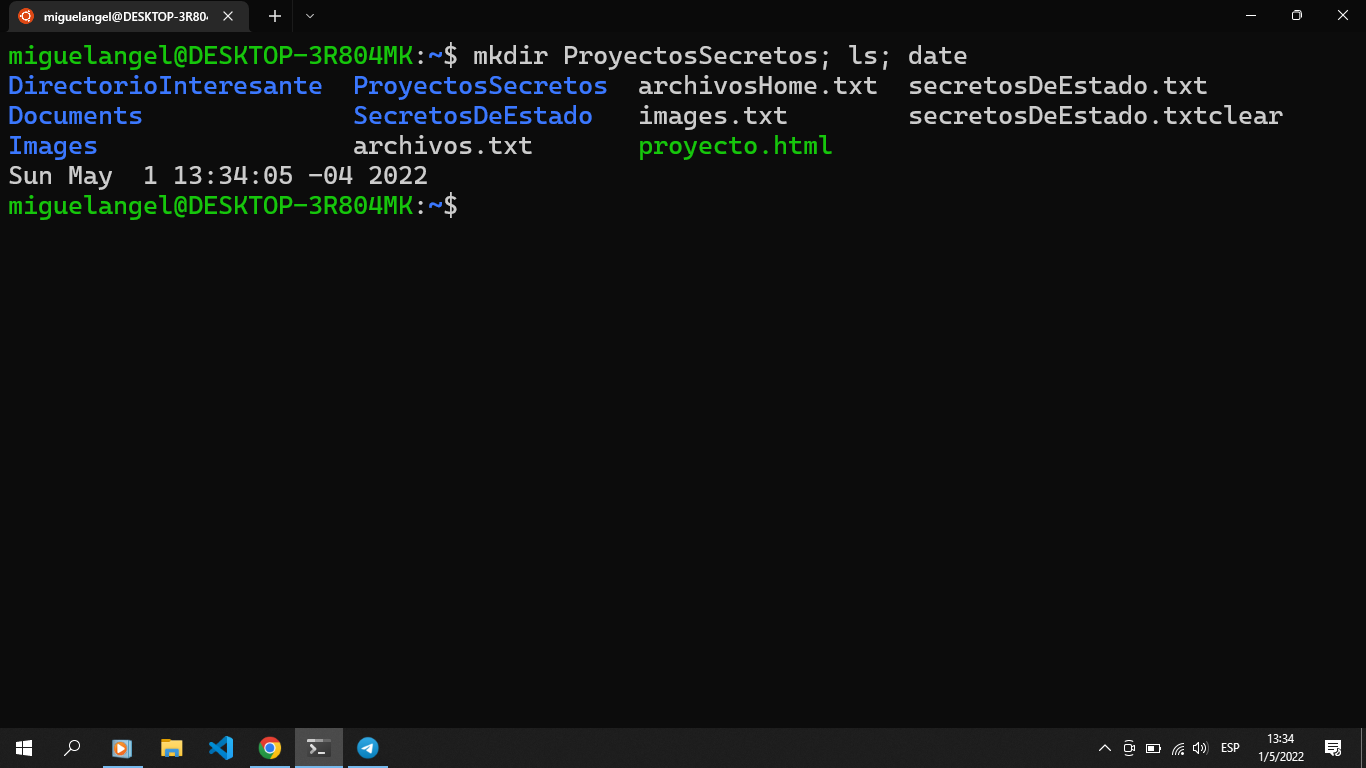
Los operadores de control son símbolos reservados por la terminal que nos permiten encadenar comandos.

Si usas constantemente la tecla enter para ejecutar varios comandos, puedes evitarlo si usas el operador **;** que separa los comandos que estamos ejecutando.

