

Redirecciones: cómo funciona la shell 7/21

RECURSOS

MARCADORES

Para entender qué son las redirecciones vamos a aprender cómo manejar entradas y salidas a través de operadores especiales.

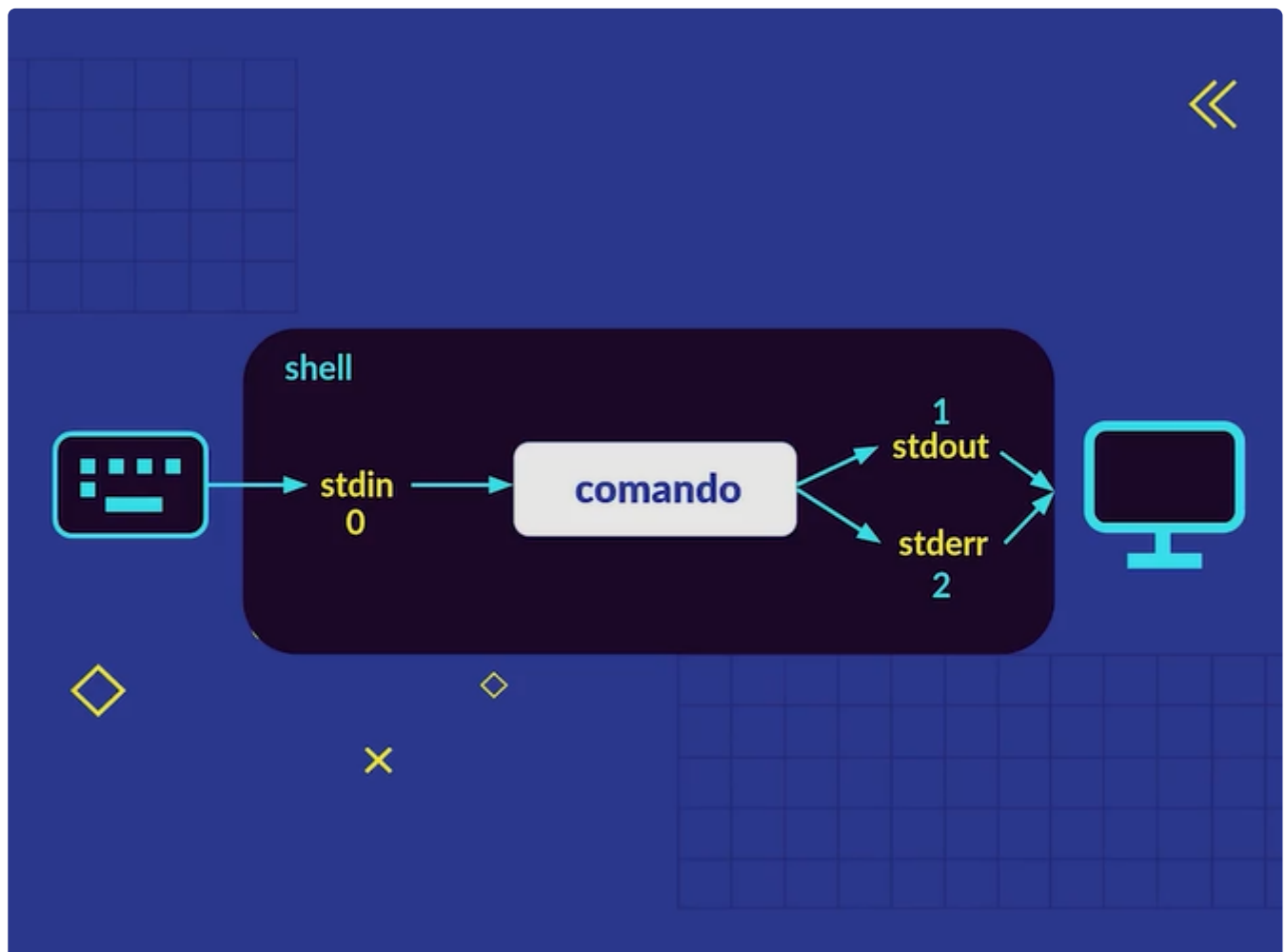
Qué son las entradas y salidas de la terminal

En la consola nosotros generamos una entrada cuando escribimos y una salida casi siempre que ejecutamos un comando.

A las entradas típicamente se les suele llamar **Standard Input** y a las salidas **Standard Output**, además se les suele abreviar como **stdin** y **stdout** respectivamente.

