

# Cómo cambiar la ubicación de los usuarios en Windows

## La situación de partida

He partido de este [artículo](#) que nos diste el cual funciona bien para cambiar lo siguiente:

- El usuario *default* (me imagino que este es un esquema usado por Windows para temas de integridad y mantenimiento de usuarios, en plan si borras la carpeta de downloads de tu usuario que esta no desaparezca o que directamente no se te permita tocarla).
- El usuario *public* el cual entiendo que contiene la información que se quiere que sea común entre los usuarios locales.
- La ubicación de creación de nuevos usuarios (bastante claro lo que esto es).

Lo que tienen en común estas 3 cosas es que ninguna contiene archivos considerados sensibles por Windows para el funcionamiento del SO. Esto significa que ni un usuario admin puede tocar estos archivos, ya que un admin no es realmente equivalente al root de Linux.

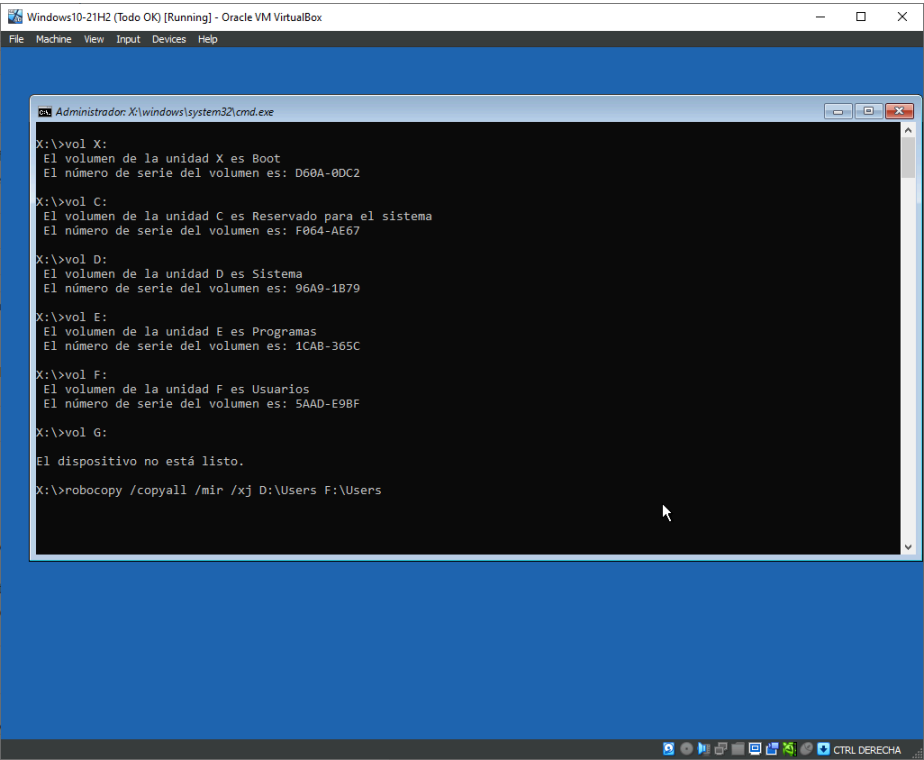
## Un dilema y una solución

Precisamente por esto el problema surge cuando tratas de mover los usuarios ya existentes a otro volumen, pues no quiero simplemente volver a crearlos como te sugiere de forma indirecta el [artículo](#) de arriba, ya que esa solución está limitada a solo poder hacerla cuando estás configurando un SO desde 0. ¿Qué hacemos entonces si hay archivos que no queremos perder, simplemente mover estos 1 a 1 como absolutos monos descerebrados?