**Comandos en la misma línea (;)**

Solo necesitas escribir los comandos que quieres ejecutar separados por **;** y luego presionar enter.

mkdir ProyectosSecretos; ls; date



Lo que sucedió es:

* Se creó un directorio llamado "Proyectos Secretos"
* Se listaron los archivos
* Se imprimió la fecha actual

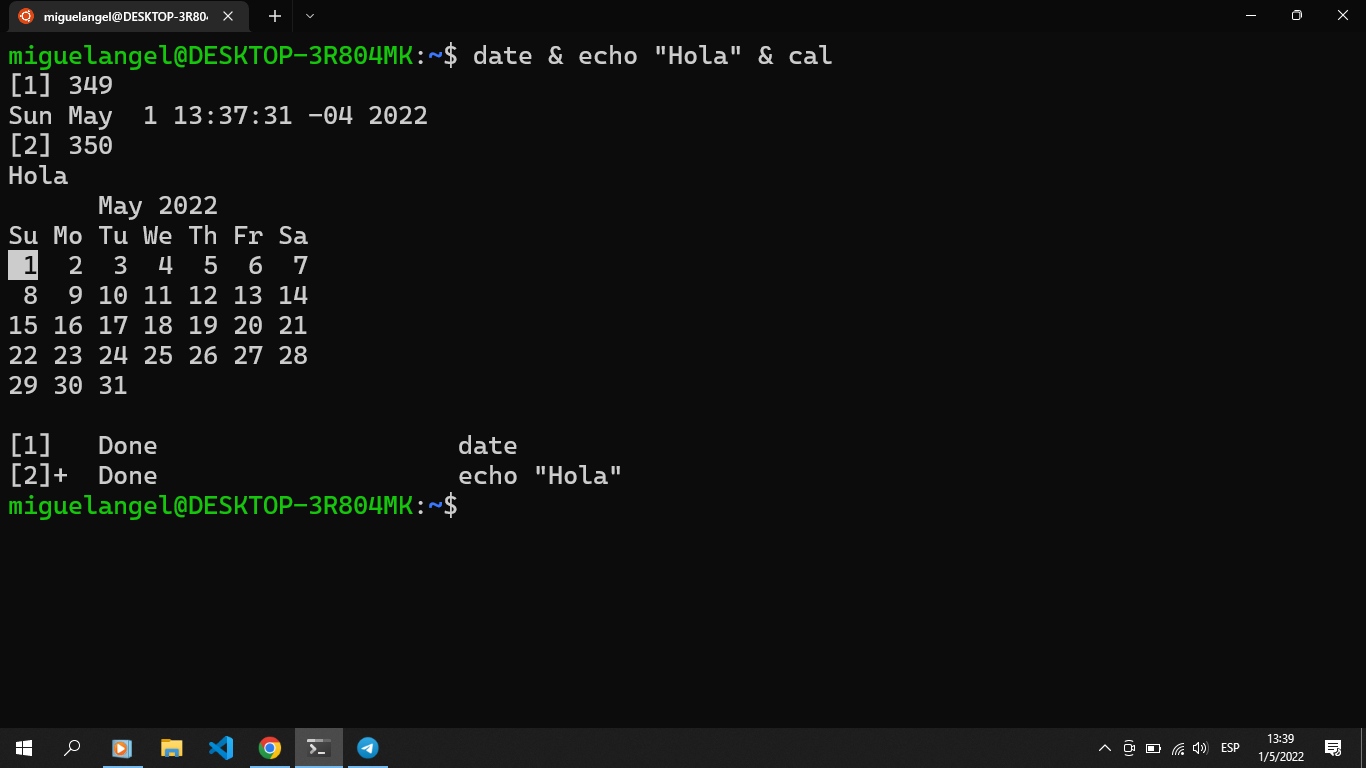
[El comando](https://platzi.com/clases/2292-terminal/37343-que-es-un-comando/) date imprime por consola la fecha actual.

**Comandos asíncronos (&)**

Cuando queremos ser más eficientes podemos ejecutar varios comandos al mismo tiempo, de modo que no tenemos que esperar a que uno se ejecute para luego ejecutar el que sigue.