Qué son file descriptors

Los file descriptors son números que identifican un recurso. Funciona asociando un número con una acción, archivo o programa, en el caso de la shell tenemos 3 file descriptors:

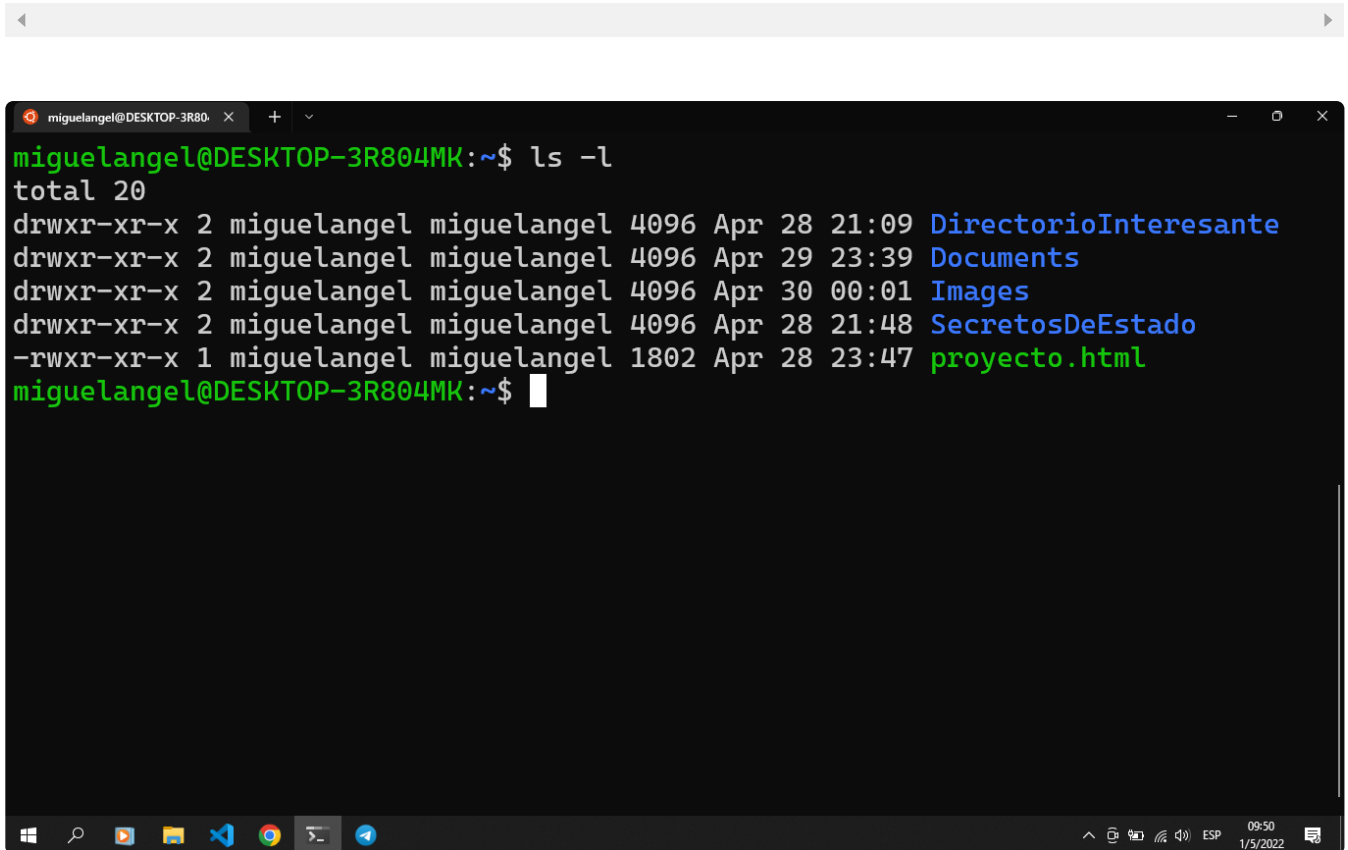


El 2 es **Standard Error**.

