

¿No te parece que el comando whoami es un poco inútil? ¿Qué sentido tiene saber quién soy yo... ¡si yo sé quién soy!.. 🤖.

Bien, pues déjame decirte que tiene mucho sentido, y aquí te voy a explicar algunos de sus usos, pero primero, déjame decirte que en la terminal puedes ejecutar comandos dentro de comandos, ¿cómo? ¡Sencillo!, simplemente escribes: \$(cualquiercomando) y este se ejecutará, por ejemplo:

```
cat $(find ~/Escritorio -name test.txt)
```

```
retaxmaster@RetaxMaster:/var/www/html/curso-intermedio-python (main) $ cat $(find ~/Escritorio -name test.txt)
Hola!!
```

```
Este es un archivo de tests uwu
```

```
retaxmaster@RetaxMaster:/var/www/html/curso-intermedio-python (main) $
```

En este comando estoy ejecutando el comando cat (un comando que ya debes saber qué hace), pero a su vez, dentro de \$() estoy ejecutando un comando llamado find (little spoiler), este comando buscará un archivo dentro de cierta ubicación y te devolverá la ruta completa de ese archivo, entonces, como ya sabes que te devuelve la ruta completa de ese archivo, el resultado de ese comando se lo está pasando al comando cat.

Pero ahora, ¿cómo podemos aplicarlo al comando whoami? ¡Fácil!, y de hecho lo haremos aplicando un comando de los que acabamos de ver para modificar permisos.

¡Ejemplo práctico!

Este ejemplo que te voy a dar es un problema que a muchas personas les pasa en el [Curso de Pework: Configuración de Entorno de Desarrollo en Windows](#).

Imagina que crear una carpeta llamada PersonalProjects, pero la creas de la siguiente manera:

```
sudo mkdir PersonalProjects
```

Al hacer eso, tu carpeta se creará, sí, pero si intentas escribir archivos dentro de ella no podrás, te dirá "Permission denied", porque tú no eres el dueño de esa carpeta. Como creaste la carpeta usando sudo, entonces el dueño de esa carpeta es el usuario root 🙌🤖,

¡Cambiemos al dueño de la carpeta!, para hacer esto, podemos usar el comando chown (Change Owner), este comando se estructura de la siguiente forma:

```
chown -R <user>:<group> folder
```

Entonces, yo podría poner esto:

```
chown -R retaxmaster:retaxmaster PersonalProjects
```

Eso sería suficiente para que yo sea el dueño de mi carpeta y pueda escribir en ella. ¿Por qué el grupo tiene mi nombre de usuario? Porque, en Linux, al crear un usuario, automáticamente se crea un grupo con el mismo nombre.

Muy bien, pero aún no te he dicho en qué interviene whoami aquí, pero seguro ya lo intuyes 🤖...

Imagina que más personas tienen el mismo problema, y en lugar de decirles algo como: "Primero tienes que saber cuál es tu usuario, luego pones este comando, pero cambias ahí donde dice 'usuario' por tu propio usuario..." mejor les dices: "Toma, con este comando lo solucionas", sería mucho más eficiente, ¿no crees? Hagamos ese comando:

```
chown -R $(whoami):$(whoami) PersonalProjects
```

¿Ves? Este comando se lo puedes dar a cualquiera y le funcionará, porque whoami está poniendo dentro de dichos parámetros el nombre del usuario!!

4/9/22, 12:43

¿No te parece que el comando whoami es un poco inútil? ¿Qué sentido tiene saber quién soy yo... ¡si yo sé quién soy!.. 🤖... Bi...

Esta es tan solo una de las muchas ventajas que tiene el comando `whoami`, al principio parece un comando inútil, pero cuando empiezas a combinarlo con más utilidades de la terminal empiezas a ver que es muy útil 🙌🤖.