Para llevar a cabo varios comandos, al mismo tiempo, usamos el operador **&** entre cada comando que queremos ejecutar.

date & echo "Hola" & cal



El comando cal imprime un pequeño calendario de la fecha actual y el comando echo imprime por el texto que le pases.

En la salida podemos ver que en la primera línea dice [1] 349 y en la tercera dice [2] 350, esto significa que se crearon dos hilos para ejecutar los 3 comandos que se le dieron.

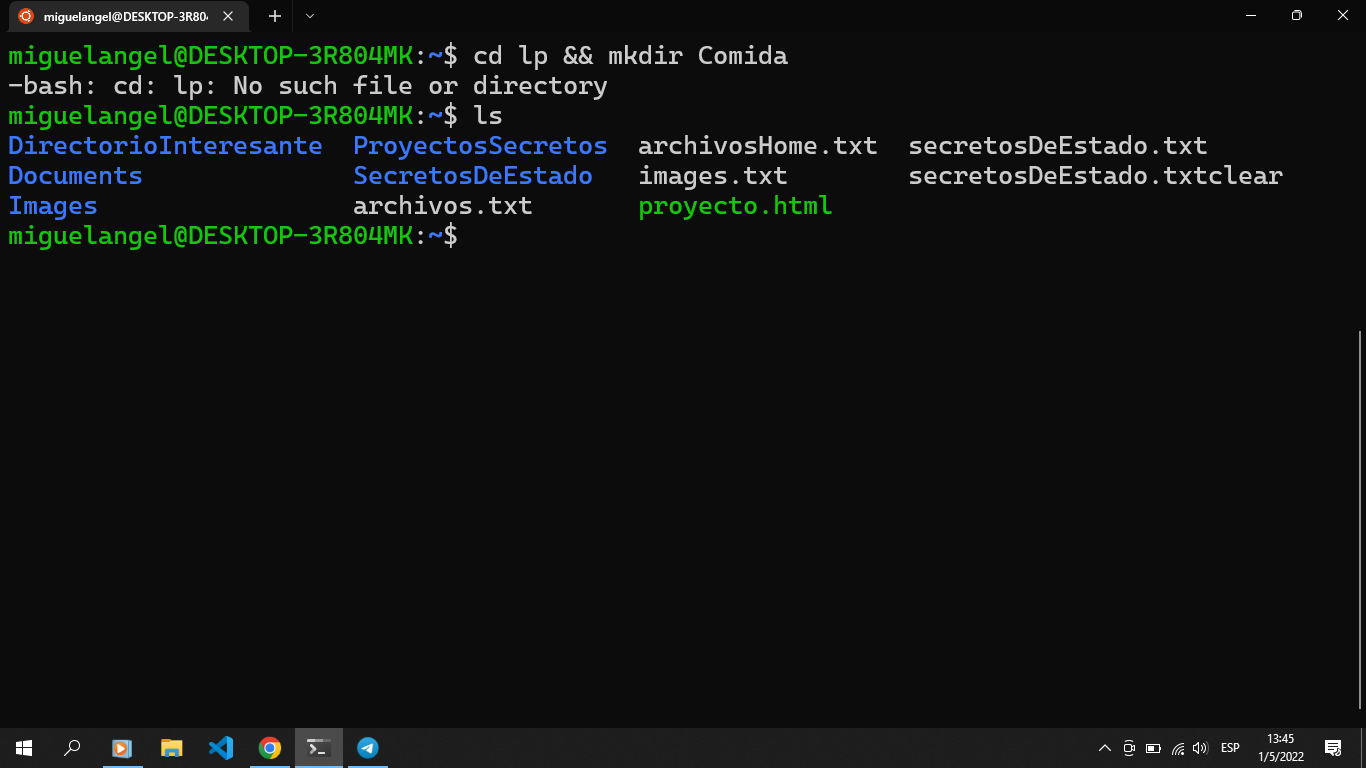
El primero, con el id 349, se usó para ejecutar el comando date y el segundo, con el id 350 se usó para ejecutar los comandos echo y cal.