Cómo usar el operador de redirección (>)

A veces queremos guardar la información de una salida porque nos puede interesar almacenar lo que esa salida contiene. Veamos el siguiente ejemplo, si utilizas el comando:

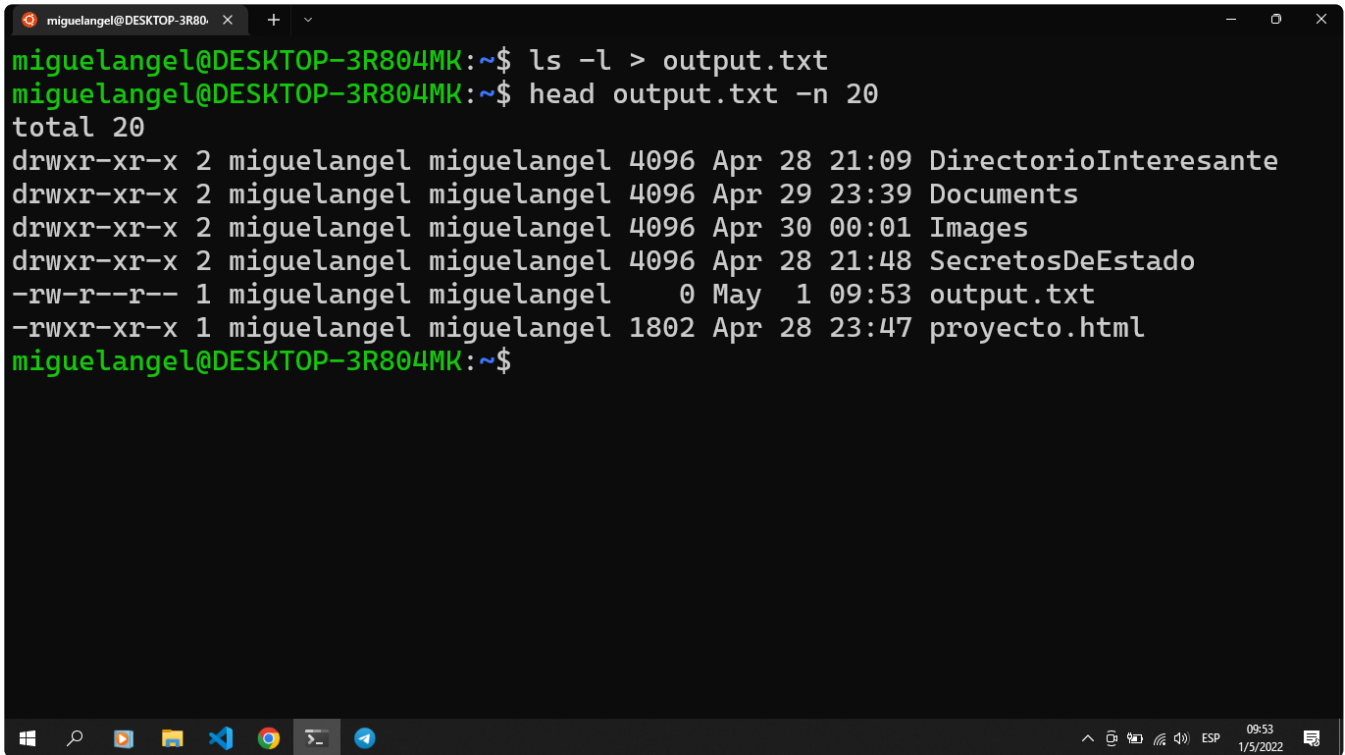
```
ls -l
```

A screenshot of a Windows terminal window. The title bar shows 'miguelangel@DESKTOP-3R804MK'. The terminal content shows the command 'ls -l' being executed, resulting in a list of files and directories with their permissions, owner, group, size, date, and name. The files listed are 'DirectorioInteresante', 'Documents', 'Images', 'SecretosDeEstado', and 'proyecto.html'. The prompt 'miguelangel@DESKTOP-3R804MK:~\$' is visible at the bottom of the terminal output. The Windows taskbar is visible at the bottom of the screen.

Lo que sucede aquí es que le diste un **Standard Input** (el comando) y obtuviste un **Standard Output** (la lista de archivos).

