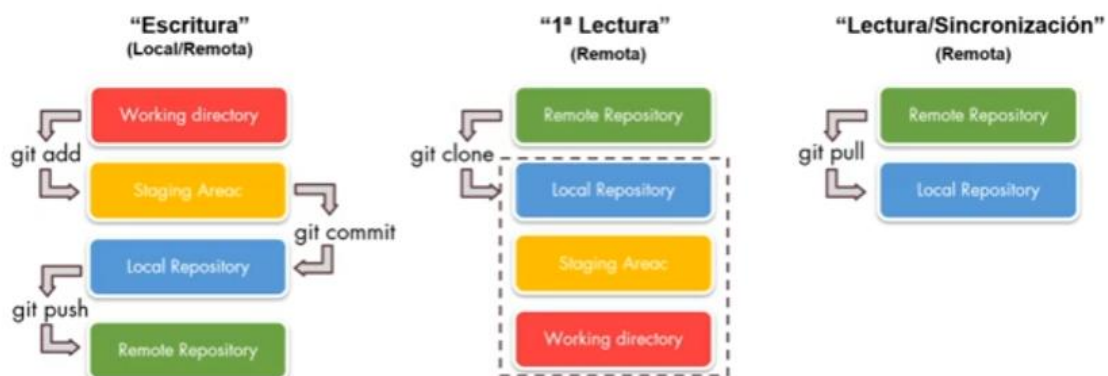
**Git es un sistema de control de versiones de código que permite guardar y mantener un proyecto de código desarrollado por varias personas de forma eficiente.** Nos permite resolver los problemas de repetición innecesarias altamente redundantes y gestión de trabajo concurrente en un mismo proyecto. Git conserva una copia de cada proyecto y mantiene diferentes vistas o ramas de cada desarrollador.

**Un repositorio es un lugar donde se almacenan los archivos de un proyecto o trabajo bajo control de versiones.** Puede ser local o remoto. Guarda el historial de cambios, permite trabajar con distintas versiones y colaborar con otros.

**Siempre se crea un repositorio local y uno remoto que se divide de la siguiente forma:**

- **Working directory:** archivos que están incompletos o se está trabajando en ello.
- **Staging área:** se guardan los archivos casi definitivos
- **Local repository:** los archivos que se encuentran validados
- **Remote repository:** pasan los archivos una vez que los consideremos terminados para que puedan acceder el resto de los desarrolladores.

Para crear un repositorio utilizamos el comando `init`. En un repositorio se puede crear archivos mediante distintos comandos `add`, `commit` y `push`. Si queremos trabajar sobre un código escrito por otra persona usamos `clone`, y lo añadimos a todos las subdivisiones, y si solo queremos sincronizar nuestro repositorio usamos `pull`.



**GitHub es un servicio basado en git que permite a los usuarios alojar sus repositorios y compartirlos con la comunidad,** habiéndose convertido en la red social de desarrolladores más famosa. Hay otros similares como GitLab y Bitbucket.

**Comandos de git en terminal Windows PowerShell:**

- **Configurar usuario:** `git config --global user.name Lunateeka`
- **Configurar cuenta de correo:** `git config --global user.email lunapereztroncoso2000@gmail.com`
- **Clonar el repositorio del curso:** primero copiamos el código del repositorio en GitHub (tenemos que haberlo creado previamente), usaremos el comando `git clone` [https://github.com/Lunateeka/ONLINE\\_DS\\_THEBRIDGE\\_Repositorio-personal-Luna.git](https://github.com/Lunateeka/ONLINE_DS_THEBRIDGE_Repositorio-personal-Luna.git)

- **Actualizar el repositorio local desde el repositorio remoto:** git pull  
[https://github.com/Lunateeka/ONLINE\\_DS\\_THEBRIDGE\\_Repositorio-personal-Luna.git](https://github.com/Lunateeka/ONLINE_DS_THEBRIDGE_Repositorio-personal-Luna.git)
- Para indicar a git que el archivo es válido para pasarlo a un repositorio (**pasarlo al área de staging**) usamos: git add primer\_push.txt
- Para comprometerse y pasarlo a un repositorio local ponemos: git commit -m "mi primer push"
- **Para mandarlo al repositorio remoto** justo después ponemos: git push

Pero todos estos comandos no son necesarios ya que normalmente vamos a trabajar con **GitHub desktop**

### 3. INTRO JUPYTER NOTEBOOKS

Para instalar **Juniper Notebooks** se usa el siguiente comando en la terminal de Windows.

*pip install notebook*

Luego nos iremos a la carpeta (cd) donde queramos tener el notebook y ponemos el siguiente comando:

*jupyter notebook*

Jupyter Notebook nos permite la creación de documentos que incluyen textos, gráficos enriquecidos, líneas de código etc...

Si observamos en la pagina web que se nos abre podemos crear un notebook con el kernel de python. Tenemos tres tipos de celdas, las celdas de código, que podemos escribir código de ese kernel, las celdas de texto que utilizan markdown para formatear el texto y luego las celdas de salida, que muestran los resultados del código. Podemos cambiar el tipo de celda de código a markdown marcándola y pulsando la tecla m, mientras que podemos volverla a pasar a código por la letra y. Se puede usar shift + enter para ejecutar.

Con la letra b se puede añadir una celda por debajo de donde hallamos marcado, y "a" se usa para crear una celda arriba.

Con las tijeras se eliminan las celdas, o también con botón derecho, delete.

### 4. INTRO MARKDOWN:

Mirar los ejercicios realizados en markdown (píldoras 7 y 8)

<https://www.markdownguide.org/basic-syntax/>

<https://daringfireball.net/projects/markdown/syntax>

<https://medium.com/analytics-vidhya/the-ultimate-markdown-guide-for-jupyter-notebook-d5e5abf728fd>