**Comandos con condicionales**

Podemos ejecutar comandos dependiendo de condiciones.

**Condición and (&&)**

Si escribimos varios comandos separados por el operador **&&** estamos indicando que para que estos se ejecuten, el comando anterior tuvo que ejecutarse correctamente.

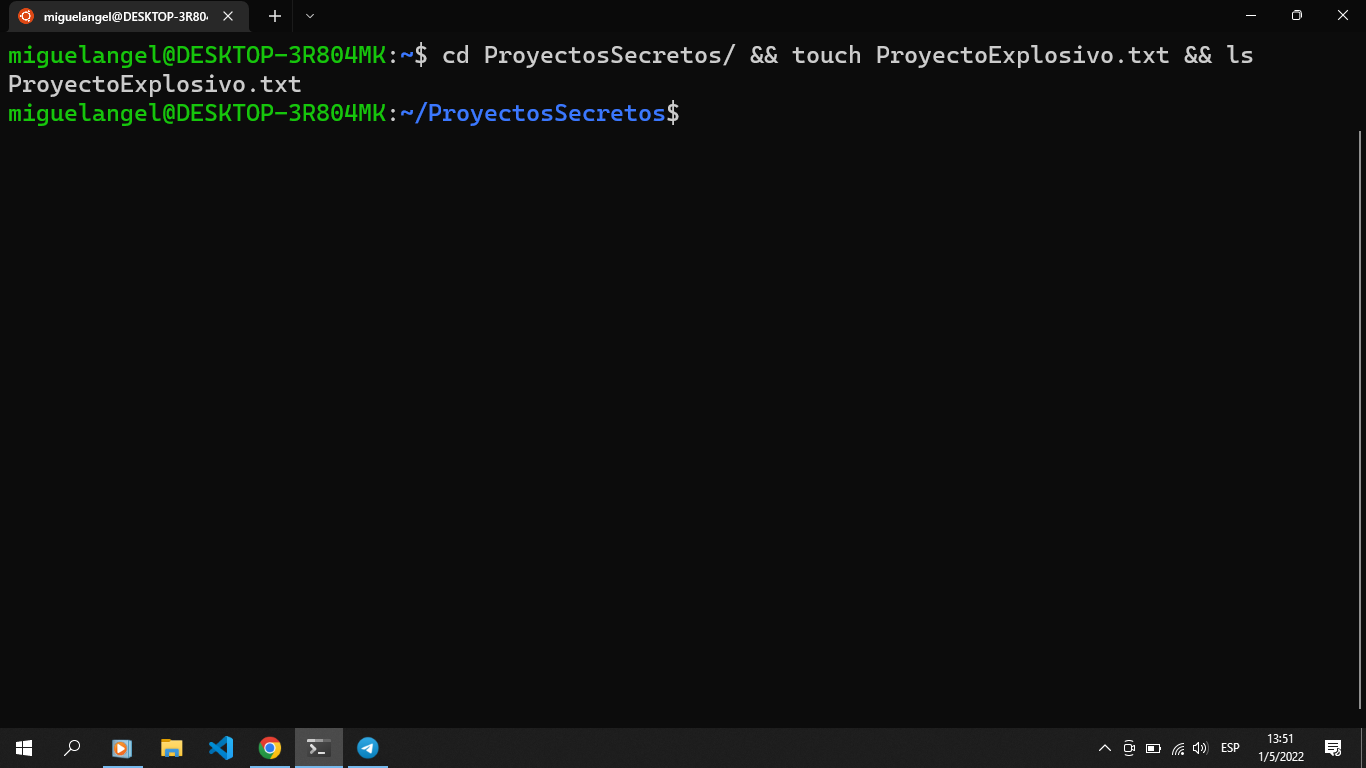
cd lp && mkdir Comida 

(Recuerda que el comando cd, **Change Directory** te cambia al directorio que le indiques).

En este ejemplo, intentamos cambiar al directrio "lp" pero ese directorio no existe, así que el comando falla y no se ejecuta el siguiente.