Si quieres que el **Standard Output** no vaya a la consola sino hacia un archivo, entonces puedes usar el operador > seguido del nombre del archivo en el que quieres guardar la salida.

```
ls -l > output.txt
```

A terminal window titled 'miguelangel@DESKTOP-3R804MK' with a dark background. The user has entered two commands: 'ls -l > output.txt' and 'head output.txt -n 20'. The output shows a directory listing with permissions, owner, group, size, date, time, and filename. The files listed are 'DirectorioInteresante', 'Documents', 'Images', 'SecretosDeEstado', 'output.txt', and 'proyecto.html'. The prompt 'miguelangel@DESKTOP-3R804MK:~\$' is visible at the bottom of the terminal. The Windows taskbar is visible at the bottom of the screen.

```
miguelangel@DESKTOP-3R804MK:~$ ls -l > output.txt
miguelangel@DESKTOP-3R804MK:~$ head output.txt -n 20
total 20
drwxr-xr-x 2 miguelangel miguelangel 4096 Apr 28 21:09 DirectorioInteresante
drwxr-xr-x 2 miguelangel miguelangel 4096 Apr 29 23:39 Documents
drwxr-xr-x 2 miguelangel miguelangel 4096 Apr 30 00:01 Images
drwxr-xr-x 2 miguelangel miguelangel 4096 Apr 28 21:48 SecretosDeEstado
-rw-r--r-- 1 miguelangel miguelangel    0 May  1 09:53 output.txt
-rwxr-xr-x 1 miguelangel miguelangel 1802 Apr 28 23:47 proyecto.html
miguelangel@DESKTOP-3R804MK:~$
```

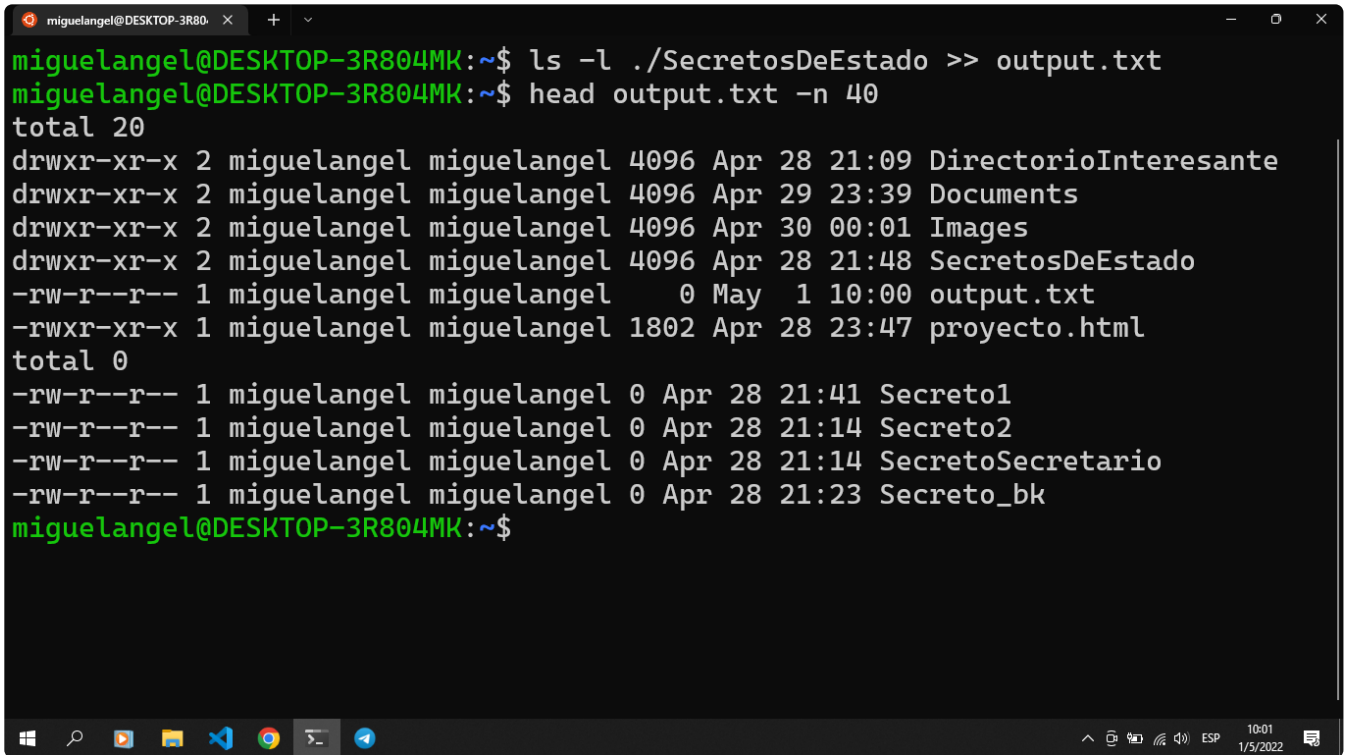
Cómo concatenar (>>)