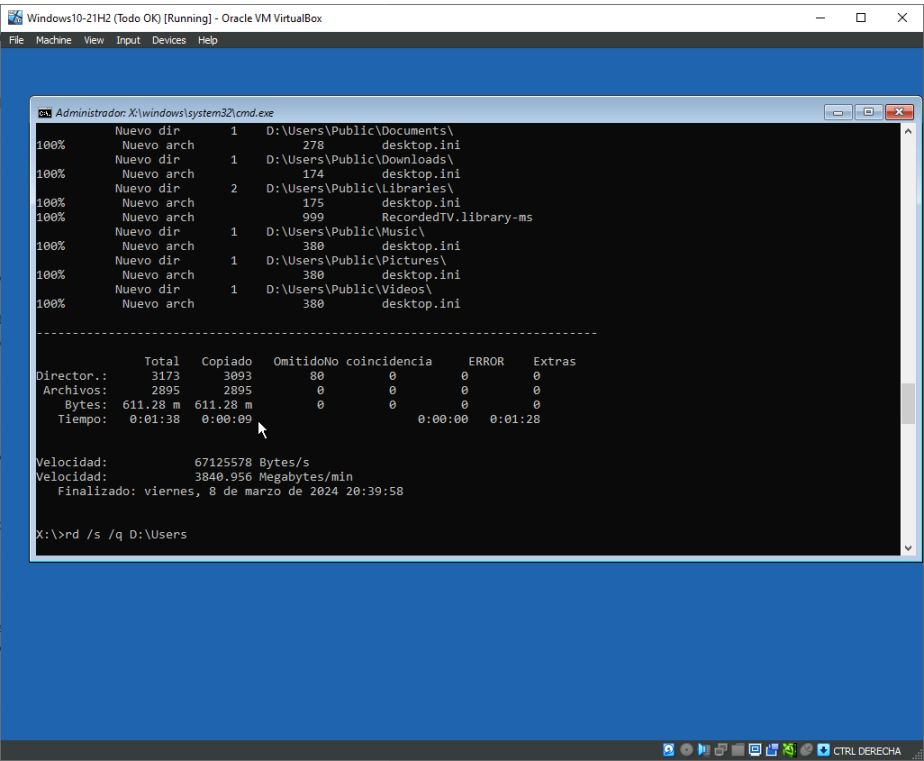
No. Por suerte alguien ya quería hacer esto mismo y se encontró falta de información y/o soluciones demasiado complejas para la simple tarea que él quería realizar (mover *users* de un volumen/partición a otro/a). [Este](#) es el artículo de esa persona en cuestión.

# La implementación de la solución

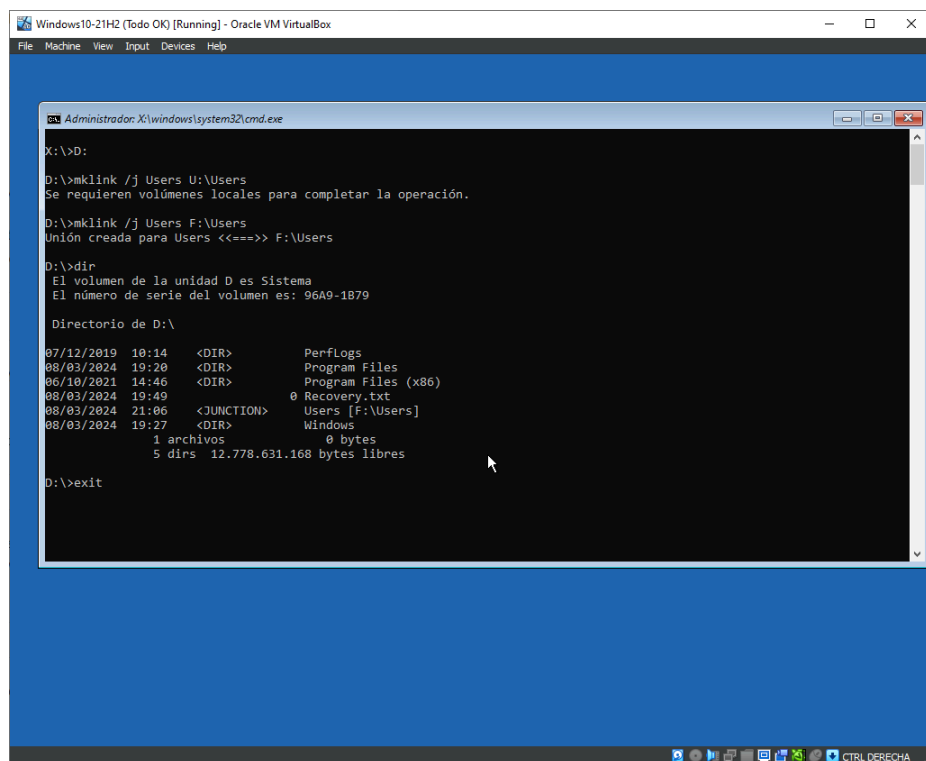
Para empezar dejar claro que se necesita no haber hecho ningún cambio de usuarios ni regedit, solo haber creado los usuarios nuevos de la tarea anterior sin moverlos de sitio.