En caso de que el primer comando se haya ejecutado correctamente pasará al siguiente, y después verá si ese se puede ejecutar.

cd ProyectosSecretos/ && touch ProyectoExplosivo.txt && ls



Como en este caso todos los comandos se ejecutaron correctamente sucede esto:

* Cambia el directorio "ProyectosSecretos"
* Crea el archivo "ProyectoExplosivo.txt"
* Lista el contenido del directorio con ls

**Condicional or (||)**

Al condicional or no le importa si el comando anterior se ejecutó o no, simplemente va probando todos los comandos a ver cuál se ejecuta.

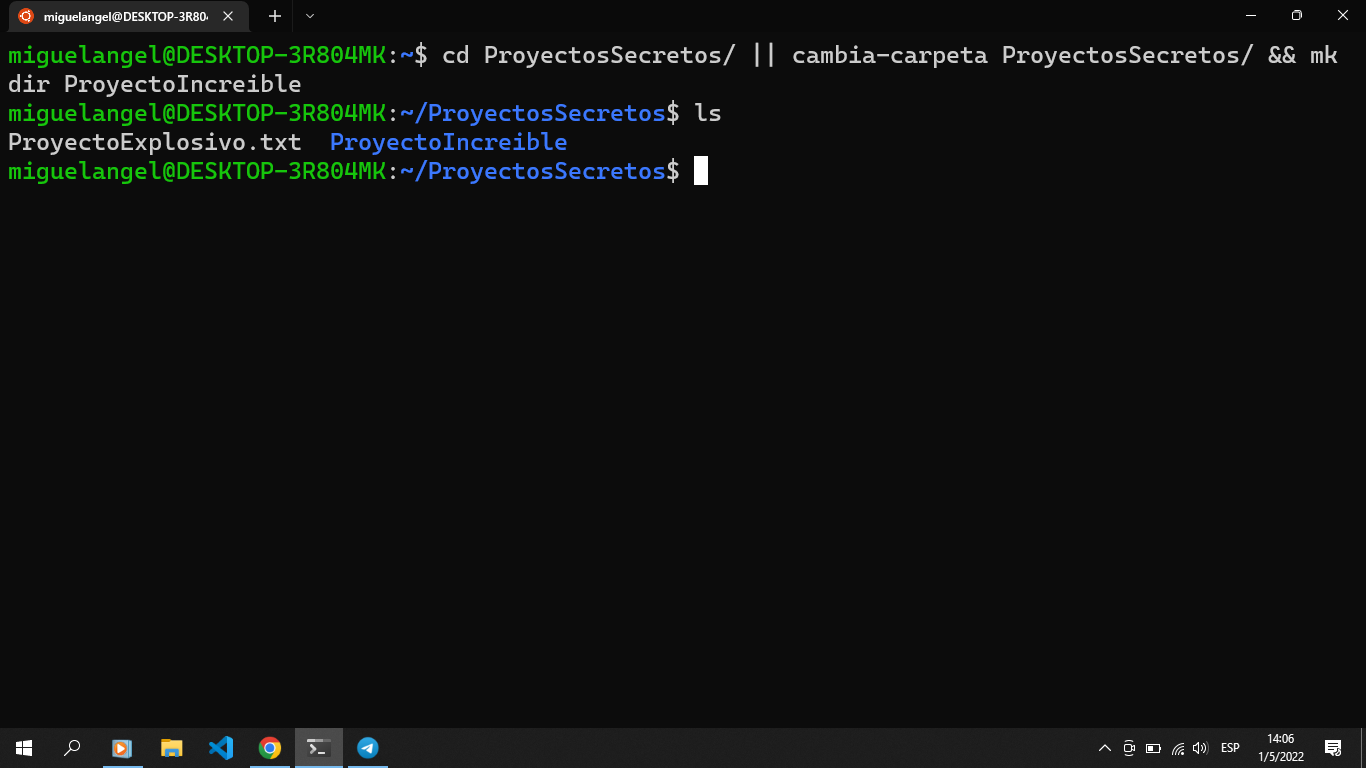
Por un momento vamos a suponer que no sabemos muy bien cuál es el comando para cambiar de directorio si es cd o cambia-carpeta, entonces para evitar un error escribimos:

cd ProyectosSecretos/ || cambia-carpeta ProyectosSecretos/

image.png

Aquí vemos que alguno de los dos comandos está mal, pero igual cambió la carpeta porque uno de esos funcionó.