Suponiendo que ya tienes el archivo output.txt y ahora también quieres guardar la información de la carpeta de documentos, entonces no puedes volver a ejecutar:

```
ls -l > output.txt
```

Esto lo que hará es reescribir el contenido del documento, lo que necesitas es concatenar el contenido del documento con el de la salida, para eso ejecutas:

```
ls -l >> output.txt
```



```
miguelangel@DESKTOP-3R804MK:~$ ls -l ./SecretosDeEstado >> output.txt
miguelangel@DESKTOP-3R804MK:~$ head output.txt -n 40
total 20
drwxr-xr-x 2 miguelangel miguelangel 4096 Apr 28 21:09 DirectorioInteresante
drwxr-xr-x 2 miguelangel miguelangel 4096 Apr 29 23:39 Documents
drwxr-xr-x 2 miguelangel miguelangel 4096 Apr 30 00:01 Images
drwxr-xr-x 2 miguelangel miguelangel 4096 Apr 28 21:48 SecretosDeEstado
-rw-r--r-- 1 miguelangel miguelangel    0 May  1 10:00 output.txt
-rwxr-xr-x 1 miguelangel miguelangel 1802 Apr 28 23:47 proyecto.html
total 0
-rw-r--r-- 1 miguelangel miguelangel 0 Apr 28 21:41 Secreto1
-rw-r--r-- 1 miguelangel miguelangel 0 Apr 28 21:14 Secreto2
-rw-r--r-- 1 miguelangel miguelangel 0 Apr 28 21:14 SecretoSecretario
-rw-r--r-- 1 miguelangel miguelangel 0 Apr 28 21:23 Secreto_bk
miguelangel@DESKTOP-3R804MK:~$
```

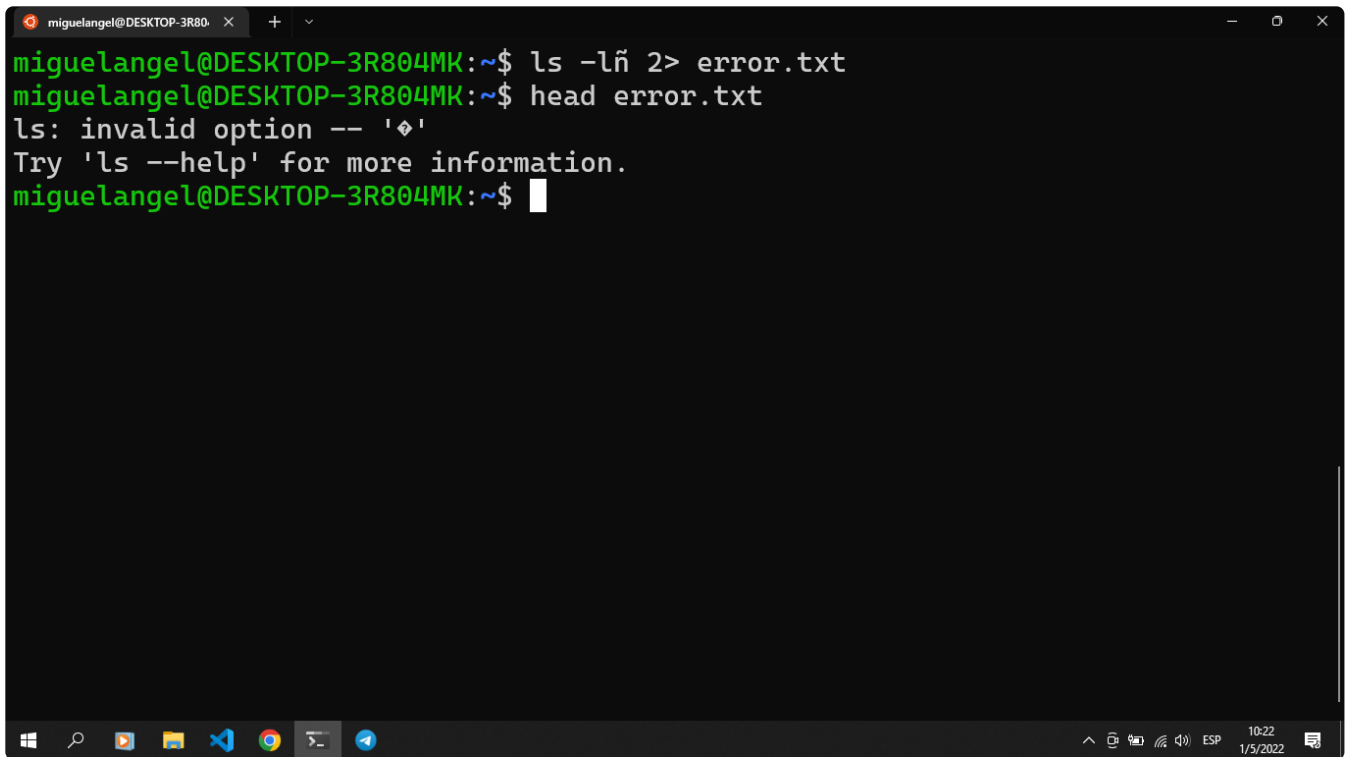
Como puedes ver, la salida del comando `ls -l` se concatenó con la salida del comando `ls -l ./SecretosDeEstado`. Te puedes dar cuenta porque la palabra `total` se repite dos veces.