Una vez dejado claro el punto de partida, lo primero es seguir los pasos del [artículo](#) hasta llegar a la consola de comandos del modo “troubleshooting”. Una vez en ella, se cerciora uno de cuáles son las letras de los volúmenes activos, tras lo cual se ejecuta el comando “robocopy /copyall /mir /xj <direccionOrigenUsuarios> <direccionDestinoUsuarios>” como se muestra en la imagen lateral.



Esto inicia un proceso de movimiento de carpetas que debería de finalizar con un resumen similar al de la imagen de la izquierda. No tengo claro cual es la “failed column” de la que se habla en el artículo (ni puedo saberlo porque mi copia de Windows está en español) pero he asumido que es ERROR. Tras ello eliminamos el directorio de origen el cual ya hemos copiado con éxito en su totalidad.



```
Windows10-21H2 (Todo OK) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help

Administrador: X:\windows\system32\cmd.exe

X:\>D:
D:\>mklink /j Users U:\Users
Se requieren volúmenes locales para completar la operación.

D:\>mklink /j Users F:\Users
Unión creada para Users <====> F:\Users

D:\>dir
El volumen de la unidad D es Sistema
El número de serie del volumen es: 96A9-1B79

Directorio de D:\

07/12/2019  10:14    <DIR>          PerfLogs
08/03/2024  19:20    <DIR>          Program Files
06/10/2021  14:46    <DIR>          Program Files (x86)
08/03/2024  19:49                0 Recovery.txt
08/03/2024  21:06    <JUNCTION>    Users [F:\Users]
08/03/2024  19:27    <DIR>          Windows
                1 archivos            0 bytes
                5 dirs  12.778.631.168 bytes libres

D:\>exit
```

Tras ello, para dar el pego hasta que arreglemos los registros de Windows con RegEdit, tenemos que hacer creer a este que los usuarios siguen estando en el volumen de origen. Para ello hacemos un acceso directo (enlace simbólico) en el volumen de origen y salimos de la consola como aparece en la imagen. Tras ello reiniciamos el SO y tratamos de entrar con el Admin. Si no podemos entrar como si nada entonces te has cargado la MV y toca volver a una snapshot o clon funcional.

Una vez de nuevo en el usuario Admin, se debe de seguir los pasos del [artículo](#) que nos proporcionaste para cambiar esos 3 registros que no dan problemas (*public*, *default* y ubicación para nuevos usuarios), tras lo cual se deberá de hacer lo mismo con los registros de dirección de cada usuario (suelen estar en los subdirectorios con un porrón de dígitos que están dentro del directorio donde se cambian los 3 primeros registros que ya habremos alterado). Una vez hecho estas dos cosas deberíamos de poder borrar el acceso directo en el viejo volumen donde solíamos alojar los usuarios sin problemas.

## Consideraciones a tener en cuenta

Tras múltiples intentos y MV destrozadas en el proceso de implementar la solución de esta buena persona, he encontrado unas cuantas peculiaridades a tener en cuenta las cuales no son tratadas en el [artículo](#) anteriormente mencionado o no son bien explicadas desde mi punto de vista. Estas son:

- La relevancia de la letra asociada a los volúmenes
- El funcionamiento de los comandos
- La recomendación de Windows de abstenerse de hacerlo

### Letras, discos y volúmenes

Algo que no se explica totalmente bien en el [artículo](#) en inglés en cuestión es la mecánica de las letras y los volúmenes, la cual está relacionada con el tipo de consola que estamos utilizando al reiniciar de esta forma. Esta es una consola de recuperación/reparación, lo cual implica que debe conceder más libertad de manipulación del SO. Esto probablemente se deba a que la herramienta se lanza desde el volumen X: (haciendo que los demás volúmenes no estén en uso). Lo cual nos lleva al nombre de los volúmenes.