**Combinando operadores de control**

Siguiendo con el ejemplo anterior, vamos a cambiar de directorio. Si se logra cambiar de directorio creamos una carpeta adentro. cd ProyectosSecretos/ || cambia-carpeta ProyectosSecretos/ && mkdir ProyectoIncreible 

Esto funcionó porque el operador **or** devolvió verdadero al tener un comando que funcionara, por lo tanto, el operador **and** lo interpretó como un comando que funcionó correctamente.

Si esto último se te es un poco complicado te invito a que tomes el [Curso de Pensamiento Lógico.](https://platzi.com/cursos/pensamiento-logico/)

**Tabla de operadores**

| Operador | Función | | --- | --- | | ; | Ejecuta de forma **síncrona** los comandos específicados | | & | Ejecuta de forma **asíncrona** los comandos específicados | | && | Ejecuta el comando si el anterior se ejecutó correctamente | | \|\| | Ejecuta el comando si el anterior o la combinación de los anteriores resultaron en verdadero |

| Comando | Función | | --- | --- | | echo | Imprime el mensaje que le indiques | | cal | De **cal**endar imprime un calendario con la fecha actual | | date | Imprime la fecha actual |

*Contribución creada con los aportes de: Miguel Gonzalez.*

**Nota importante:**

* **Comandos separados por punto y coma “;” :** Se ejecutan uno seguido del otro en el orden en que fueron puesto. Uno no se ejecuta hasta que el otro proceso aya terminado.  
  –
* **Comandos separados por & :** Se ejecutan todos al mismo tiempo, es decir de forma asíncrona.  
  –
* **Comandos separados por && :** Se ejecutan solo si el comando anterior se aya ejecutado exitosamente. Suponemos que A, B y C son comando:  
  A && B && C  
  El B solo se va ejecutar si el A se ejecuta exitosamente, y el C solo se va ejecutar si el B si ejecuta exitosamente. Si el B no se ejecuta exitosamenta el C no se ejecuta. Si el A no se ejecuta exitomante el B y el C no se ejecutan.  
  –
* **Comandos separados por || :** Solo se ejecuta uno. Sea cuantos comandos tienes separados por || solo ejecuta o toma en cuenta el primer que se ejecuta exitosamente (bajo la redundacia), y descarta automaticamente los demas. Cuando uno de los comandos se ejecuta exitosamente, descarta los demas comandos.

**Nota importante:**

* **Comandos separados por punto y coma “;” :** Se ejecutan uno seguido del otro en el orden en que fueron puesto. Uno no se ejecuta hasta que el otro proceso aya terminado.  
  –
* **Comandos separados por & :** Se ejecutan todos al mismo tiempo, es decir de forma asíncrona.  
  –
* **Comandos separados por && :** Se ejecutan solo si el comando anterior se aya ejecutado exitosamente. Suponemos que A, B y C son comando:  
  A && B && C  
  El B solo se va ejecutar si el A se ejecuta exitosamente, y el C solo se va ejecutar si el B si ejecuta exitosamente. Si el B no se ejecuta exitosamenta el C no se ejecuta. Si el A no se ejecuta exitomante el B y el C no se ejecutan.  
  –
* **Comandos separados por || :** Solo se ejecuta uno. Sea cuantos comandos tienes separados por || solo ejecuta o toma en cuenta el primer que se ejecuta exitosamente (bajo la redundacia), y descarta automaticamente los demas. Cuando uno de los comandos se ejecuta exitosamente, descarta los demas comandos.