Por cierto, esa palabra `total` es el tamaño total de la carpeta en kilobytes y dice que la carpeta `SecretosDeEstado` pesa 0, porque los archivos y carpetas vacías no ocupan espacio.

Redirección de errores (2>|2>&1)

El operador de redirección por defecto solo redirecciona el file descriptor 1 (es decir, el **Standard Output**). Pero, ¿qué tal si queremos redirigir un error? Pues tenemos que especificar que queremos el **Standar Error**, que tiene el file descriptor 2.

Vamos a generar un error ejecutando un comando que saldrá mal para redirigirlo a un archivo llamado "error.txt".

A screenshot of a Windows terminal window with a dark background. The window title bar shows 'miguelangel@DESKTOP-3R80...'. The terminal text shows a user prompt 'miguelangel@DESKTOP-3R804MK:~\$' followed by the command 'ls -lñ 2> error.txt'. The next line shows the user prompt again followed by 'head error.txt'. The terminal then displays an error message: 'ls: invalid option -- 'ñ'', followed by 'Try 'ls --help' for more information.' and then the user prompt again. The Windows taskbar is visible at the bottom with various icons and a system clock showing 10:22 on 1/5/2022.

```
miguelangel@DESKTOP-3R804MK:~$ ls -lñ 2> error.txt
miguelangel@DESKTOP-3R804MK:~$ head error.txt
ls: invalid option -- 'ñ'
Try 'ls --help' for more information.
miguelangel@DESKTOP-3R804MK:~$
```

En este caso la opción “ñ” no existe, por lo que produce un error.

También podemos especificar que no importa lo que pase si me da un **Standard Output** o un **Standard Error**, igual tiene que guardar la salida en un archivo. Esto lo hacemos así:

```
ls -l > output.txt 2>&1
```

La orden `2>&1` significa que debe redirigir el file descriptor 2 y el file descriptor 1.

```
miguelangel@DESKTOP-3R804MK:~$ ls -l > output.txt 2>&1
miguelangel@DESKTOP-3R804MK:~$ head output.txt
total 24
drwxr-xr-x 2 miguelangel miguelangel 4096 Apr 28 21:09 DirectorioInteresante
drwxr-xr-x 2 miguelangel miguelangel 4096 Apr 29 23:39 Documents
drwxr-xr-x 2 miguelangel miguelangel 4096 Apr 30 00:01 Images
drwxr-xr-x 2 miguelangel miguelangel 4096 Apr 28 21:48 SecretosDeEstado
-rw-r--r-- 1 miguelangel miguelangel 64 May 1 10:22 error.txt
-rw-r--r-- 1 miguelangel miguelangel 0 May 1 10:26 output.txt
-rwxr-xr-x 1 miguelangel miguelangel 1802 Apr 28 23:47 proyecto.html
miguelangel@DESKTOP-3R804MK:~$ ls -lñ > output.txt 2>&1
miguelangel@DESKTOP-3R804MK:~$ head output.txt
ls: invalid option -- 'ñ'
Try 'ls --help' for more information.
miguelangel@DESKTOP-3R804MK:~$
```

En la primera ejecución del comando, se ejecuta correctamente y guarda el **Standar Output**, pero en la segunda ejecución, el comando falla y guarda el **Standar Error**.

Tabla de operadores

Operador	Función
>	Redirecciona la salida. Por defecto redirecciona el Standar Output
>>	Concatena la salida con lo que ya tenga el archivo a donde se está redirigiendo la salida
2>	Redirecciona el file descriptor 2 (En este caso Standar Error)
2>&1	Redirecciona el file descriptor 2 y 1