Para asegurarte es mejor simplemente usar el comando “vol [A-Z]” para ver qué letras están disponibles y cuáles no, pero la asignación de letras en esta herramienta tiene pinta de depender del tipo de disco que tienes (básico o dinámico):

Si son básicos, entonces empieza a asignarles letras en C y va recorriendo el abecedario hasta asignar a cada volumen una letra. Ahora viene la parte divertida (y no, no es el por qué empieza en A en vez de C, probablemente porque Windows antiguamente reservaba los volúmenes A y B para algo del kernel o así, o porque C es de *Core*, a saber, porque no me importa). La parte divertida es que cuando se instala Windows (7 en adelante al menos) se asigna un volumen sin letra al SO y ese es primero para la herramienta (para ser justos para el administrador de discos también es así solo que ahí no se le asigna letra).

Lo importante de esto es que los volúmenes que uno tiene van en D en adelante en la consola (aunque tu les cambies la letra en el admin de discos). El quid de la cuestión es entonces el siguiente. Necesitas alinear la letra que le pongas a los volúmenes con las que la consola les asigna. Esto no sería necesario si no fuera por 2 razones:

1. La consola no te deja operar con una letra a la que no le tenga asignada un volumen.
2. El comando *mklink* necesita en nuestro caso que le introduzcas una ruta absoluta de destino (cuya letra de volumen no va a ser parseada a la letra que tu le tengas asignada realmente).

Esto implica que si tienes el volumen Usuarios en la U: (de usuarios), no te dejará ponerla en el comando *mklink* porque, seguramente, no tenga 19 volúmenes para rellenar de C a U (y sí, hablo por experiencia propia y me cargué un MV por probar con la letra que la consola asignaba a mi volumen por no poder usar la que yo le había asignado).

Mi recomendación entonces es dejar la asignación base de las letras de los volúmenes (estos se ordenan solos de C: en adelante) y recordar que la letra del volumen en la consola es la siguiente a la que tiene asignada en el volumen.

Y si te estás preguntado qué pasa si el disco es dinámico, es muy sencillo. La consola copia las letras que les asignes tú a los volúmenes en el admin de discos (de hecho a la partición del sistema que antes no tenía letra, ahora si la tiene). La desventaja directa de esto es que el único volumen de arranque es el que tiene el SO instalado, aunque a cambio de no tener restricciones con la asignación de letras la verdad es que me conformo. Esto puede ser cambiado en el admin de discos o tras crear varias particiones (que es como yo lo descubrí).

## Comandos de la consola

Ahora sobre los comandos, supongo esto es porque asume que los lectores los sabrán pero al investigar lo que hacían los comandos me encontré primero con el atributo */e* en vez de */mir* en el caso de *robocopy* y no entendía porque no querías hacer */mir* y */e* a la vez, resulta que */mir* es */e* + */purge*, obviamente. Por otro lado */xj* es simplemente para excluir los accesos directos (supongo que viene de *exclude junction/join*). Sobre */j* en *mklink* es para crear un *soft link* que aparentemente es el tipo de vínculo que te permite apuntar a otro volumen.

## Un falso rayo de esperanza

Sobre la advertencia de Microsoft de no hacer esto hace falta aclarar que lo dicen porque puede ser que no se pueda actualizar el SO. Lo sé, suena como un bonus más que una desventaja. Por desgracia para nosotros, esta limitación no se aplica a partir de Windows 10 así que no podremos aprovecharnos de ello.

## Cierre

Sé que esto es algo extenso pero es lo que consigues al preguntarme que te explique lo que he hecho para conseguir algo relativamente complejo e interesante. Hay varias cosas de las que no he hablado como que es *mklink*, *robocopy*, */copyall*, etc pero me parecía excesivo explicar cosas que son tan obvias (para el uso que les queremos dar, porque no me voy a poner a examinar el código fuente detrás de la función *robocopy* sin que me lo conteis como proyecto de fin de grado).